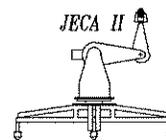




Universidade Estadual de Campinas  
 Faculdade de Engenharia Elétrica  
 LABORATÓRIO DE SISTEMAS MODULARES ROBÓTICOS  
 LSMR - DSCE



# Sistema de Sensoriamento de Robôs Industriais: Espaço de Juntas e Espaço Cartesiano.

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas, como parte dos requisitos exigido para obtenção do título de

**Doutor em Engenharia Elétrica**

por

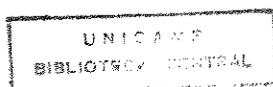
**Katsuhito Takita**

Mestre - Tokyo Institute of Technology

**Prof. Dr. Álvaro Geraldo Badan Palhares**  
 Orientador - DSCE/FEE/UNICAMP.

07 de julho de 1995

Este exemplar corresponde à redação final da tese defendida por Katsuhito Takita avaliada pela Comissão Julgadora em 07.07.95.  
  
 Orientador



UNIDADE	BC
N.º CHAMADA:	T/UNICAMP
	T139s
V.	E.
TÍTULO	125998
PROC.	433/95
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	R\$ 11,00
DATA	01/11/95
N.º CPD	

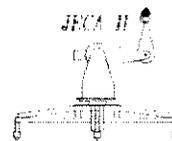
CM-00079395-5

**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP**

T139s Takita, Katsuhito  
Sistema de sensoriamento de robôs industriais: espaço de juntas e espaço cartesiano / Katsuhito Takita.--  
Campinas, SP: [s.n.], 1995.

Orientador: Alvaro Geraldo Badan Palhares  
Tese (doutorado) - Universidade Estadual de Campinas,  
Faculdade de Engenharia Elétrica.

1. Robôs industriais. 2. Detectores. 3. \*Sistemas mecatrônicos. I. Palhares, Alvaro Geraldo Badan. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica. III Título

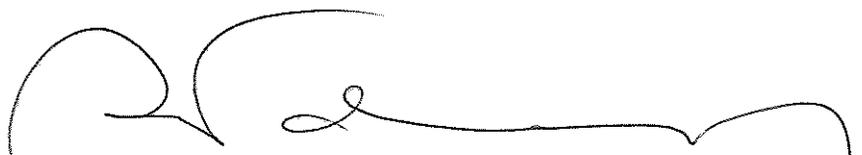


Tese: **Sistema de Sensoriamento de Robôs Industriais: Espaço de Juntas e Espaço Cartesiano.**

Autor: **Katsuhito Takita**

Orientador: **Prof. Dr. Álvaro Geraldo Badan Palhares**

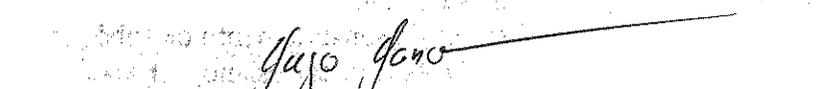
Aprovada em 7 de julho de 1995 pela comissão julgadora:



Prof. Dr. Álvaro Geraldo Badan Palhares (Presidente)



Prof. Dr. Clésio Luis Tozzi (FEE-UNICAMP)



Prof. Dr. Yuizo Iano (FEE-UNICAMP)



Prof. Dr. Geraldo Nonato Telles (FEM-UNICAMP)



Prof. Dr. Alberto Adade Filho (CTA-ITA-IEMP)

aos meus filhos Antonio e Julia

## AGRADECIMENTOS

Ao Professor Dr. Álvaro Geraldo Badan Palhares, pela orientação e dedicação profissional durante a realização deste trabalho.

À Fundação Banco do Brasil pelo auxílio financeiro, sem o qual não teria sido realizado este trabalho.

À CAPES pela bolsa de estudo.

Ao Sr. Yoshikazu Kanatani pelo apoio recebido.

A todos os membros da Divisão de Robótica do Centro Tecnológico para Informática.

À FADESP, em especial Prof. Afonso Brito Chermont.

Ao Pró-Reitor de Pós-Graduação e Pesquisa da UFPA, Prof. Cristovam Picanço Diniz.

Ao Prof. Dr. Kensuke Hasegawa pelas sugestões.

À Professora Antonieta do Lago Vieira pela discussão técnica e ajuda na elaboração do trabalho.

# SUMÁRIO

RESUMO . . . . .	iv
<b>CAPÍTULO 1: INTRODUÇÃO . . . . .</b>	<b>1</b>
1.1 Introdução . . . . .	1
1.2 Sistemas Mecatrônicos. . . . .	3
1.2.1 Controle em um Sistema Mecatrônico. . . . .	5
1.3 Definição e Tipos de Sensores para Robôs. . . . .	5
1.4 Estrutura do Sistema Didático . . . . .	8
1.5 Estrutura da Tese . . . . .	9
<b>CAPÍTULO 2: ESTRUTURA MECÂNICA DE POSICIONAMENTO E ORIENTAÇÃO . . . . .</b>	<b>10</b>
2.1 Introdução. . . . .	10
2.2 Descrição Global. . . . .	11
2.3 Descrição Detalhada. . . . .	15
2.3.1 Base Linear. . . . .	15
2.3.2 Juntas do Ombro e do Cotovelo. . . . .	17
2.3.3 Juntas do Punho . . . . .	22
2.4 Sistema de Coordenadas e Ângulos das Articulações. . . . .	24
2.5 Transformações Cinemáticas . . . . .	25
2.5.1 Transformação Cinemática Direta . . . . .	25
2.5.2 Transformação Cinemática Inversa . . . . .	26
2.6 Seleção dos Motores . . . . .	33
2.6.1 Modelo Dinâmico . . . . .	33
2.6.2 Cálculo dos Torques Necessários Para Acionamento Das Juntas. . . . .	37
2.7 Volume de Trabalho . . . . .	44
<b>CAPÍTULO 3: ELEMENTO FINAL DE AÇÃO . . . . .</b>	<b>46</b>
3.1 Introdução . . . . .	46
3.2 Garra com Sensores. . . . .	47
3.3 Parte Mecânica da Garra . . . . .	48
3.4 Sensor de Força. . . . .	48
3.4.1 Sensor Analógico de Força. . . . .	48
3.4.2 Sensor Digital de Força. . . . .	50
3.4.3 Calibração de Sensor de Força. . . . .	53
3.5 Sensor de Deslizamento. . . . .	55
3.6 Sensor de Abertura dos Dedos. . . . .	56
3.7 Controlador da Garra. . . . .	58

3.7.1 Hardware . . . . .	58
3.7.2 Software . . . . .	60
3.8 Experimentos Realizados. . . . .	62
3.8.1 Controle de Força. . . . .	62
3.8.2 Controle de Abertura dos Dedos. . . . .	62
3.8.3 Controle de Força dos Dedos com Sensores de Deslizamento. . . . .	64
<b>CAPÍTULO 4: SENSOR DE VISÃO . . . . .</b>	<b>67</b>
4.1 Introdução . . . . .	67
4.2 Descrição Técnica. . . . .	67
4.2.1 Dispositivo de Captura da Imagem. . . . .	68
4.2.2 Circuito Digitalizador e Armazenamento de Imagem. . . . .	69
4.3 Software de Sensor de Visão. . . . .	69
4.3.1 Cálculo do Centro Geométrico do Objeto. . . . .	72
4.3.2 Detecção da Borda do Objeto . . . . .	76
4.3.3 Determinação da Função Característica do Objeto. . . . .	77
4.3.4 Reconhecimento do Objeto. . . . .	84
4.3.5 Normalização . . . . .	88
4.3.6 Determinação da Orientação do Objeto. . . . .	88
4.4 Resultados Obtidos. . . . .	92
4.5 Futura Aplicação do Sensor de Visão . . . . .	104
<b>CAPÍTULO 5: SISTEMA HIERÁRQUICO DE CONTROLE DE POSIÇÃO E ORIENTAÇÃO</b> . . . . .	<b>106</b>
5.1 Introdução. . . . .	106
5.1.1 Controle Supervisor de Tarefas . . . . .	107
5.1.2 Controle de Execução de Tarefas . . . . .	107
5.1.3 Controle de Movimentos . . . . .	108
5.1.4 Controle de Juntas . . . . .	108
5.2 Hardware . . . . .	108
5.3 Software. . . . .	111
5.3.1 Módulo de Análise. . . . .	113
5.3.2 Módulo de Calibração e Inicialização. . . . .	114
5.3.3 Módulo de Programação. . . . .	114
5.3.4 Módulo de Execução Passo a Passo. . . . .	124
5.3.5 Módulo de Execução Automática. . . . .	125
5.3.6 Algoritmo de Ajuste e Controle de Trajetória. . . . .	126
5.4 Experimentos Realizados . . . . .	145
5.4.1 Rastreamento de Trajetória Retilínea Realizados. . . . .	145
5.4.2 Agarramento do Objeto Visto pelo Sensor de Visão. . . . .	148
<b>CAPÍTULO 6: CONCLUSÃO . . . . .</b>	<b>151</b>
6.1 Estrutura Mecânica de Posicionamento e Orientação . . . . .	151
6.2 Garra com Sensores . . . . .	151
6.3 Sensor de Visão . . . . .	152
6.4 Sistema Hierárquico de Controle de Posição e Orientação . . . . .	153
<b>APÊNDICE A: HARDWARE DO SENSOR DE VISÃO . . . . .</b>	<b>155</b>
A.1 Sinal de Vídeo Composto . . . . .	155

A.2 Digitalização da Informação da Imagem . . . . .	156
A.3 Circuito de Extração do Sinal de Vídeo . . . . .	162
<b>APÊNDICE B: <i>HARDWARE</i> DO CONTROLADOR DO ROBÔ . . . . .</b>	<b>170</b>
B.1 Descrição de Circuitos de Interfaceamento . . . . .	170
B.1.1 Decodificador de Endereços. . . . .	170
B.1.2 Contador <i>UP/DOWN</i> . . . . .	172
B.1.3 Conversor D/A. . . . .	176
B.1.4 Entrada Digital de 16 Bits. . . . .	178
B.1.5 Circuito de Comunicação Paralela. . . . .	178
<b>APÊNDICE C: FOTOGRAFIAS DO ROBÔ . . . . .</b>	<b>181</b>
C.1 Visão Geral do Sistema do Robô . . . . .	181
C.2 Detalhes do Punho e da Garra . . . . .	182
C.3 Vista Frontal do Robô . . . . .	183
C.4 Transporte de Objeto . . . . .	184
<b>BIBLIOGRAFIAS . . . . .</b>	<b>185</b>

## RESUMO

Este trabalho apresenta o projeto e construção de um Robô Industrial de cinco graus de liberdade com a finalidade de atender um largo espectro de aplicações industriais, desde as aplicações cujas tarefas exigem baixo desempenho dinâmico até aquelas onde o alto desempenho é requerido.

A filosofia adotada no projeto levou em conta que este robô pudesse ser programado com estratégias de controle que permitissem um ótimo desempenho para cada tipo de aplicação a que for destinado. Com isso procurou-se ressaltar neste trabalho um dos problemas básicos para se obter um perfeito desempenho dos algoritmos de controle, de acordo com a estratégia adotada para cada aplicação, que é o problema das medidas em tempo das variáveis de movimento, força, torques, deslizamento, proximidade e visão.

Classificamos tais medidas como sendo contidas em dois espaços distintos e que denominamos:

- Espaço de Juntas, sensores de juntas ou internos;
- Espaço Cartesiano, sensores cartesianos ou externos.

A realimentação das variáveis cartesianas em um robô que não conta com sensores cartesianos é feita de forma indireta pela resolução do modelo cinemático direto e inverso, e isso traz algumas desvantagens tanto do ponto de vista de erros na resolução do modelo como do ponto de vista do tempo de processamento requerido para a sua solução, principalmente no que diz respeito ao modelo reverso.

Adotamos então a filosofia de analisar a viabilidade de dotar o robô industrial de um sistema sensorial que permita a adoção de estratégia de controle no espaço de juntas ou de forma híbrida.

Nesta fase do projeto, que corresponde a esta tese de doutorado, o esforço foi dedicado na geração de conhecimentos básicos sobre a construção de um robô com ênfase no sistema de sensoriamento, porém tendo a preocupação de conhecer todas as etapas e sistemas que fazem parte do equipamento, tais como:

1. Sistema Digital de Controle
  - a. *Hardware*
  - b. *Software*
2. Sistema de Acionamento
  - a. Atuadores - Critério de Seleção
  - b. Amplificadores
3. Estrutura Mecânica
  - a. Atingibilidade do volume de trabalho
  - b. Capacidade de Carga

Como resultado principal deste trabalho, apresentamos um robô que executa operações tais como:

- Rastreamento de trajetórias retilíneas;
- Operação pega-larga com monitoramento da força sobre o objeto e deslizamento;
- Operação de manuseio de objetos sensíveis à força como, por exemplo, um ovo.

No desenvolvimento deste trabalho, a etapa que demandou maior esforço foi o projeto do mecanismo.

PALAVRAS CHAVES: Robô, Sensores, Sensor de Visão, Controle de Trajetória e Sistema Mecatrônico.

## ABSTRACT

This work presents the design and construction of the five degrees of freedom Industrial Robot with external sensors integration such as a visual sensor, slip and force sensors in the two fingers hand.

The main subjects of this work are:

- Development of digital control system composed by hardware and software;
- Design of driving system with selection criteria of actuators and amplifiers;
- Development of the mechanical structure with study about working volume and load capacity;
- Development of a two fingers mechanical hand with slip and force sensors;
- Development of a visual sensor with their application in object coordinate monitoring and shape recognition.

It was obtained, as result, a robot with capability to execute operations such as:

- Straight line trajectory tracing;
- Pick and place operation with gripping force and slip monitoring;
- Handling of fragile object such as an egg;
- Detection of coordinates of the geometrical center of an object and shape recognition.

The major effort of this work was dedicated to the mechanism design.

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 INTRODUÇÃO

O homem criou continuamente, ao longo da história novas ferramentas de trabalho. A primeira ferramenta foi criada pelo homem das cavernas, a pedra lascada e posteriormente a pedra polida, para o trabalho e a guerra. A importância das ferramentas está intimamente ligada à história da riqueza do homem, pois em cada nova conquista tornava-se necessário desenvolver novas ferramentas, e ao desenvolver novas ferramentas mais conquistas eram obtidas. Neste processo contínuo hoje podemos ver diversos tipos de ferramentas facilitando ao homem a execução das suas tarefas.

Nesse sentido, o robô industrial é uma ferramenta versátil, flexível e multifuncional e como qualquer outra ferramenta serve para facilitar a execução das tarefas dos homens, principalmente naquelas que são perigosas ou nocivas à sua saúde, ou também, nas atividades onde a sensibilidade e a habilidade humana é insuficiente para a sua execução, tal como no caso de um robô que realiza a soldagem dos fios condutores que interligam os terminais externos do encapsulamento com o circuito integrado propriamente dito, conhecido como *bonding robot*.

A utilização dos robôs, incorporados à automação industrial, no processo de fabricação de produtos nas indústrias, tem beneficiado os consumidores tanto pela redução do custo de produção e conseqüente barateamento do preço final quanto pela maior confiabilidade do produto.

A automação industrial e a robótica no Brasil não teve

a evolução esperada nesses últimos anos. Podem ser apontadas as mais variadas razões para isso, desde a instabilidade política e econômica no Brasil até a crise financeira que o mundo está atravessando nesses últimos anos. Entretanto, não se pode ignorar que uma das razões é a falta de pesquisa aplicada nesta área nas Escolas de Engenharia e nos Centros de Pesquisa do Brasil. Por outro lado, dificilmente são divulgados os aspectos práticos e detalhados das técnicas envolvidas na robótica, pois de um lado, as indústrias detentoras dessas técnicas não se interessam em divulgá-las, para manter os seus segredos industriais, e de outro lado, as Universidades têm dificuldades em realizar as pesquisas aplicadas, principalmente pela falta de recursos financeiros, dificultando assim o desenvolvimento dos robôs nacionais pelas indústrias brasileiras.

Um fato mais grave é que a falta de recursos financeiros nas Universidades dificulta a formação de recursos humanos altamente qualificados, não somente em robótica mas também em mecatrônica, que é o conhecimento fundamental para a automação industrial.

Baseado nesse ponto de vista, o presente trabalho tem como objetivo desenvolver um sistema didático para mecatrônica de alto nível, composto de um robô, um sensor de visão e uma garra com diversos sensores. A opção pelo desenvolvimento de um sistema didático deve-se à necessidade urgente de integrar o sistema educacional brasileiro à realidade tecnológica que surge com a terceira revolução industrial, seguindo uma tendência mundial de aproximar a educação universitária à filosofia industrial do terceiro milênio, na área de mecatrônica.

Um outro objetivo é o desenvolvimento da tecnologia e competência nacional em robótica para que possamos dar suporte técnico aos fabricantes nacionais e usuários finais de robôs que se supõe que num futuro próximo deverão iniciar este tipo de atividades no Brasil.

O projeto referente a este trabalho foi desenvolvido no Departamento de Sistemas e Controle de Energia - Laboratório de Robótica, durante o estágio em Campinas para realização dos Cursos de Pós-Graduação e numa segunda etapa, a construção de equipamento, foi concluída na cidade de Belém, Universidade Federal do Pará, sob a concordância e acompanhamento periódico do orientador, Prof. Dr. Álvaro Geraldo Badan Palhares, por acreditar na necessidade de formação de pessoal qualificada nesta região na área de automação industrial e robótica, para atender as crescentes necessidades de mão de obra qualificada nas indústrias da região norte, principalmente nas indústrias instaladas no parque industrial da Zona Franca de Manaus onde algumas indústrias já são usuárias de robôs.

## 1.2 SISTEMAS MECATRÔNICOS.

Mecatrônica é o paradigma tecnológico que tornou possível a evolução da automação dos sistemas, centrada no controle de movimentos das máquinas e é composta de sub-áreas tais como: engenharia mecânica, engenharia elétrica, engenharia de controle e computação. Na mecatrônica, a evolução tecnológica tem modificado a forma de concepção dos sistemas de acionamento e sensoriamento, permitindo que mais informações sejam incorporadas aos algoritmos de controle e que as estruturas mecânicas tenham seus modelos dinâmicos mais simples, proporcionando uma forma mais inteligente de controle.

Com relação aos sistemas de acionamento, um grande salto tecnológico foi dado, caracterizado pela viabilidade dos ímãs permanentes de terras raras, que permite projetos e implementações de motores de baixo peso e alto torque; assim, pode-se constatar que houve uma tendência a se substituir os motores eletrohidráulicos por motores eletromecânicos nas faixas de potências acima das que anteriormente era mais vantajoso, do ponto de vista econômico e técnico, quando feito por motores eletrohidráulicos.

Os elementos de controle, também foram, ao longo do tempo, substituídos por sistemas eletrônicos.

Na realidade dos dias atuais, é grande a quantidade dos sensores eletrônicos incorporados nesses atuadores, onde as mais diversas razões são citadas como responsáveis por essas contínuas mudanças de tendências, entre elas:

1. Os sinais elétricos são mais fáceis de serem tratados em transmissão, amplificação, distribuição, monitoração e registro;
2. Possibilidade de compactação;
3. Facilidade na instalação e interligação;
4. Os sistemas elétricos e eletrônicos são superiores em confiabilidade, manutenção e inferiores em custo;
5. Programabilidade.

As razões (1) a (4) surgem como consequência direta da evolução do *hardware* e a (5) como evolução do *software*, ou seja, flexibilidade na programação através dos computadores. Portanto, a técnica da mecatrônica de alto nível na atualidade é a utilização dos computadores na implementação desses sistemas. Inicialmente, os computadores foram utilizados no nível supervisor dos processos produtivos com a função apenas de monitorar os estados das máquinas. Com o surgimento dos minicomputadores, múltiplos processos passaram a ser controlados por um minicomputador que fazia parte do sistema de controle.

Com o surgimento do microprocessador, os anseios tecnológicos foram bastante saciados uma vez que os microprocessadores forneceram a programabilidade tão desejada pelos engenheiros nos elementos de controle, possibilitando a criação de elementos inteligentes. Este processo é aqui denominado de *inteligentização* dos elementos do sistema. Um bom exemplo disso pode ser visualizado nos atuais robôs, quando neles são incorporados sensores inteligentes, tais como, sensor de visão e controlador inteligente, fazendo com que eles possam tomar decisões que eram impossíveis aos seus antecessores.

Assim, a mecatrônica evoluiu com a *inteligentização* das máquinas.

#### 1.2.1 CONTROLE EM UM SISTEMA MECATRÔNICO.

Como o objetivo de um sistema mecatrônico é a realização das tarefas através de um sistema mecânico, é natural que o controle do sistema mecatrônico seja relacionado ao controle de movimento do sistema mecânico. Existem dois tipos distintos de controle num sistema mecatrônico, estruturados nas diferentes hierarquias: Um deles, denominado de controle de movimento, que é responsável pelo controle de posição, velocidade, aceleração ou força do sistema mecânico e um outro, denominado de controle de seqüenciamento ou controle seqüencial, que é responsável pelo seqüenciamento dos movimentos. Uma outra função do controle seqüencial é monitorar os estados dos equipamentos periféricos e enviar comandos para iniciar as tarefas desses equipamentos para que se possa sincronizar o movimento do sistema mecânico com os equipamentos periféricos.

Esta distinção do tipo de controle, estruturado em diferentes níveis hierárquicos, tem como principal objetivo, aumentar a operacionalidade de uso do sistema mecatrônico, para que o usuário final possa programar o seu sistema para execução das tarefas, sem necessitar do conhecimento especializado da teoria de controle, necessitando apenas do conhecimento sobre a tarefa e as características geométricas e cinemáticas do mecanismo.

### 1.3 DEFINIÇÃO E TIPOS DE SENSORES PARA ROBÔS

Como o robô industrial é um sistema eletromecânico de múltiplos elementos em movimentos relativos cujos movimentos são controlados por computador, é necessário que haja sensores para que se possa realizar o controle desses movimentos, tanto do ponto de vista cinemático como dinâmico, além da necessidade de verificação das características técnicas.

Os primeiros robôs industriais eram simplesmente caracterizados por um sistema mecânico com diversos graus de liberdade sendo cada grau de liberdade servocontrolado independentemente ou seja, cada grau de liberdade atingia o seu *set point* de forma independente e a composição desses diversos graus de liberdade fazia com que a sua extremidade alcançasse uma determinada posição no espaço tridimensional.

Desta forma, os sensores para robôs projetados naquela época tinham a função apenas de informar ao sistema de controle a situação (por exemplo, posição e velocidade) atual de cada grau de liberdade para que o sistema de controle pudesse decidir uma ação sobre aquele grau de liberdade conforme os *set point* previamente definidos. A esta classe de sensores denomina-se de sensores internos.

Com o sucesso da utilização dos robôs na automação industrial nas tarefas simples e repetitivas, tais como soldagem a ponto das chapas metálicas nas indústrias automobilísticas, onde até hoje o robô industrial demonstra uma grande eficiência, não apenas em redução do custo de fabricação mas também na qualidade do produto, a utilização dos robôs industriais foi sendo cada vez mais intensa nas diversas áreas da automação. Com esta difusão, a própria característica desses foi evoluindo conforme a necessidade da automação industrial, sendo que a mais significativa é a integração dos sensores com a função de monitorar o ambiente onde o robô executa a sua tarefa ou monitorar o grau de interação entre o robô e o seu ambiente. A esta classe de sensores denomina-se sensores externos.

Os sensores internos utilizados nos robôs industriais são basicamente os sensores de posição, velocidade e força. Os sensores de posição podem ser o tradicional potenciômetro, codificador angular, *syncro resolver*, régua óptica e escala magnética linear. O gerador tacométrico ou tacogerador ainda é um elemento muito utilizado como sensor de velocidade embora hoje este dispositivo esteja sendo gradualmente substituído por

codificador angular óptico devido ao seu alto custo de aquisição assim como o de manutenção.

Os sensores externos estão em fase de intensa evolução e a ampliação da gama de utilização do robô está diretamente dependente da existência de determinados tipos de sensores externos. De um outro ponto de vista, cada vez que surge a necessidade de automatizar um determinado processo industrial há a evolução de novos tipos de sensores externos.

Um dos primeiros sensores externos desenvolvido foi o sensor de presença de peça instalado na garra do robô para que este possa detectar a ausência de objeto que deveria estar num determinado local pré-estabelecido. Um outro sensor externo utilizado no robô foi o sensor de força instalado no braço para evitar que o robô tente carregar um objeto de peso excessivo que possa causar dano na estrutura do braço ou nos motores que acionam o braço.

Um sensor externo que surgiu com as mais recentes utilizações dos robôs é o sensor de força de agarramento instalado na garra ou, mais precisamente, nos dedos da garra para possibilitar o agarramento de um objeto frágil [21].

Na fase inicial da sua utilização, os usuários acreditavam que os robôs industriais fossem equipamentos flexíveis de múltiplas utilidades. Em parte, isso era verdade pois um determinado robô pode ser utilizado nas diversas tarefas. Entretanto, um robô, uma vez instalado, por exemplo para transportar um objeto de um ponto ao outro, torna-se necessário valer-se de um mecanismo auxiliar adicional (conhecido como *loader*) para posicionar e orientar aquele objeto sempre numa posição e orientação definidas. Em outras palavras, o robô instalado para executar uma tarefa já não era tão flexível pois cada mudança na tarefa implicaria na mudança do *loader*.

A utilização de sensores de visão vem contribuir para resolver, em parte, este problema pois com o seu auxílio o robô pode procurar a peça, desde que ela se encontre dentro do

campo visual do sensor. Isto significa que desde que o robô esteja equipado com sensor de visão o *loader* pode ser algo menos rigoroso e assim, o mesmo *loader* pode ser utilizado para diversas classes de objetos. Se o sensor de visão estiver equipado com a capacidade de reconhecer e distinguir a forma física dos objetos, a flexibilidade do robô aumenta consideravelmente, pois o próprio robô poderá decidir os tipos de tarefas que ele deverá executar em função do objeto.

A integração de sensores na garra permite agarramento mais confiável de objetos, possibilitando a manipulação de objetos frágeis e ampliando a gama de aplicação do robô.

#### 1.4 ESTRUTURA DO SISTEMA DIDÁTICO

Com o objetivo de atender os principais requisitos de um sistema mecatrônico de alto nível, uso de sensores inteligentes e efetuação de movimentos combinados, foi desenvolvido um sistema didático de controle baseado em órgãos sensoriais, ou sensores externos composto de: um sistema de braço mecânico composto de parte mecânica e sistema de controle, uma garra inteligente com sensor de abertura dos dedos, sensor de deslizamento e sensor de força de agarramento, um sensor de visão, linguagem de programação para robô, software de controle do robô, controlador da garra e software de controle baseado nos órgãos sensoriais.

O sistema didático desenvolvido no presente trabalho é suficiente para verificar na prática, os seguintes aspectos técnicos da mecatrônica:

1. Informatização do sistema mecânico, tornando-o mais inteligente;
2. Controle de movimento do sistema mecânico através do sistema de controle hierárquico baseado em microcomputador;
3. O efeito da integração de diversos sistemas autônomos através da integração do robô com sensor de visão;
4. O efeito do algoritmo de controle no movimento do sistema

mecânico;

5. A importância da utilização de sensores externos;
6. A importância da programabilidade do sistema através da linguagem de programação do robô.

### 1.5 ESTRUTURA DA TESE.

No capítulo 2 são descritas a parte mecânica do robô articulado desenvolvido neste trabalho e os tratamentos necessários para o seu controle;

No capítulo 3 são descritos o desenvolvimento de uma garra de dois dedos com integração de sensor de abertura, sensor de força e sensor de deslizamento;

No capítulo 4 são descritos o desenvolvimento de um sensor de visão com capacidade para calcular coordenadas do centro do objeto, a orientação do mesmo e a identificação;

No capítulo 5 são descritos o *hardware* e *software* do sistema de controle hierárquico para este robô;

No capítulo 6 são descritas as conclusões.

No apêndice A é descrito o *hardware* do sensor de visão.

No apêndice B é descrito o *hardware* do controlador do robô.

No apêndice C se encontram as fotografias do sistema desenvolvido.

## CAPÍTULO 2

# ESTRUTURA MECÂNICA DE POSICIONAMENTO E ORIENTAÇÃO

### 2.1 INTRODUÇÃO.

A estrutura mecânica de um robô tem a função de posicionar e orientar o elemento final de ação no espaço tridimensional para execução de tarefas. A quantidade e o tipo de grau de liberdade, rotativo ou deslizante, denominada de junta, que compõe a estrutura mecânica estão relacionados com a natureza da tarefa que o robô deve executar.

Se um robô é projetado, por exemplo, para tarefa de pintura de uma superfície plana, é necessário somente duas juntas deslizantes em direções ortogonais entre si, com a pistola injetora de tinta direcionada na direção ortogonal ao plano formado por essas juntas. Naturalmente, a superfície plana a ser pintada deve estar paralela ao plano formado por essas juntas. Se a superfície a ser pintada tiver uma curvatura, a pistola injetora de tinta deve ser posicionada no espaço tridimensional para manter sempre a mesma distância com a superfície e também precisa ser orientada para que tenha sempre a direção ortogonal àquela superfície. Se a curvatura da superfície a ser pintada for cilíndrica, o mecanismo de orientação é de apenas um grau de liberdade rotativa desde que esta superfície esteja posicionada adequadamente em relação à estrutura mecânica mas se a curvatura da superfície for esférica o mecanismo de orientação precisa ter duas juntas para manter a pistola apontada sempre na direção ortogonal à superfície.

Assim pode-se afirmar que a quantidade e a natureza das

juntas de um robô depende da tarefa a ser executada e também do posicionamento e da orientação do objeto em relação ao robô.

Por outro lado, muitas vezes o robô trabalha num ambiente com a presença de obstáculos dentro do volume de trabalho tais como no caso do robô de solda a ponto utilizado na soldagem da armação da porta ou da janela de um automóvel, ou no caso do robô que verifica as falhas da parte interna de uma tubulação sinuosa. Nestas aplicações o projeto da estrutura mecânica do robô deve levar em consideração a presença desses obstáculos.

De um modo geral, três graus de liberdade para posicionamento e três graus de liberdade para orientação, na estrutura mecânica do robô, é suficiente para posicionar e orientar aleatoriamente o elemento final de ação no espaço tridimensional.

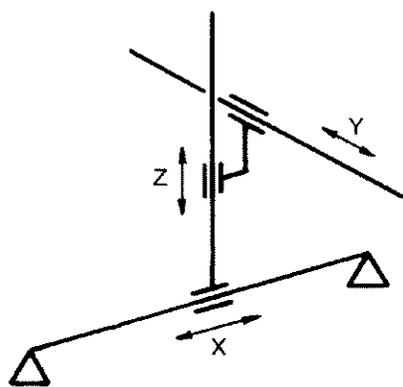
Existem algumas variedades de mecanismos de posicionamento que basicamente podem ser classificados como mecanismos cartesianos, polares ou cilíndricos, cujas juntas têm os movimentos segundo o sistema de coordenadas de mesmo nome, e mecanismos articulados, inspirados na geometria do braço humano.

A figura [2.1] ilustra esquematicamente esses quatro tipos básicos de mecanismo para posicionamento do elemento final de ação de um robô.

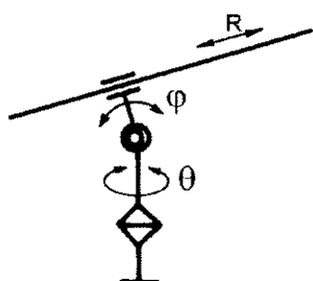
O mecanismo de orientação do elemento final de ação, normalmente denominado de punho, é composto de juntas rotativas, como são ilustrados na figura [2.2]. A denominação dessas três juntas são *YAW*, *PITCH* e *ROLL* como são ilustradas nesta figura.

## 2.2 DESCRIÇÃO GLOBAL.

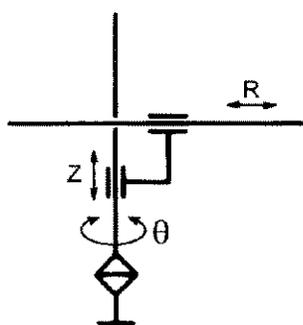
O robô industrial desenvolvido no presente estudo é composto de cinco graus de liberdade sendo uma base com movimento deslizando de 1.200 mm sobre a qual está colocado o braço articulado com articulação do ombro e do cotovelo. O



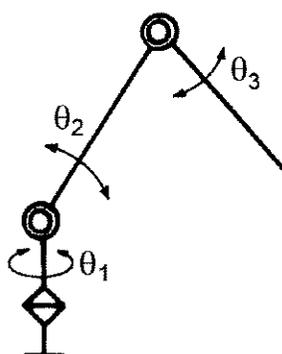
ROBÔ CARTESIANO



ROBÔ ESFÉRICO



ROBÔ CILINDRICO



ROBÔ ARTICULADO

FIG. 2.1 ESTRUTURAS BÁSICAS DOS ROBÔS

braço tem comprimento de 400 mm e o ante-braço de 600 mm. Na extremidade deste braço está situado o mecanismo de punho de dois graus de liberdade responsáveis pela orientação e rotação da garra. Não foi implementada a junta YAW por limitação financeira e tecnológica. As juntas *PITCH* e *ROLL* serão aqui denominadas de mergulho do punho e rotação da garra respectivamente. A figura [2.3] ilustra a composição dos graus de liberdade de movimentos de forma esquemática.

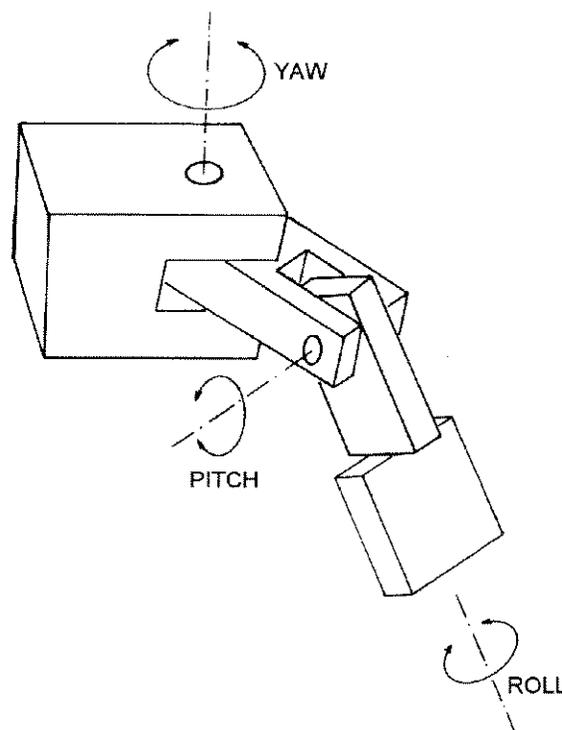


FIG. 2.2 GRAU DE LIBERDADE PARA ORIENTAÇÃO

Todos os cinco graus de liberdade e o movimento de abre-e-fecha da garra são acionados por servomotores de corrente contínua. Estão instalados sensores de posição e de velocidade nos três graus de liberdade de posicionamento (da base, do

ombro e do cotovelo) e sensores de posição nos dois graus de liberdade do punho.

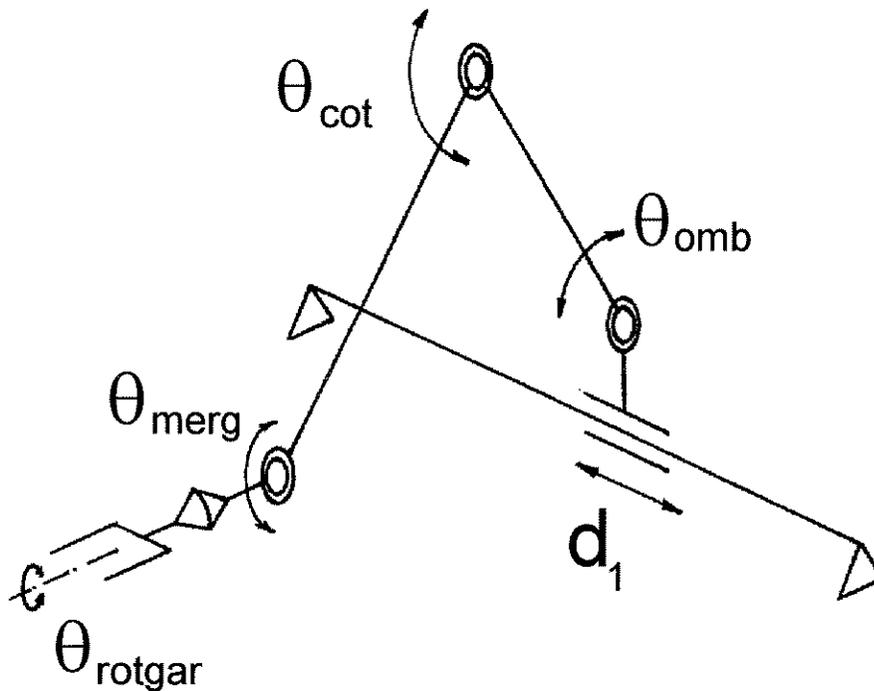


FIG. 2.3 COMPOSIÇÃO DOS GRAUS DE LIBERDADE

A tabela 1.1 mostra a especificação técnica deste robô.

NOME DA JUNTA	TIPO	VARIAÇÃO MÁXIMA	VELOCIDADE MÁXIMA	ACELERAÇÃO MÁXIMA
BASE DESLIZANTE	PRISMÁTICA	1,2 m	0,635 m/s	1,27m/s <sup>2</sup>
OMBRO	ROTATIVA	95°	100°/s	100°/s <sup>2</sup>
COTOVELO	ROTATIVA	80°	100°/s	100°/s <sup>2</sup>
MERGULHO DA GARRA	ROTATIVA	270°	100°/s	100°/s <sup>2</sup>
ROTAÇÃO DA GARRA	ROTATIVA	360°	100°/s	100°/s <sup>2</sup>
CAPACIDADE DE GARGA NA PONTA DO ROBÔ			4,9 N	

TABELA 1.1 - ESPECIFICAÇÃO TÉCNICA DO ROBÔ

Este conjunto é controlado por um microcomputador PC/AT-386, com microprocessador 80386 e um coprocessador 80387, *clock* de 40 MHz, Disco Rígido 110 MByte, 4 MByte de RAM e um acionador de disco flexível de 1,44 MByte.

O interfaceamento entre o sistema mecânico e o controlador é realizado através de 2 placas que totaliza 6 canais de conversor digital analógico, 5 canais de contador *UP/DOWN* e entradas digitais e um canal de comunicação paralela com o sensor de visão.

O software de controle do robô é composto de diversos módulos:

1. Programação de tarefas;
2. Controle supervisor;
3. Controle de movimentos.

Este robô, assim composto, tem a capacidade de realizar o movimento, posicionando e orientando a garra nas posições previamente programadas, agarrando e transportando objeto de uma posição para outra.

A seguir serão descritos detalhes técnicos do mecanismo e o tratamento necessário para o controle de movimentos.

## 2.3 DESCRIÇÃO DETALHADA.

### 2.3.1 Base Linear.

É acionada por servomotores de corrente contínua de 110 watts/70 volts, e torque de pico de 3,626 N.m com sensor de posição (codificador angular) e sensor de velocidade (tacogerador) incorporados, sistema de fuso-castanha e guias lineares (Figura [2.4]). Este servomotor é acionado por acionador eletrônico a PWM com regulador de velocidade.

Como o passo do fuso é de 12,7 mm e o codificador angular emite 500 pulsos/volta que são multiplicados eletronicamente por 4, o deslocamento linear da base correspondente a cada pulso emitido por codificador angular é de 0,00635 mm.

A relação entre o valor digital da entrada do conversor

D/A e a velocidade de deslocamento da base é determinada pelo ganho do conversor D/A e a característica do acionador eletrônico do servomotor. O ganho do conversor D/A é de 9 volts/511 e a característica do acionador é de 1000 RPM/3 volts. Sabendo que o fuso tem passo 12,7 mm a relação entrada do D/A versus velocidade de deslocamento é de:

$$V_x [\text{mm/s}] = 1,242661466 \cdot \text{valorDA} \quad (2.1)$$

onde: valorDA é o valor numérico na entrada do conversor D/A.

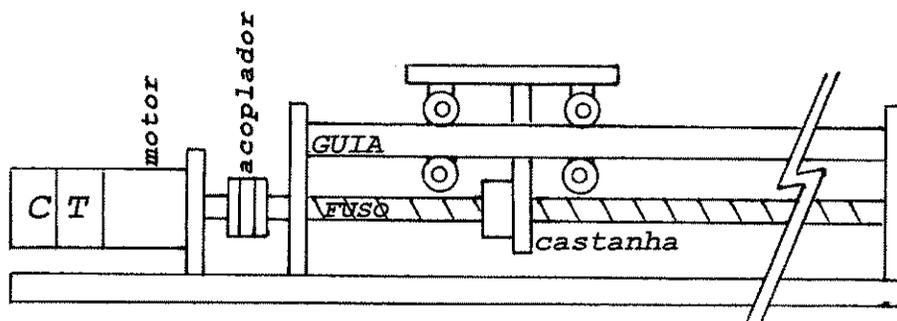


FIG. 2.4 BASE DESLIZANTE ONDE: C É O CODIFICADOR ANGULAR E T O TACOGERADOR

### 2.3.2 - Juntas do Ombro e do Cotovelo.

A junta do ombro é composta de servomotor e acionador com as mesmas características do sistema da base, conjunto fuso-castanha e *link* para transformar o movimento linear em movimento rotativo. O princípio de funcionamento e suas dimensões estão ilustrados na figura [2.5].

A composição e o princípio de funcionamento da junta do cotovelo é semelhante a do ombro, diferindo apenas pelo fato de que o antebraço é colocado em paralelo com o *link*. (Figura [2.6]).

O conjunto braço e antebraço compõe um mecanismo diferencial com a estrutura de paralelogramo, de tal forma que ao acionar somente a junta do ombro o antebraço mantém o mesmo ângulo em relação ao sistema de coordenadas da base ou universal (figura [2.7]).

A utilização do conjunto fuso-castanha e *link* como o mecanismo de redução da rotação do eixo do motor assegura a robustez mecânica do braço e do antebraço. Entretanto como os sensores de posição e de velocidade estão instalados no eixo do motor e não diretamente nas articulações, é necessário realizar um cálculo a partir da leitura dos sensores, para que o computador conheça os ângulos reais das articulações. Este cálculo é realizado pelo próprio computador de controle e o procedimento leva em conta que na posição inicial, a situação da figura [2.5], a contagem de pulsos é inicializada e o ângulo do ombro é de 90 graus. Nesta situação a distância entre os pontos B e C é  $L_0 = 111,8033989$  mm e o ângulo  $\alpha_0 = 42,51045$  graus (valores calculados). A variação da distância BC provocada pelo acionamento do fuso é calculada a partir da contagem de pulsos emitidos por codificador angular, sabendo que o passo deste fuso é de 5 mm e que o codificador angular emite 2000 pulsos/volta. A distancia BC é calculada pela equação:

$$L = L_0 + K_F CP \quad (2.3)$$

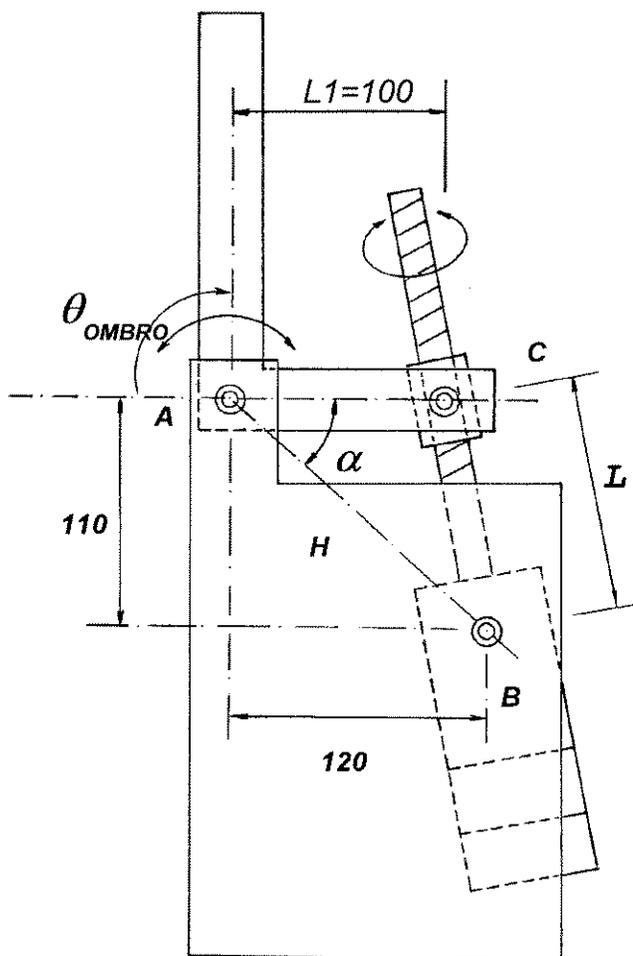


FIG. 2.5 MECANISMO DO OMBRO

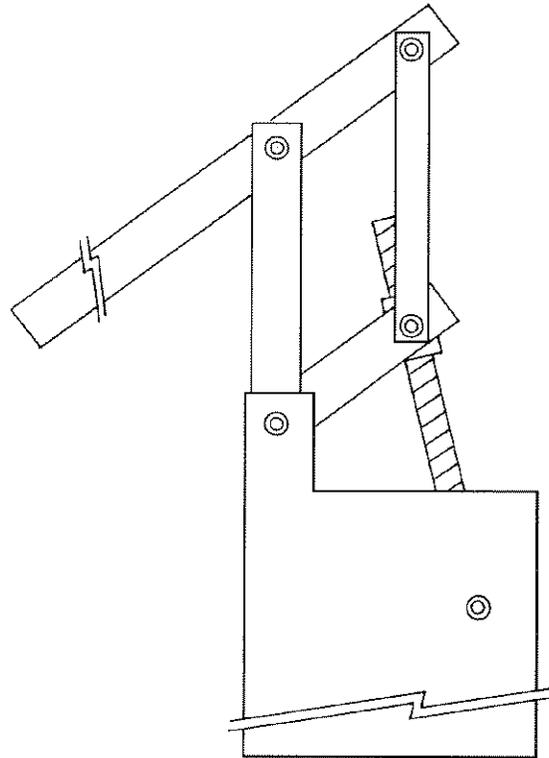


FIG. 2.6 MECANISMO DO COTOVELO

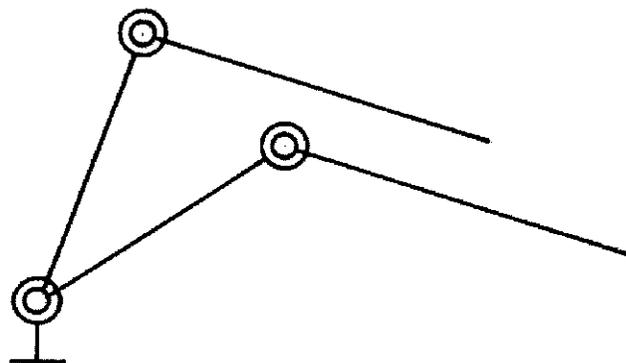


FIG. 2.7 MOVIMENTOS DIFERENCIAIS

onde;

$K_f$ : Variação de  $L$  correspondente a 1 pulso ( $K_f = 0,0025$ );

$CP$ : Quantidade de pulsos emitidos desde a posição inicial até a posição atual.

Feito o cálculo do  $L$ , calcula-se o valor do ângulo através da equação (2.4).

$$L^2 = L_1^2 + H^2 - 2 \cdot H \cdot L_1 \cos \alpha \quad (2.4)$$

Como  $L_1 = 100$  e  $H = 162,788206$ , tem-se:

$$\alpha = \arccos \left( \frac{36500 - L^2}{32557,6412} \right) \quad (2.5)$$

Finalmente o ângulo da junta do ombro correspondente à quantidade  $CP$  de pulsos é dado pela equação (2.6) em graus ou pela equação (2.7) em radianos.

$$\theta_{\text{OMBRO}} = 132,5102719 - \alpha_{\text{OMBRO}} \quad [\text{graus}] \quad (2.6)$$

ou

$$\theta_{\text{OMBRO}} = 2,312740538 - \alpha_{\text{OMBRO}} \quad [\text{radiano}] \quad (2.7)$$

E pelo mesmo procedimento, para o cotovelo tem-se

$$\theta_{\text{COTOVELO}} = 42,510444708 - \alpha_{\text{COTOVELO}} \quad [\text{graus}] \quad (2.8)$$

ou

$$\theta_{\text{COTOVELO}} = 0,741947268 - \alpha_{\text{COTOVELO}} \quad [\text{radiano}] \quad (2.9)$$

O procedimento para calcular os valores das quantidades de pulsos emitidos pelos codificadores angulares para alcançar determinados ângulos de ombro e de cotovelo é o seguinte:

Pelas equações (2.4) e (2.7) tem-se:

$$L = \sqrt{36500 - 32557,6412 \cdot \cos \alpha_{\text{OMBRO}}} \quad (2.10)$$

Calculado L sabe-se pela equação (2.3) que:

$$L = 111,8038989 + CP \cdot K_F \quad (2.11)$$

Logo, a quantidade de pulsos necessários para atingir o ângulo desejado será:

$$CP = \frac{L - 111,8033989}{K_F} \quad (2.12)$$

É importante observar que no presente mecanismo, a relação velocidade angular do eixo do motor e a velocidade angular das juntas, tanto do ombro como do cotovelo, não são constantes e são dependentes dos ângulos atuais dessas juntas.

O procedimento para calcular o valor numérico a ser colocado no conversor D/A para obter a velocidade angular desejada da junta em rad/seg é dado a seguir.

O conjunto acionador eletrônico, servomotor e tacogerador, assim como o conversor D/A tem as mesmas características do conjunto utilizado para a base. Como o fuso tem o passo 5 mm, a velocidade da variação da distância L é:

$$\frac{dL}{dt} = \frac{50}{511} \cdot 5 \cdot \text{valorDA} \quad (2.13)$$

$$\text{valorDA} = 2,044 \cdot \frac{dL}{dt}$$

Pela geometria do mecanismo da figura [2.5] tem-se:

$$L^2 = L_1^2 + H^2 - 2 \cdot H \cdot L_1 \cos \alpha \quad (2.14)$$

$$\frac{dL}{dt} = \frac{\text{sen} \alpha \cdot H \cdot L_1}{\sqrt{L_1^2 + H^2 - 2 \cdot H \cdot L_1 \cdot \cos \alpha}} \cdot \frac{d\alpha}{dt}$$

$$\frac{d\alpha}{dt} = \frac{d\theta}{dt} \quad (2.15)$$

Portanto:

$$\text{valorDA} = 2,044 \cdot \frac{\text{sen} \alpha \cdot H \cdot L_1}{\sqrt{L_1^2 + H^2 - 2 \cdot H \cdot L_1 \cdot \cos \alpha}} \cdot \frac{d\theta}{dt} \quad (2.16)$$

### 2.3.3 Juntas do Punho

O mecanismo do punho é ilustrado na figura [2.8]. Nesta figura:

RD - Redutor harmônico com a taxa de redução 1:88;

$M_4$  e  $M_5$  - Servomotores de 16 Watts/32 Volts, torque de pico de 0,2254 N.m, velocidade máxima de 4000 rpm, corrente máxima de 1 A;

$COD_4$  e  $COD_5$  - Codificadores angulares de 100 pulsos/volta;

$CA_4$  e  $CA_5$  - Contadores;

$\theta_{ROTGR}$  - Ângulo de rotação da garra;

$\theta_4$  e  $\theta_5$  - Ângulo de saída dos redutores.

A relação individual  $\frac{\theta_4}{CA_4}$  e  $\frac{\theta_5}{CA_5}$  é:

$$\theta_4 = -CA_4 \times 0,00008925 \quad [\text{radiano}] \quad (2.17)$$

$$\theta_5 = -CA_5 \times 0,0001785 \quad [\text{radiano}] \quad (2.18)$$

O ângulo  $\theta_{ROTGR}$  é uma função de  $\theta_4$  e  $\theta_5$  devido ao mecanismo diferencial. A função que calcula  $\theta_{ROTGR}$  é:

$$\theta_{ROTGR} = \frac{\theta_5 - \theta_4}{2} \quad (2.19)$$

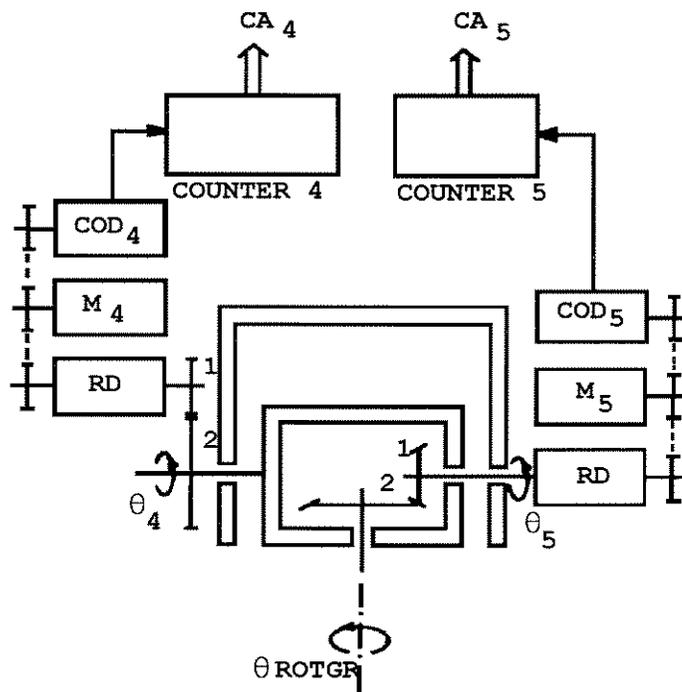


FIG 2.8 MECANISMO DO PUNHO

## 2.4 SISTEMA DE COORDENADAS E ÂNGULOS DAS ARTICULAÇÕES.

O sistema de coordenadas do robô e o ângulo das articulações em relação ao sistema de coordenadas universal são ilustrados na figura [2.9]. Os ângulos das articulações do ombro, do cotovelo e do mergulho da garra tem os seus valores nulos quando estão na posição horizontal. Esses ângulos aumentam quando se movimentam no sentido anti-horário visto pelo lado positivo do eixo X ou visto pela frente, no caso da garra. (figura [2.10]).

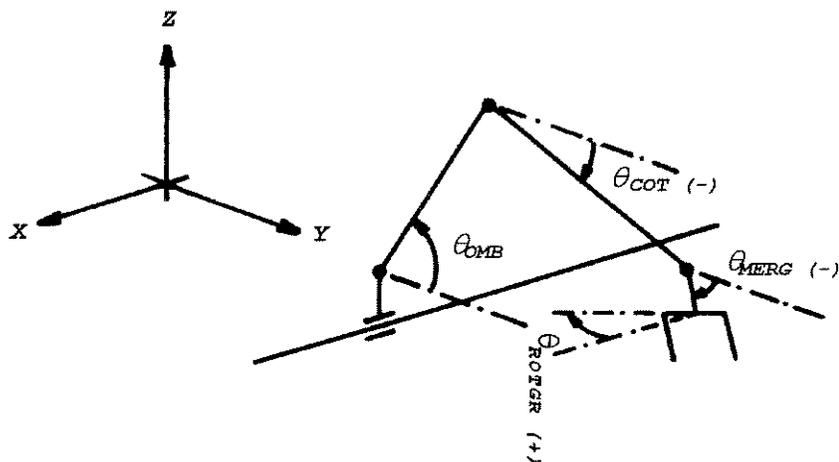


FIG. 2.9 SISTEMA DE COORDENADAS DO BRAÇO

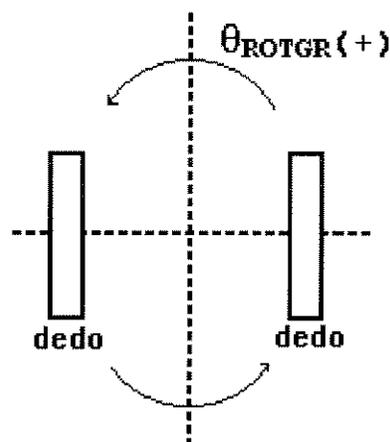


FIG. 2.10 DEFINIÇÃO DO ÂNGULO DE ROTAÇÃO DA GARRA

## 2.5. TRANSFORMAÇÕES CINEMÁTICAS.

A transformação cinemática direta determina a posição da ponta do braço em coordenadas cartesianas da base, à partir das posições das juntas e a transformação cinemática inversa determina a posição das juntas a partir das coordenadas cartesianas da ponta do braço.

A transformação cinemática direta é utilizada, por exemplo, no caso do controle de trajetória da ponta do braço para verificar se a ponta do braço está rastreando corretamente a trajetória definida segundo o sistema de coordenadas da base.

A transformação cinemática inversa é necessária em algumas aplicações do robô, tal como no caso do agarramento de um objeto cuja posição é detectada por sensor de visão.

### 2.5.1 Transformação Cinemática Direta

As equações que calculam as coordenadas da ponta do braço e do centro da garra a partir das posições das juntas são extraídas da figura [2.11].

$$Y_P = l_2 \cos \theta_{OMBRO} + l_3 \cos \theta_{COTOVELC} \quad (2.20)$$

$$Z_P = l_2 \sin \theta_{OMBRO} + l_3 \sin \theta_{COTOVELC} \quad (2.21)$$

$$Y_G = Y_P + l_G \cos \theta_{MERGULHO} \quad (2.22)$$

$$Z_G = Z_P + l_G \sin \theta_{MERGULHO} \quad (2.23)$$

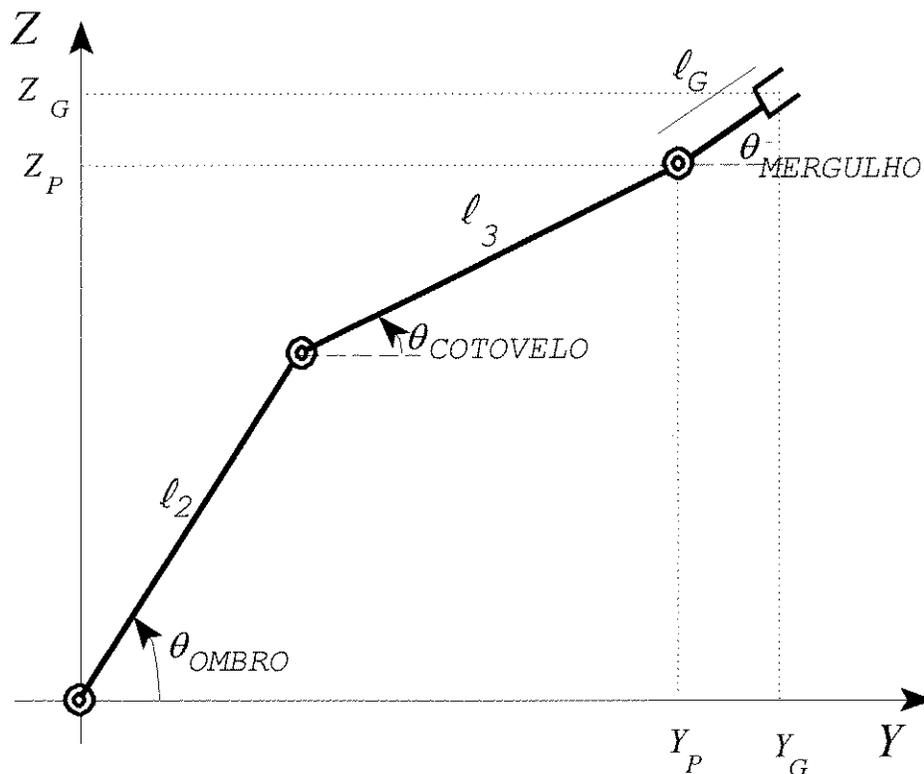


FIG.2.11 TRANSFORMADA CINEMÁTICA DIRETA

A coordenada  $X$  da ponta do braço é o próprio deslocamento da base linear e o ângulo de rotação da garra não afeta as coordenadas do centro da garra.

### 2.5.2 Transformação Cinemática Inversa

No caso de um braço com duas articulações, a transformação cinemática inversa geralmente tem duas soluções, como são ilustradas na figura [2.12], como cotovelo acima e cotovelo abaixo. Entretanto, no presente caso é impossível o braço assumir a cotovelo abaixo devido a própria característica do mecanismo em paralelogramo. A determinação dos ângulos das articulações é feita da seguinte maneira:

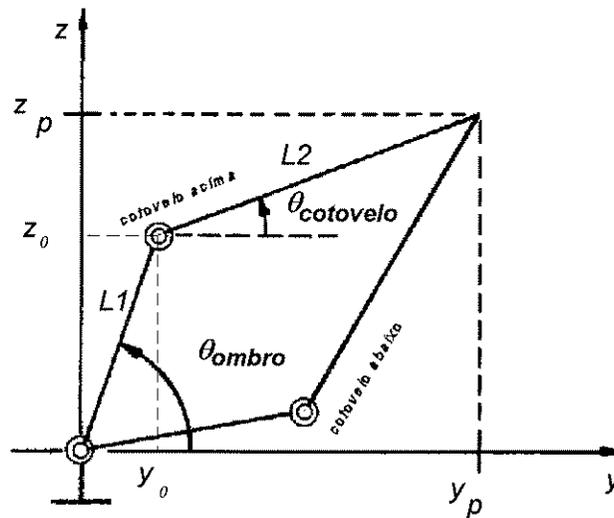


FIG. 2.12 TRANSFORMAÇÃO CINEMÁTICA INVERSA

$$L_1^2 = Y_0^2 + Z_0^2 \quad (2.24)$$

$$L_2^2 = (Y_0 - Y_P)^2 + (Z_0 - Z_P)^2 \quad (2.25)$$

Resolvendo este sistema de equações, tem-se:

$$-(Y_P^2 + Z_P^2) \cdot Z^2 + (2 \cdot w \cdot Z_P) \cdot Z + (L_1^2 Y_P^2 - w^2) = 0 \quad (2.26)$$

onde:

$$w = \frac{L_1^2 - L_2^2 + Y_P^2 + Z_P^2}{2} \quad (2.27)$$

Denominando de  $a$ ,  $b$ ,  $c$  e  $\Delta$ , os seguintes fatores:

$$a = -(Y_P^2 + Z_P^2) \quad (2.28)$$

$$b = 2 \cdot w \cdot Z_P \quad (2.29)$$

$$c = L_1^2 \cdot Y_P - w^2 \quad (2.30)$$

$$\Delta = b^2 - 4 \cdot a \cdot c \quad (2.31)$$

tem-se:

$$z_{01} = \frac{-b + \sqrt{\Delta}}{2a} \quad (2.32)$$

$$z_{02} = \frac{-b - \sqrt{\Delta}}{2a} \quad (2.33)$$

$$Y_{01} = \frac{w - z_1 \cdot Z_P}{Y_P} \quad (2.34)$$

$$Y_{02} = \frac{w - z_2 \cdot Z_P}{Y_P} \quad (2.35)$$

A decisão de qual das soluções corresponde às coordenadas  $(Y, Z)$  da junta do cotovelo é feita baseada no fato de que o braço só pode assumir a situação cotovelo acima, na figura [2.12]. Como o cotovelo acima é caracterizada pelo ângulo do ombro maior que o do cotovelo abaixo, calcula-se o ângulo do ombro fazendo:

$$\theta_{OMBRO1} = \arctg \left( \frac{z_{01}}{Y_{01}} \right) \quad (2.36)$$

$$\theta_{OMBRO2} = \arctg \left( \frac{z_{02}}{Y_{02}} \right) \quad (2.37)$$

e o que tiver maior valor corresponde ao ângulo do ombro da configuração cotovelo acima.

Após a determinação das coordenadas da junta do cotovelo -  $(y_1, z_1)$  ou  $(y_2, z_2)$  - calcula-se o ângulo da junta do cotovelo fazendo:

$$\theta_{COTOVELO} = \arctg \left( \frac{z_P - z_i}{Y_P - Y_i} \right) \quad (2.38)$$

onde:  $i=1$  ou  $2$  dependendo do ângulo que corresponde a configuração.

Obs.: Sempre que determina arco tangente, se a função utilizada na programação não analisa o quadrante do argumento, é preciso verificar o quadrante através da análise do sinal do numerador e do denominador do argumento e quando o denominador for muito próximo a zero, o valor do ângulo é determinado sem se utilizar a equação do arco tangente.

A solução citada até a equação (2.37) é genérica para qualquer braço articulado de dois graus de liberdade. Entretanto, quando está claro, como no presente caso, que o braço só pode estar em cotovelo acima, existe uma solução mais direta.

A partir do triângulo ABC da figura [2.13], pode-se

extrair a seguinte equação:

$$Y_P^2 + Z_P^2 = l_2^2 + l_3^2 + 2 \cdot l_2 \cdot l_3 \cdot \cos(180 - \phi)$$

$$\phi = \arccos \left( \frac{Y_P^2 + Z_P^2 - l_2^2 - l_3^2}{2 \cdot l_2 \cdot l_3} \right) \quad (2.39)$$

A partir do triângulo retângulo ADC pode-se extrair a seguinte equação:

$$\alpha = \arctg \left( \frac{l_3 \sin \phi}{l_2 + l_3 \cos \phi} \right) \quad (2.40)$$

Observando a figura [2.13], nota-se que.

$$\theta_{\text{COTOVELO}} = \theta_{\text{OMBRO}} - \phi \quad (2.41)$$

$$\theta_{\text{OMBRO}} = \alpha + \beta \quad (2.42)$$

onde

$$\beta = \arctg \left( \frac{Z_P}{Y_P} \right)$$

Este método de determinar a cinemática inversa resolve todas as situações possíveis para o presente robô porquê o braço articulado sempre apresenta  $\theta_{\text{OMBRO}} > \theta_{\text{COTOVELO}}$ , a ponta do braço não alcança o terceiro quadrante e o ângulo  $\phi$  da figura [2.13] não atinge 180 graus. Entretanto, este método não é

aplicável para braços que podem alcançar o ângulo  $\phi$  igual ou maior que  $180$  graus como no caso do robô PUMA.

A transformação cinemática inversa incluindo o ângulo de mergulho do punho e rotação da garra é feita em duas etapas.

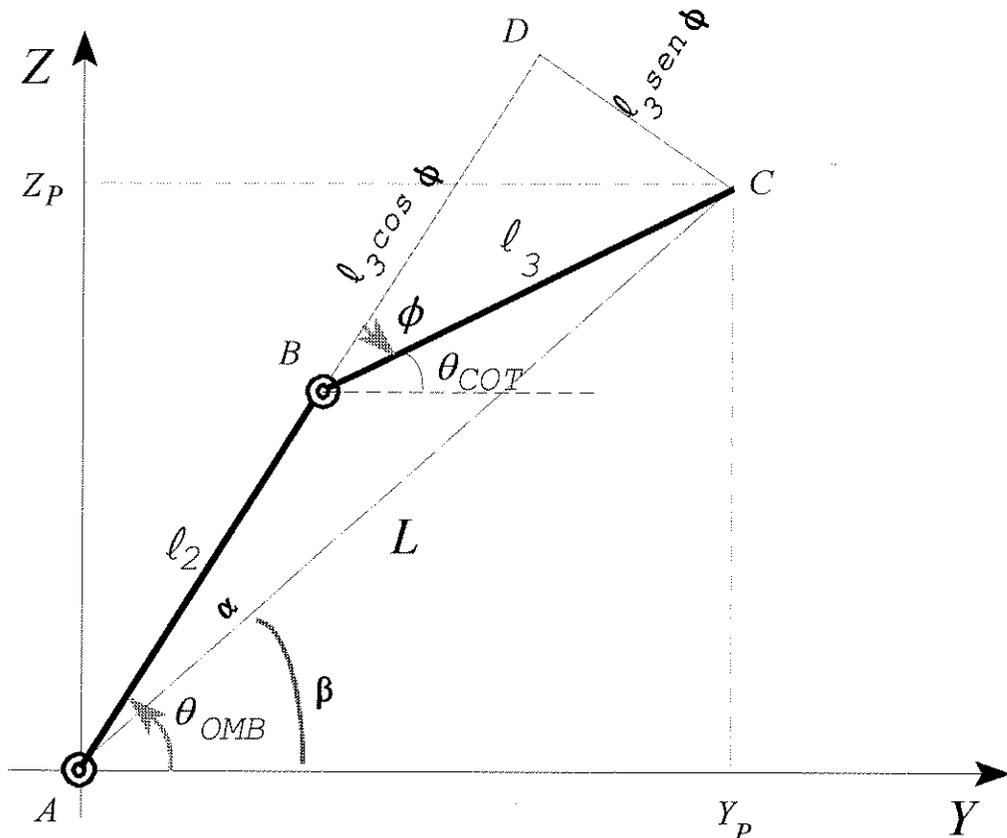


FIG.2.13 BRAÇO ARTICULADO - COTOVELO ACIMA

O ângulo de mergulho do punho e de rotação da garra são definidas pela tarefa, ou pela necessidade de agarrar um objeto com uma determinada orientação ou pela orientação da ferramenta tal como a pistola injetora de tinta, ferramenta de solda a arco, etc. Em outras palavras, os ângulos de mergulho do punho e de rotação da garra são definidos pela própria tarefa.

Assim, inicialmente são calculadas as coordenadas da junta do braço a partir das informações das coordenadas do

centro da garra e do ângulo de mergulho do punho e da rotação da garra e posteriormente calcula-se os demais ângulos pelo método descrito anteriormente.

Como as coordenadas do centro da garra são independentes do ângulo de rotação da garra, a determinação das coordenadas da ponta do braço ilustrado na figura [2.14] a partir das coordenadas do centro da garra e do ângulo de mergulho do punho é feita pela equação (2.43)

$$Y_P = Y_G - l_G \cos \theta_{\text{MERGULHO}} \quad (2.43)$$

$$Z_P = Z_G - l_G \text{sen} \theta_{\text{MERGULHO}}$$

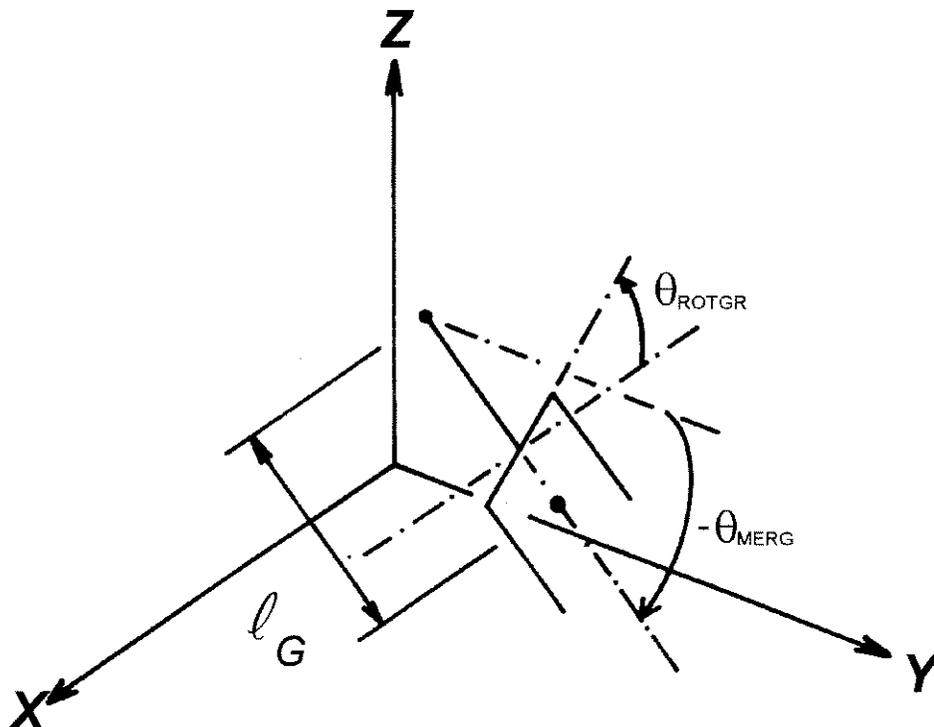


FIG. 2.14 ÂNGULOS DE ORIENTAÇÃO DA GARRA

## 2.6 SELEÇÃO DOS MOTORES

A seleção dos motores para acionamento das juntas de um robô industrial é uma etapa muito importante no projeto do robô pois a seleção errada destes motores, provoca um desempenho insatisfatório no comportamento do robô ou acréscimo desnecessário no custo de fabricação, por estar utilizando motores acima da necessidade.

No presente trabalho, foram feitos estudos para seleção dos motores das juntas da base deslizante, junta do ombro e do cotovelo, baseado no modelo dinâmico dessas juntas.

### 2.6.1 Modelo Dinâmico

O modelo dinâmico do robô, aqui utilizado, foi desenvolvido por Vieira [43] baseado no método apresentado por Spong [28] e está separado em modelo da base deslizante e modelo da junta do ombro conjuntamente com o da junta do cotovelo pois o movimento da base não interfere no movimento das juntas do ombro e do cotovelo e vice-versa, por se tratar de movimento translacional em direção paralela aos vetores de velocidades angulares de ombro e cotovelo.

No desenvolvimento deste modelo, não foram considerados os atritos dos rolamentos, dos fusos de esfera e do guia linear, a fim de se obter um critério de seleção factível.

#### a) Base Deslizante

Para efeito de determinação do modelo dinâmico da base deslizante, foi considerado como a massa móvel, a somatória da massa da base, da estrutura do braço articulado, do punho e da garra.

Assim, o modelo dinâmico da base é dado pela equação de equilíbrio dinâmico de forças de D'Alembert, descrito em (2.44).

$$\begin{aligned} F - ma &= 0 \\ F - (m_1 + m_b)a &= 0 \end{aligned} \quad (2.44)$$

onde:

$F$ : Força de acionamento;

$F_r$ :  $F_r = m_1 a$  - Força de reação imposta pelo restante do mecanismo;

$m_1$ : massa do mecanismo;

$m_b$ : massa da base;

$m$ :  $m = m_1 + m_b$ ;

$a$ : Aceleração da base deslizante

#### b) Junta do Ombro e do Cotovelo

O mecanismo das juntas do ombro e do cotovelo é composto de mecanismo de paralelogramo, conforme apresentado anteriormente.

A figura [2.15] ilustra este mecanismo das juntas para efeito de determinação do modelo dinâmico.

Os valores das distâncias e das massas são:

$$l_1 = 400 \text{ mm}$$

$$l_2 = 100 \text{ mm}$$

$$l_3 = 400 \text{ mm}$$

$$l_{c1} = 241 \text{ mm}$$

$$l_{c2} = 50 \text{ mm}$$

$$l_{c3} = 200 \text{ mm}$$

$$l_{c4} = 471 \text{ mm}$$

$$m_1 = 1617 \text{ g}$$

$$m_2 = 383 \text{ g}$$

$$m_3 = 345 \text{ g}$$

$$m_4 = 5264 \text{ g}$$

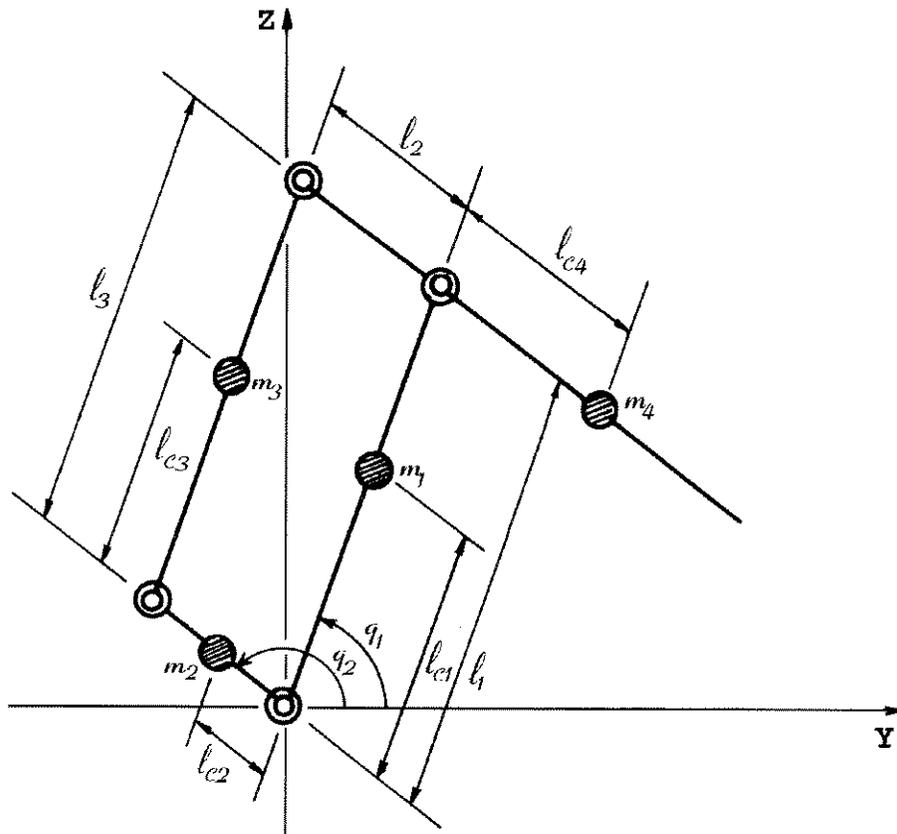


FIG. 2.15 MECANISMO DO BRAÇO

O modelo dinâmico deste mecanismo é obtido através da equação Euler-Lagrange dada por

$$D(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau \quad (2.45)$$

Esta expressão é reescrita no caso de duas juntas como:

$$D_{11} \ddot{q}_1 + D_{12} \ddot{q}_2 + d_{122} \dot{q}_2^2 + G_1 = \tau_1 \quad (2.46)$$

$$D_{21} \ddot{q}_1 + D_{22} \ddot{q}_2 + d_{211} \dot{q}_1^2 + G_2 = \tau_2$$

onde:

$$D_{11} = m_1 \ell_{C1}^2 + m_3 \ell_{C3}^2 + m_4 \ell_1^2 + I_1 + I_3$$

$$D_{12} = d_{21} = (m_3 \ell_2 \ell_{C3} - m_4 \ell_1 \ell_{C4}) \cos(q_2 - q_1)$$

$$D_{22} = m_2 L_{C2}^2 + m_3 L_2^2 + m_4 L_{C4}^2 + I_2 + I_4$$

$$d_{122} = (m_3 \ell_2 \ell_{C3} - m_4 \ell_1 \ell_{C4}) \sin(q_2 - q_1)$$

(2.47)

$$d_{211} = - (m_3 \ell_2 \ell_{C3} - m_4 \ell_1 \ell_{C4}) \sin(q_2 - q_1)$$

$$G_1 = \cos q_1 (m_1 L_{C1} + m_3 L_{C3} + m_4 L_1) g$$

$$G_2 = \cos q_2 (m_2 L_{C2} + m_3 L_2 - m_4 L_{C4}) g$$

$$I_1 = 3,8458 \times 10^{-3} \text{ Kg.m}^2$$

$$I_2 = 3,19166665 \times 10^{-4} \text{ Kg.m}^2$$

$$I_3 = 4,6 \times 10^{-3} \text{ Kg.m}^2$$

$$I_4 = 0,610136 \text{ Kg.m}^2$$

Note que a expressão  $G_1$  só depende da variável  $q_1$  e que  $G_2$  só depende da variável  $q_2$ , pois outros termos que aparecem nestas equações são constantes. Note também que, a expressão  $d_{12}(q)$  que é igual a  $d_{21}(q)$  pode ser nula se  $(m_3 \ell_2 \ell_{C3} - m_4 \ell_1 \ell_{C4})$ , acontecendo o mesmo para  $d_{122}(q)$  e  $d_{211}(q)$ , resultando no modelo

dinâmico desacoplado, ou seja, o acionamento da junta  $q_1$  e  $q_2$  são totalmente independentes entre si, o que facilita o controle dessas juntas.

Convém esclarecer que este fato de possibilitar o desacoplamento através do projeto mecânico é a vantagem do mecanismo de paralelogramo, além da vantagem de poder colocar os motores na parte fixa da base, ou seja, o braço não precisa carregar o peso dos motores que o acionam.

Por estas razões, o mecanismo de paralelogramo vem sendo freqüentemente utilizado na construção dos braços articulados, por permitir controle com algoritmo de compensação dinâmica por técnica de juntas independentes, ou desacopladas dinamicamente.

No presente projeto, outro fato decisivo na escolha deste mecanismo foi a possibilidade de utilizar o fuso de esfera como o mecanismo de redução sem folga da rotação do eixo do motor, que torna o modelo mais linear, com os componentes facilmente adquirível no Brasil e com a construção mecânica simples. Entretanto, não foi possível satisfazer a condição de desacoplamento dinâmico devido principalmente ao peso do mecanismo e dos motores do punho, elementos que totalizam uma massa de 3200 gramas, estarem concentrados na ponta do braço. A solução para satisfazer a condição de desacoplamento seria colocar os motores e os redutores na outra extremidade do braço, mas isso requer praticamente a reconstrução total do braço articulado, que foi impossibilitado pelas razões financeiras.

### 2.6.2 Cálculo dos Torques Necessários para Acionamento das Juntas.

#### a) Base Deslizante

A condição de operação da base deslizante é alcançar a velocidade de 0,635 m/seg em 0,5 segundo, a partir do repouso, numa carga de massa 30,5 Kg. O passo do fuso é de 12,7 mm, com o diâmetro de 19,2 mm e o comprimento de 1200 mm. A massa específica do fuso é de 7913,88 Kg/m<sup>3</sup>.

i) Cálculo do momento de inércia da carga refletido no eixo do motor.

$$J_L = \text{massa} \cdot \left( \frac{\text{passo}}{2 \cdot \pi} \right)^2 \quad (2.48)$$

$$J_L = 0,000124608 \quad [\text{Kg} \cdot \text{m}^2]$$

ii) Cálculo do momento de inércia do fuso.

Como o fuso tem um corpo cilíndrico, o momento de inércia é calculado pela equação (2.49).

$$J_f = 0,5MR^2 \quad (2.49)$$

$$J_f = 0,5\rho\pi LR^4$$

onde:

$\rho$ : massa específica

R: raio da secção

L: comprimento do fuso

$$J_f = 0,000126699 \quad [\text{Kg} \cdot \text{m}^2]$$

A partir dos resultados dos itens (i) e (ii), calcula-se o torque necessário para movimentar a carga com a aceleração desejada. A aceleração linear desejada da carga é de 1,27 m/s<sup>2</sup> que corresponde a aceleração angular de 628,318 rad/s<sup>2</sup> no fuso ou no eixo do motor. Portanto o torque necessário é de.

$$\tau = J_{total} \cdot \alpha \quad (2.50)$$

onde

$$J_{total} = J_f + J_L \quad (2.51)$$

$\alpha$  = aceleração angular  
portanto

$$\tau = 0,000251307 \cdot 628,318$$

$$\tau = 0,157900712 \quad [N.m]$$

Consultando o catálogo dos motores de *SANYO DENKI*, encontra-se 2 motores com os torques imediatamente superiores:

R506 - torque 0,1862 [N.m]

R511 - torque 0,343 [N.m]

Então escolhe-se preliminarmente o motor R506 e calcula-se o torque consumido pelo próprio motor, sabendo-se pelo catálogo que o momento de inércia do rotor é de 0,00008232 Kg.m<sup>2</sup>.

$$\tau_{consumido} = J_M \cdot \alpha \quad (2.52)$$

$$\tau_{consumido} = 0,051723 \quad [N.m]$$

Portanto o torque útil produzido pelo motor R506 é de:

$$\tau_{util} = \tau_{motor} - \tau_{consumido} \quad (2.53)$$

$$\tau_{util} = 0,134477 \quad [N.m]$$

o que é insuficiente para movimentar a massa sob a condição desejada.

Refazendo o mesmo cálculo para o motor R511, cujo momento de inércia é de  $0,00014896 \text{ Kg.m}^2$ , tem-se

$$\tau_{util} = 0,24940575 \quad [\text{N.m}]$$

que satisfaz o torque necessário para condição desejada. Este foi, portanto, o motor selecionado.

b) Junta do Ombro e do Cotovelo.

Devido a estrutura de paralelogramo utilizada nas juntas do ombro e do cotovelo, a taxa de redução entre os eixos dos motores e as juntas não são constantes e são dependentes do ângulo da junta. A equação que calcula a taxa de redução é baseada na equação (2.54).

$$L^2 = L_1^2 + H^2 - 2HL_1\cos\alpha \quad (2.54)$$

onde  $L$ ,  $L_1$ ,  $H$  e  $\alpha$  são indicados na figura [2.5]. Derivando  $L$  em relação a  $\alpha$ , tem-se:

$$\frac{dL}{d\alpha} = \frac{HL_1\sin\alpha}{\sqrt{L_1^2 + H^2 - 2HL_1\cos\alpha}}$$

A variação de  $L$  é uma função do deslocamento angular  $\theta_M$  do eixo do motor, pela equação (2.55)

$$\frac{dL}{d\theta_M} = - \frac{P_F}{2\pi} \quad (2.55)$$

onde  $P_F$  é o passo do fuso e é igual a 5 mm.

Assim, a variação do  $\alpha$  em função do deslocamento angular do eixo do motor  $\theta_M$  é dada pela equação (2.56).

$$\frac{d\alpha}{d\theta_M} = \frac{-P_F \sqrt{L_1^2 + H^2 - 2HL_1 \cos\alpha}}{2\pi HL_1 \sin\alpha} \quad (2.56)$$

Como,

$$\frac{d\alpha}{d\theta} = 1 \quad (2.57)$$

onde  $\theta$  é o ângulo da junta, o cálculo da taxa de redução entre o eixo do motor e a junta é feito pela equação (2.58).

$$\frac{d\theta}{d\theta_M} = \frac{-P_F \sqrt{L_1^2 + H^2 - 2HL_1 \cos\alpha}}{2\pi HL_1 \sin\alpha} \quad (2.58)$$

O torque do motor necessário para acionamento de cada uma das juntas é calculado inicialmente baseado no modelo dinâmico do mecanismo de paralelogramo e posteriormente é dividido pela taxa de redução entre o eixo do motor e a junta para obter o valor do torque necessário do motor.

Os torques necessários para acionar as juntas do ombro e do cotovelo são dependentes dos ângulos das juntas. Assim, para calcular os torques necessários foram escolhidos os ângulos das juntas, do ombro e do cotovelo, numa situação onde são demandados os torques máximos, onde  $q_1$  e  $q_2$  da figura [2.15] são 25 e 180 graus respectivamente, com a aceleração de  $100^\circ/\text{s}^2$  ( $1,745329 \text{ rad}/\text{s}^2$ ) e velocidade das juntas são  $100^\circ/\text{s}$ . Nestas condições, os torques necessários para acionamento são:

$$\tau_1 = 24,738317 \quad [\text{N.m}]$$

$$\tau_2 = 29,708754 \quad [\text{N.m}]$$

Nesta situação de ângulos das juntas, as reduções são:

$$RG_1 = 90,668846$$

$$RG_2 = 123,63654$$

Com estes valores das reduções, os torques refletidos nos motores são:

$$T_{M1} = 0,272843 \quad [\text{N.m}]$$

$$T_{M2} = 0,240291 \quad [\text{N.m}]$$

E pela mesma taxa de redução calcula-se a aceleração angular nos eixos dos motores e dos fusos:

$$\alpha_{M1} = 158,246966 \quad [\text{rad/seg}^2]$$

$$\alpha_{M2} = 215,78644 \quad [\text{rad/seg}^2]$$

Por outro lado, deve ser considerado os torques consumidos pelos momentos de inércia dos fusos, que têm diâmetros 16 mm e comprimento 260 mm, o que dá um momento de inércia igual a 0,000013238 Kg.m<sup>2</sup>.

Assim, os torques consumidos pelos fusos são:

$$\tau_{f1} = J_f \cdot \alpha_{M1}$$

$$\tau_{f1} = 0,002094873 \quad [\text{N.m}]$$

$$\tau_{f2} = J_f \cdot \alpha_{M2}$$

$$\tau_{f2} = 0,00285658 \quad [\text{N.m}]$$

Portanto, os torques necessários para acionar as juntas e fusos são:

$$\tau_{JF1} = 0,274938 \quad [\text{N.m}]$$

$$\tau_{JF2} = 0,243148 \quad [\text{N.m}]$$

Após calculados os torques necessários para acionar as juntas e os fusos, foi escolhido o motor R511 da SANYO DENKI que tem o torque de 0,343 N.m. Como este motor tem o rotor com momento de inércia igual a 0,00014896 Kg.m<sup>2</sup>, os torques consumidos são, respectivamente:

$$\tau_{cons1} = J_M \cdot \alpha_{M1} \quad [\text{N.m}] \quad (2.59)$$

$$\tau_{cons1} = 0,023572$$

$$\tau_{cons2} = J_M \cdot \alpha_{M2} \quad [\text{N.m}] \quad (2.60)$$

$$\tau_{cons2} = 0,0321435$$

Portanto os torques úteis produzidos pelos motores são:

$$\tau_{util1} = \tau_M - \tau_{cons1} \quad \text{[N.m]} \quad (2.61)$$

$$\tau_{util1} = 0,319428$$

$$\tau_{util2} = \tau_M - \tau_{cons2} \quad \text{[N.m]} \quad (2.62)$$

$$\tau_{util2} = 0,3108565$$

Uma vez que, tanto na junta do ombro quanto na junta do cotovelo, o torque útil deste motor é maior do que o torque necessário para acionar a junta e o fuso, este motor é adequado para o acionamento das juntas.

Convém esclarecer que os cálculos realizados no presente trabalho são cálculos preliminares e quando o torque útil do motor é próximo ao valor do torque necessário, é necessário realizar outros cálculos tais como a elevação da temperatura do motor considerando o ciclo de trabalho e o coeficiente de dissipação do calor do motor pois a temperatura do motor poderá alcançar a temperatura de desimantação, danificando o motor [6].

## 2.7 VOLUME DE TRABALHO.

O volume de trabalho de um robô é um conjunto de todos os pontos alcançáveis pelo robô e são expressos em forma de desenhos que mostram a vista lateral onde é mostrado projeção horizontal da área alcançável pelo robô.

A figura [2.16] ilustra o volume de trabalho do robô do presente projeto.

Foram, também, representados nesta figura, os ângulos máximo e mínimo da junta do ombro e do cotovelo.

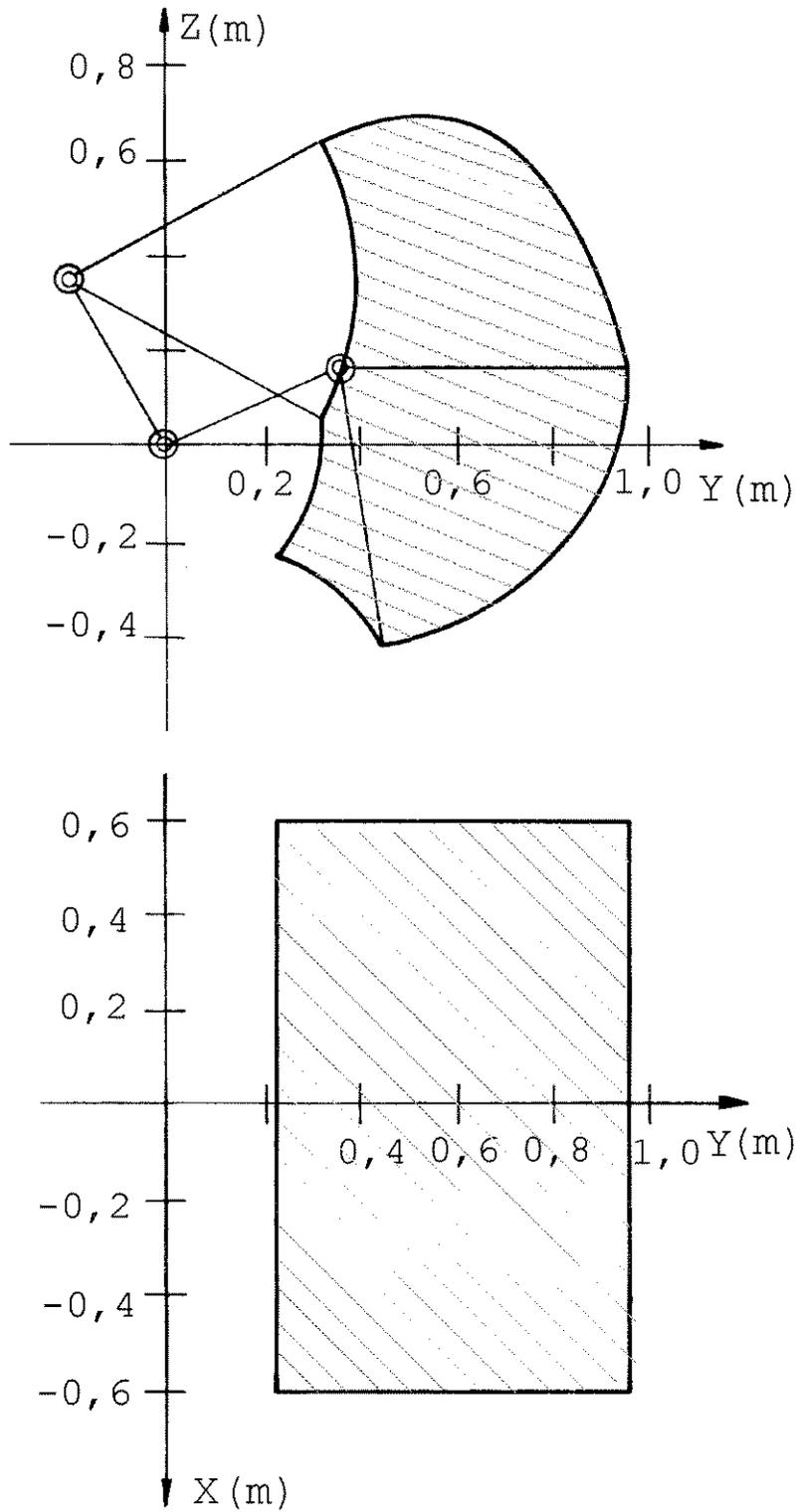


FIG. 2.16 VOLUME DE TRABALHO DO ROBÔ

## CAPÍTULO 3

### ELEMENTO FINAL DE AÇÃO

#### 3.1 INTRODUÇÃO

O elemento final de ação de um robô é um dispositivo instalado na extremidade do braço do robô que é posicionado e orientado pelo mecanismo de braço e punho, sendo o braço responsável pelo posicionamento e o punho pela orientação, e tem a função de executar efetivamente as tarefas específicas.

Muitas vezes, este elemento consiste de ferramentas específicas para determinadas tarefas tais como: pistola injetora de tinta, que tem a função de pulverizar a tinta para a pintura de uma determinada superfície; ferramenta de solda a ponto para soldagens de chapas metálicas finas ou a ferramenta para solda a arco; sistema de sucção a vácuo para segurar e transportar chapas metálicas finas, utilizado para carregar e descarregar chapas metálicas na prensa para conformação ou para segurar, transportar e fixar o pára-brisa na janela frontal de um automóvel nas indústrias automobilísticas; bico injetor de adesivos para aplicação no pára-brisa de automóvel na linha de montagem; sistema de rebolo rotativo de pequena dimensão para retirar as *rebarbas* das peças metálicas usinadas pelas máquinas operatrizes e outras ferramentas projetadas para aplicações específicas.

Por outro lado, existe uma classe de elemento final de ação para agarramento e manipulação de objetos genéricos que normalmente denomina-se de GARRA.

No presente projeto foi desenvolvida uma garra de dois dedos com a integração de diversos tipos de sensores para possibilitar o agarramento de um objeto frágil.

### 3.2 GARRA COM SENSORES.

A garra do robô industrial exerce uma função muito importante na utilização industrial ou não industrial do robô assim como a mão humana na execução de diversas tarefas exercidas pelo homem na indústria. É inegável que a mão do homem é a parte do corpo onde estão concentrados diversos sensores tais como, o sensor de tato, de força e a capacidade de detectar o deslizamento entre a mão e algum objeto.

A garra do robô industrial é tradicionalmente projetada para agarrar objetos metálicos de alta resistência mecânica e devido ao seu tamanho reduzido, o mecanismo costuma ser bem simples de dois ou três dedos com poucas articulações. Com isso, ao agarrar um objeto, a garra aplica uma força máxima para assegurar um agarramento firme.

Com a ampliação do campo de aplicações dos robôs industriais, hoje, os objetos a serem agarrados podem não ter alta resistência mecânica. Em outras palavras, existe uma necessidade de se ter uma garra que realize o agarramento seguro de objetos frágeis, com controle de força que permita que o objeto não seja danificado.

Para isso existem várias linhas de estudo e algumas delas têm optado pela garra semelhante à mão humana em mecanismo, ou seja com três ou quatro dedos e tendo várias articulações em cada um dos dedos, quando se necessita a manipulação do objeto além de seu agarramento. Outros optaram pelo mecanismo simples de dois dedos rígidos e paralelos mas instalando diversos tipos de sensores, para os casos onde se requer somente agarramento.

O presente trabalho apresenta o desenvolvimento de uma garra baseada na estratégia de mecanismo simples com sensores.

A garra desenvolvida possui dois dedos rígidos de movimentos paralelos acionados por um servomotor de corrente contínua, com sensor de abertura dos dedos, sensor de força de agarramento e sensor de deslizamento entre os dedos e o objeto.

O sistema de controle é baseado em microcomputador

compatível ao IBM-PC-AT/286 com circuito de interfaceamento composto de conversores A/D e D/A e contadores *up/down*.

### 3.3 PARTE MECÂNICA DA GARRA.

A parte mecânica da garra é composta de dois dedos e um servomotor de corrente contínua de 12 Volts/2 Watts e torque de pico 0,00245 N.m com caixa de redução de 1:100 baseada em trem de engrenagens planas, um par de engrenagens que provoca os movimentos dos dedos nos sentidos opostos entre si e quatro barras rígidas que mantém o paralelismo dos dedos.

A figura [3.1] ilustra o mecanismo da garra e a posição dos sensores instalados.

Ao acionar o motor-reductor  $M_T$ , a engrenagem  $E_M$  transmite a rotação para a engrenagem  $E_1$  que por sua vez aciona o link  $L_1$  e a engrenagem  $E_2$ . A engrenagem  $E_2$  aciona o link  $L_2$  em sentido oposto a  $L_1$ . Outros dois *links*  $L_3$  e  $L_4$  têm a função de manter os dedos da garra em movimentos paralelos, realizando o movimento de abre-fecha dos dedos.

O sensor de abertura  $S_A$  é um potenciômetro acionado por  $E_2$  através da engrenagem  $E_P$  conectada no seu eixo. Os sensores de deslizamento e de força,  $S_D$  e  $S_F$  respectivamente, estão instalados na parte interna dos dedos, pois aqui a finalidade dos sensores é detectar o grau de interação entre os dedos e o objeto que está sendo agarrado.

### 3.4 SENSOR DE FORÇA.

Foram desenvolvidos dois tipos de sensor de força:

#### 3.4.1 Sensor Analógico de Força.

É constituído de um par de resistores e um par de *strain gages* compondo uma ponte de *Wheatstone* com a tensão de alimentação de 12 Volts e um amplificador diferencial de ganho 100, figura [3.2].

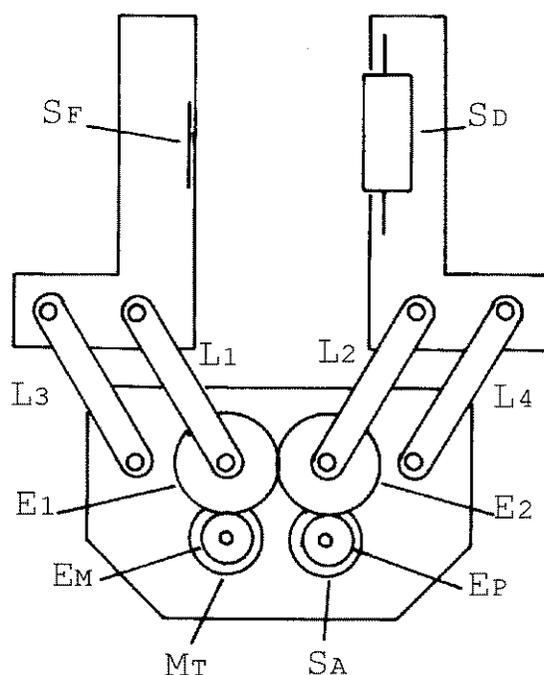


FIG. 3.1 GARRA COM SENSORES

A utilização da ponte de *Wheatstone* tem duas finalidades:

1. Para aumentar a sensibilidade dos *strain gages* que estão colados nas duas faces opostas de uma lâmina flexível que, ao sofrer deflexão pela aplicação da força, faz com que um dos *strain gages* seja comprimido, diminuindo a sua resistência elétrica, enquanto o outro é tensionado, aumentando a sua resistência elétrica, desbalanceando assim a ponte com a sensibilidade dupla.
2. Para cancelar o efeito de variação da resistência elétrica dos *strain gages* com a variação da temperatura.

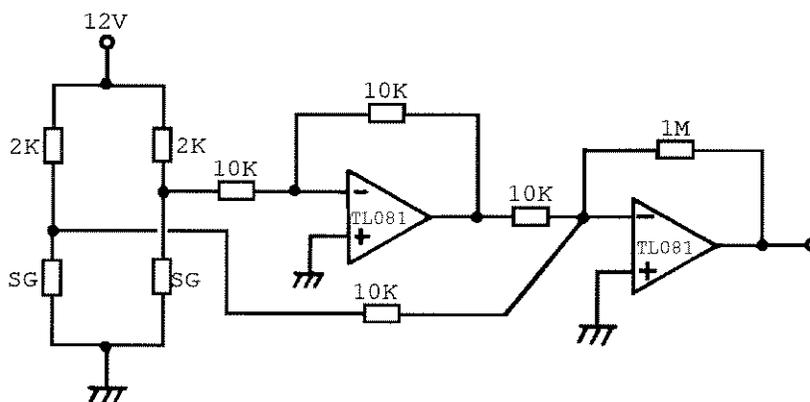


FIG. 3.2 CIRCUITO DO SENSOR ANALÓGICO DE FORÇA

A instalação desta lâmina com dois *strain gages* na parte interna de um dos dedos é ilustrada na figura [3.3].

Nesta figura, a chapa flexível está em contato direto com objeto e é flexionada com a força dos dedos, transmitindo a deflexão para lâmina com *strain gages*. O conjunto parafuso e mola serve para calibrar a sensibilidade do sensor de força.

### 3.4.2 Sensor Digital de Força.

O sensor de força baseado nos *strain gages* tem a vantagem de ter um aspecto construtivo simples mas como os sinais elétricos que ele emite são muito pequenos, a contaminação destes sinais em ambiente industrial pode causar problemas.

Baseado nesse ponto de vista, os autores deste trabalho desenvolveram um sensor de força, inédito no mundo, que emite sinais digitais ao invés do sinal analógico.

O princípio básico de detecção da força neste sensor, baseado na roda de atrito, é ilustrado na figura [3.4].

Neste mecanismo, a haste móvel está em contato com o

eixo do codificador angular e quando a haste é movimentada em  $\Delta x$  na direção indicada na figura, o eixo do codificador é rotacionado em  $\Delta\theta$ .

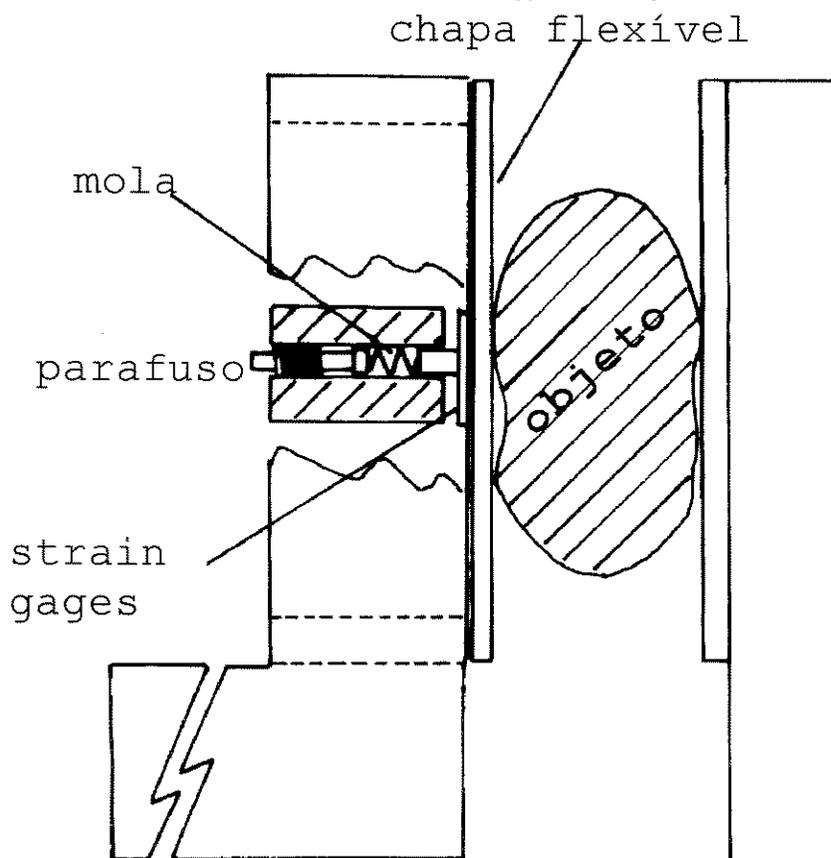


FIG. 3.3 SENSOR ANALÓGICO DE FORÇA

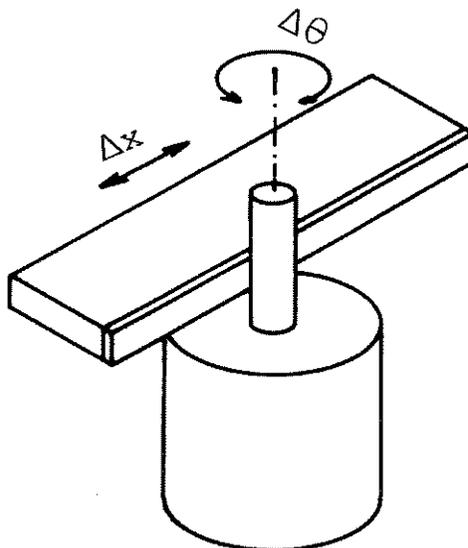


FIG. 3.4 PRINCÍPIO BÁSICO DO SENSOR DIGITAL DE FORÇA

Este codificador angular tem o eixo de diâmetro 2,5 mm e emite 200 pulsos/volta que pode ser multiplicado eletronicamente por 4, resultando em 800 pulsos/volta.

A relação quantidade de pulsos X deslocamento linear da haste é calculada da seguinte maneira:

Sabendo-se que uma volta no eixo do codificador angular corresponde a 800 pulsos e que a mesma corresponde ao deslocamento linear da haste equivalente ao perímetro do eixo, uma vez que a haste está em contato com a superfície do eixo do codificador angular, o deslocamento linear da haste  $\Delta x$  correspondente a uma volta do eixo do codificador angular é:

$$\Delta x = \pi \cdot D \quad (3.1)$$

Como  $\Delta x = 7,854 \text{ mm}$  corresponde a 800 pulsos, cada pulso do codificador corresponde a 0,0098 mm. Esta resolução é suficiente para detectar a força aplicada pelos dedos no objeto no presente caso, pois a força máxima dos dedos provoca um recuo de 1mm na chapa flexível instalada no dedo que corresponde a 102 pulsos.

Caso seja necessária uma resolução maior no sensor de força, atualmente já se encontra comercialmente disponível um codificador angular com a mesma dimensão, mas com a resolução de 500 pulsos/volta e com a utilização deste pode-se obter uma resolução de 0,00392 mm/pulso.

A contagem dos pulsos é realizada por circuito similar ao utilizado pelo sensor de posição das juntas do braço mecânico.

### 3.4.3 Calibração de Sensor de Força.

A calibração do sensor de força tem como objetivo determinar a relação entre o valor digital fornecido pelo sensor e o valor real da força nele aplicada.

Esta determinação foi feita através de ensaios, da seguinte maneira: Inicialmente o dedo onde está instalado o sensor é colocado em posição horizontal e com a face do sensor virada para cima. Posteriormente são colocados sobre o dedo objetos de massas conhecidas, um de cada vez, e é realizada a leitura do valor digital indicado pelo sensor. Os valores das massas colocadas foram 200g, 300g, 400g, 500g, 600g, 700g, 800g, 900g e 1000g e foi detectado que este sensor é insensível para massa abaixo de 180 gramas. A tabela [3.1] indica os valores das massas e os respectivos valores digitais e a figura [3.5] mostra o gráfico de massa em grama versus o valor digital indicado pelo sensor digital de força (curva de calibração do sensor).

massa do objeto em grama	saída do contador
200	4
300	9
400	14
500	19
600	24
700	29
800	34
900	39
1000	44

TABELA 3.1 RELAÇÃO ENTRE A MASSA E A SAÍDA DO CONTADOR

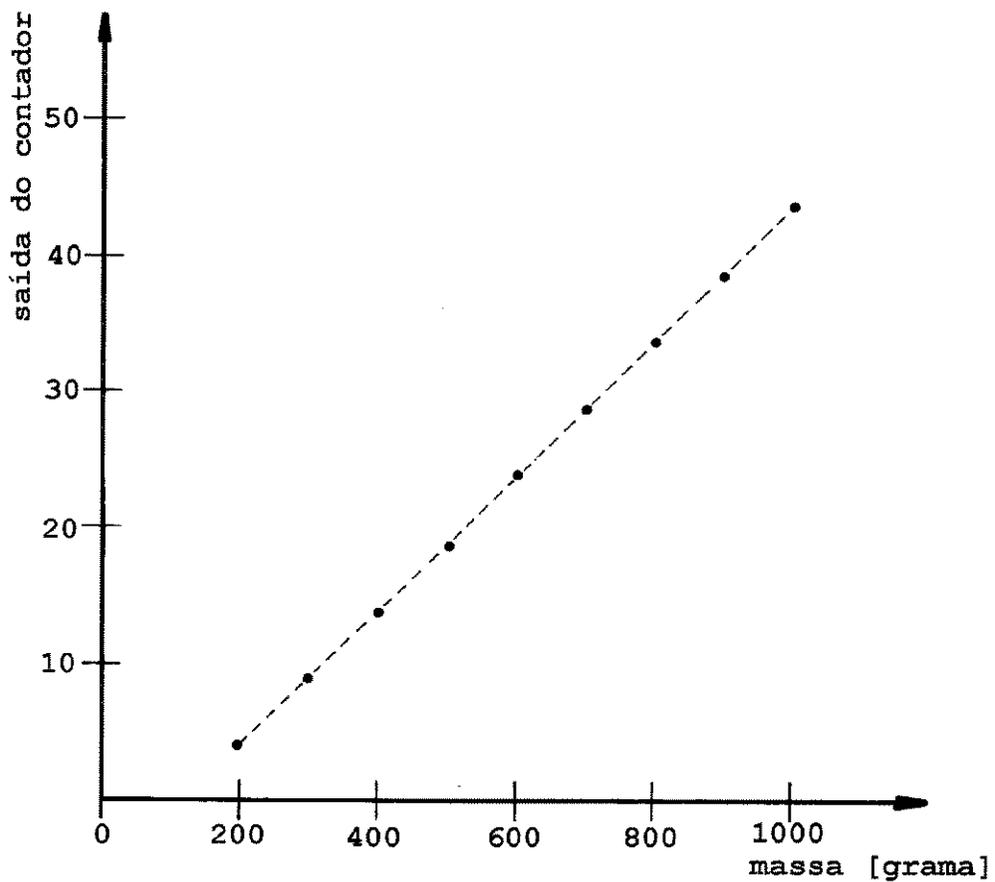


FIG. 3.5 MASSA VERSUS SAÍDA DO CONTADOR

Considerando a aceleração da gravidade de  $9,8 \text{ m/s}^2$  o valor real da força, em Newton, aplicada ao sensor digital, é calculada pela equação [3.2].

$$\text{FORÇA} = \frac{\text{SAÍDA DO CONTADOR} + 5,99}{5,1} \quad (3.2)$$

Como o sensor baseado em *strain gages*, tem o seu valor analógico convertido em digital pelo conversor A/D, o valor digital indicado pelo sensor é numericamente 4 vezes maior do que o resultado indicado pelo sensor digital. Portanto, para converter o resultado do sensor analógico, é necessário que se divida inicialmente por 4 antes de aplicar a equação (3.2).

### 3.5 SENSOR DE DESLIZAMENTO.

A utilização do sensor de deslizamento na garra do robô é uma técnica ainda não muito divulgada mas é a forma mais adequada para aplicar a força mínima para agarrar e transportar um objeto. Este sensor foi inspirado na sensibilidade da mão humana que ao agarrar e levantar um objeto utiliza a sensibilidade em detectar deslizamento.

Diversos são os mecanismos de detecção de deslizamento e aqui foi adotado o mecanismo ilustrado na figura [3.6].

Neste mecanismo, o sensor é constituído de um cilindro e codificador angular conectados através de correia. Na ausência do objeto, este cilindro está saliente na face interna do dedo, pressionado por molas.

Quando os dedos agarram um objeto, este conjunto recua e o cilindro permanece pressionando o objeto com a força das molas.

Assim, quando ocorre deslizamento do objeto na direção normal ao plano formado por dois dedos, o objeto em deslizamento provoca movimento rotativo no cilindro que por sua

vez gira o eixo do codificador que emite uma quantidade de pulsos proporcional à magnitude do deslizamento.

O diâmetro do cilindro utilizado neste trabalho é de 18 mm, a resolução do codificador é de 100 pulsos/volta e a relação entre os diâmetros das polias é de 1:8. Assim, o deslizamento mínimo que pode ser detectado é de aproximadamente 0,016 milímetros.

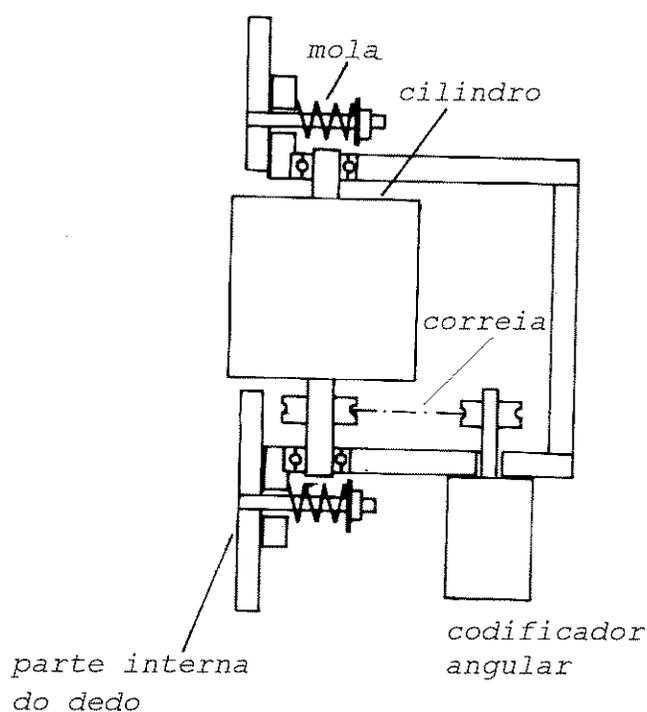


FIG. 3.6 SENSOR DE DESLIZAMENTO

Foi colocado um contador digital na saída do codificador para que a sensibilidade do sensor de deslizamento possa ser controlada pelo *software*.

### 3.6 SENSOR DE ABERTURA DOS DEDOS.

O sensor de abertura dos dedos é um potenciômetro cujo eixo é conectado ao eixo de acionamento através de engrenagens.

Devido ao mecanismo de *link*, a abertura dos dedos é calculada por *software* a partir da informação obtida por este sensor. A equação (3.1) calcula a abertura dos dedos onde  $B$ ,  $D$  e  $L$  estão indicados na figura [3.7] e  $\theta$  é o valor obtido por sensor de abertura.

$$a = D + 2(L \cos \theta - B) \quad (3.3)$$

onde:  $D = 30 \text{ mm}$

$L = 42 \text{ mm}$

$B = 14 \text{ mm}$

A conversão do valor digital indicado pelo conversor A/D em valor do ângulo  $\theta$  é calculado, considerando a seguinte situação: Quando os dedos estão fechados o ângulo  $\theta$  assume o valor 91,36 graus e quando os dedos estão totalmente abertos, o ângulo  $\theta$  é igual a 36,53 graus e foram ajustados o ganho do conversor A/D e a posição do sensor para que o valor 91,36

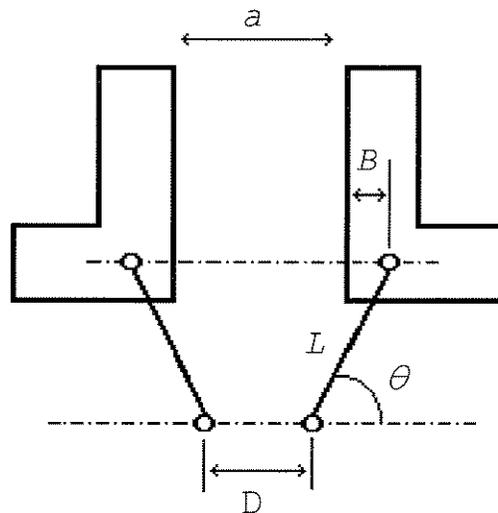


FIG. 3.7  $B$ ,  $D$ ,  $L$  E  $\theta$  DA EQUAÇÃO (3.3)

graus no ângulo  $\theta$  corresponda ao valor digital zero na saída do A/D e o valor 36,53 graus no ângulo  $\theta$  corresponda ao valor digital 255 na saída do A/D. Assim, para calcular o valor do ângulo  $\theta$  a partir da leitura do A/D utiliza-se a equação (3.4).

$$\theta = 91,36 - 0,215 \cdot AD \quad [\text{graus}] \quad (3.4)$$

onde:  $AD$  é o valor indicado pelo conversor A/D.

### 3.7 CONTROLADOR DA GARRA.

O controlador da garra é baseado em microcomputador compatível a PC-AT/286 adicionando neste um circuito de interfaceamento e *software* de controle.

#### 3.7.1 Hardware

O *hardware* do sistema é composto de:

1. Microcomputador PC-AT com a CPU 80286, *clock* de 10 Mhz, 1 Mbyte de RAM, disco rígido de 42 Mbytes.
2. Circuito de acionamento do motor tipo amplificador DC;
3. Circuito de tratamento de sinal do codificador angular para sensor de força e para sensor de deslizamento;
4. Conversor digital/analógico para circuito de acionamento do motor;
5. Conversor analógico/digital para sensor de abertura;
6. Contadores digital para sensor de força e para sensor de deslizamento;
7. Canal de comunicação paralela com controlador do braço.

Estes circuitos estão integrados no sistema conforme a ilustração da figura [3.8].

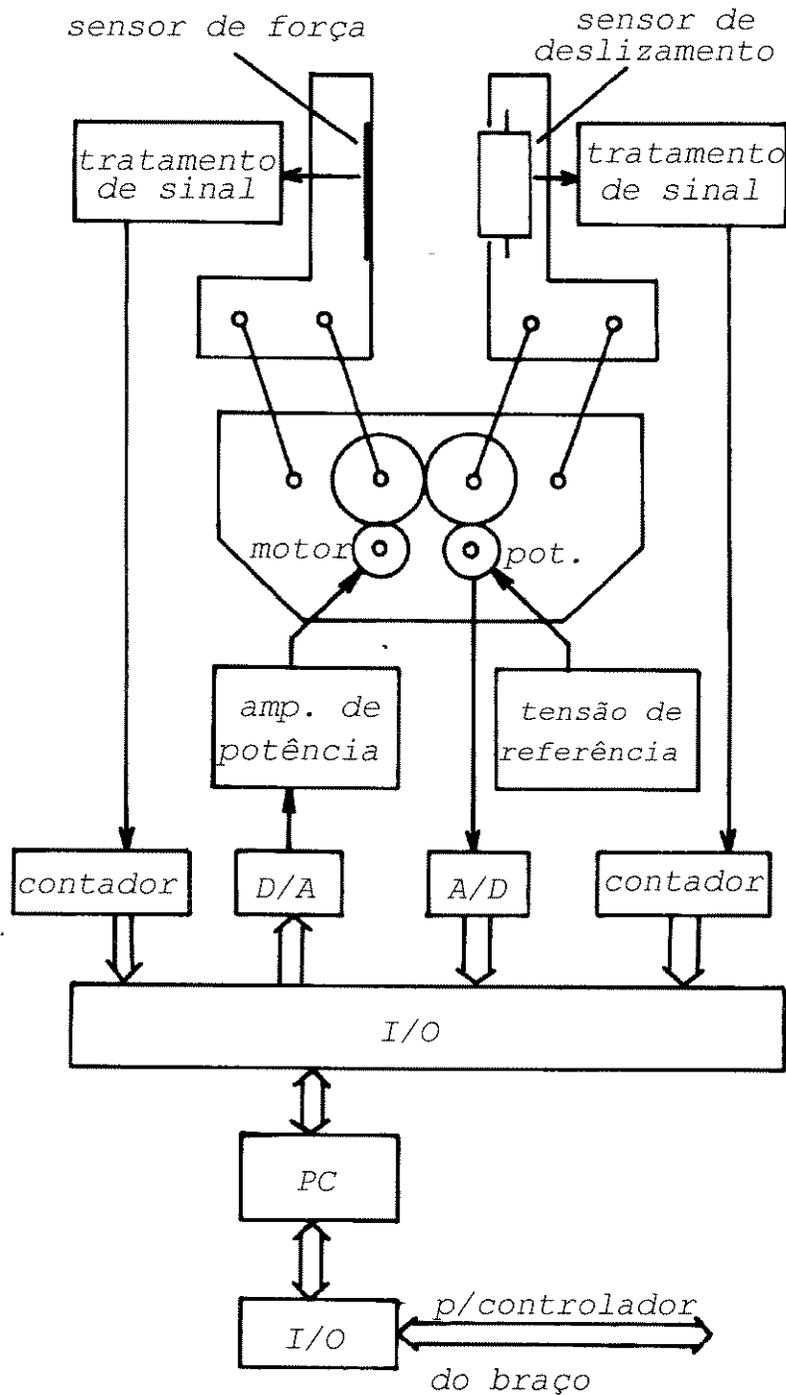


FIG. 3.8 HARDWARE DA GARRA

### 3.7.2 Software

O software do sistema de controle da garra é dividido em dois níveis hierárquicos sendo:

1. Controle Supervisor;
2. Controle do servomecanismo.

O controle supervisor é responsável pela recepção e interpretação dos comandos proveniente do controlador do braço que define quando e de que modo a garra deve entrar em ação. Outra função do supervisor é enviar informação ao controlador do braço, quanto ao estado atual da garra, para que possa decidir quanto ao movimento correspondente do braço.

O controlador do servomecanismo é responsável pela operação da garra em quatro modos de operação:

1. Fechar ou abrir a garra conforme a abertura solicitada pelo supervisor;
2. Fechar a garra até o sensor de força acusar contato do dedo com o objeto e aumentar a força dos dedos sobre o objeto quando detectado o deslizamento do objeto entre os dedos até a parada do deslizamento e/ou até a força atingir o valor limite indicado pelo controlador supervisor;
3. Abrir a garra quando detectado o deslizamento;
4. Diminuir a força da garra até o início de deslizamento do objeto e permanecer nesta força até a parada do deslizamento. Como efeito, o objeto desce suavemente até encostar no patamar por onde será colocado.

Cada um desses modos de operação tem a aplicação típica.

O controle de abertura é necessário quando o espaço onde o objeto se encontra tem obstáculos em volta dele que impeçam a entrada da garra se a mesma estiver totalmente aberta, como é ilustrado na figura [3.9].

Os sensores de força e deslizamento, utilizados em conjunto, possibilitam o agarramento de objetos frágeis pois inicialmente os dedos da garra ficam apenas em contato com o

objeto sem aplicar a força. Ao tentar levantar o objeto ocorrerá o deslizamento e conseqüentemente os dedos aumentarão a força até a parada do deslizamento.

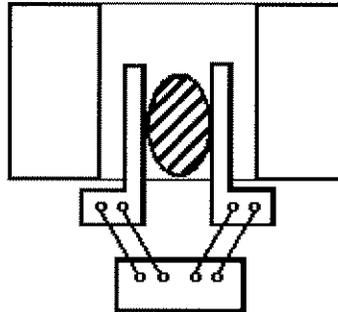


FIG. 3.9 GARRA ENTRANDO NO AMBIENTE  
COM OBSTÁCULOS

Uma outra aplicação do sensor de deslizamento é descrito a seguir: Ao programar o movimento do braço, normalmente é necessário programar a coordenada do ponto de parada para largar o objeto. Entretanto, se a altura do objeto que o braço transporta for diferente daquela programada, poderão ocorrer duas situações de risco:

1. o objeto atinge o patamar e o braço continua forçando para baixo, correndo o risco de danificar o objeto;
2. o objeto ainda se encontra longe do patamar e a garra abre para largar o objeto que cairá em queda livre.

Estas duas situações de risco poderão ser resolvidas utilizando-se a garra com sensor no modo (3) de operação.

### 3.8 EXPERIMENTOS REALIZADOS.

Foram realizados os seguintes experimentos para convalidação do controle da garra com sensores.

#### 3.8.1 Controle de Força.

Foram feitos alguns experimentos de agarramento de objeto com controle de força, utilizando o sensor analógico de força e posteriormente utilizando o sensor digital de força.

A figura [3.10] ilustra o resultado do controle de força utilizando sensor analógico de força onde o eixo vertical indica o valor da força obtida no conversor analógico digital. Pode-se notar nesta figura, a contaminação do sinal de força, resultando numa faixa larga em torno do *set point*.

A figura [3.11] ilustra o resultado de controle de força onde o eixo vertical indica o valor da força obtida pelo contador digital de pulsos emitidos pelo sensor digital de força e pode-se notar que não há variação brusca no valor da força como foi observado no experimento com sensor analógico.

#### 3.8.2 Controle de Abertura dos Dedos.

O algoritmo de controle de abertura dos dedos experimentado é o controle PI para posicionamento e demonstrou um bom desempenho, anulando o erro de regime. A figura [3.12] mostra o comportamento da abertura da garra.

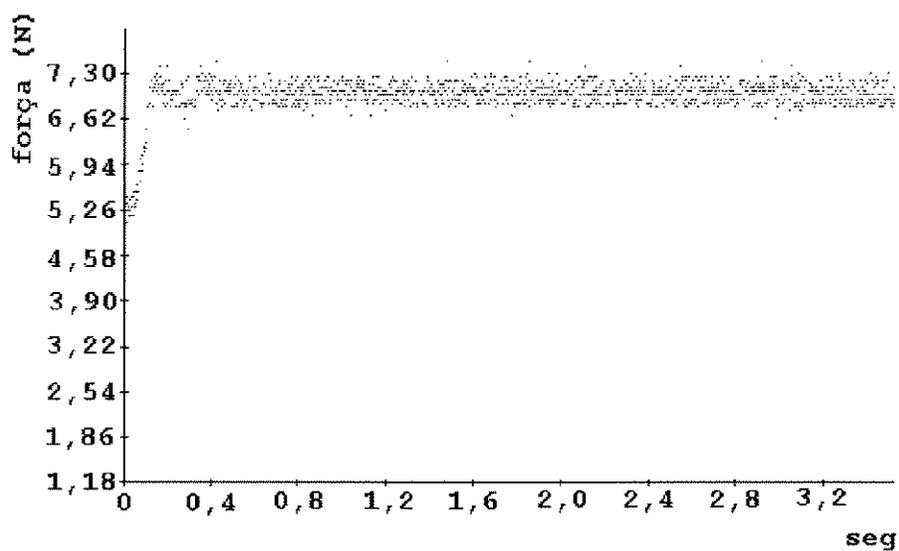


FIG. 3.10 SENSOR ANALÓGICO DE FORÇA

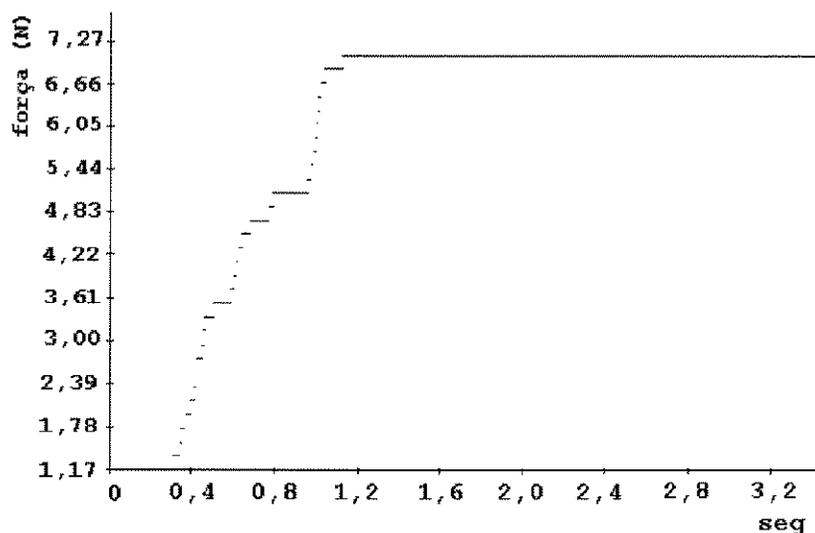


FIG. 3.11 SENSOR DIGITAL DE FORÇA

### 3.8.3 Controle de Força dos Dedos com Sensores de Deslizamento.

Foram realizados três tipos de experimentos para verificar diferentes modos de atuação do sensor de deslizamento.

1. Inicialmente a garra segura um recipiente vazio com força suficiente para segurar somente o recipiente. A variação da força dos dedos aplicada ao recipiente é monitorada através da tensão aplicada ao motor. Em seguida, são colocados neste recipiente pesos adicionais, aumentando-se gradativamente. Foi observado que não houve deslizamento perceptível (o deslizamento foi aproximadamente de 0,5 mm). Entretanto, a tensão aplicada ao motor aumenta à medida que aumenta o peso do recipiente, mostrando assim a atuação do sensor de deslizamento. Este mesmo experimento foi realizado sem o sensor de deslizamento e foi observado um deslizamento de aproximadamente 15 mm do recipiente.
2. A garra segura um objeto com força constante. Quando provoca um pequeno deslizamento no objeto a garra abre imediatamente os dedos, largando o objeto, figura [3.13].
3. Como experimento de agarramento de objetos frágeis, foi carregado um ovo de galinha, sem quebrá-lo.

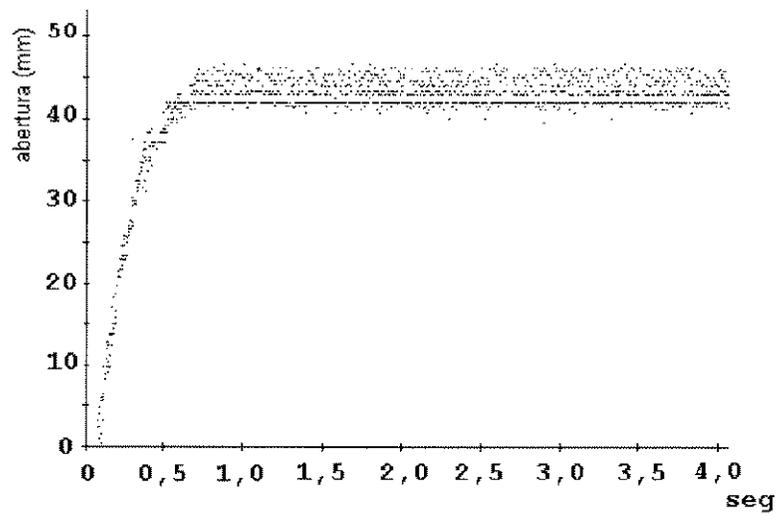


FIG. 3.12 CONTROLE DE ABERTURA DOS DEDOS

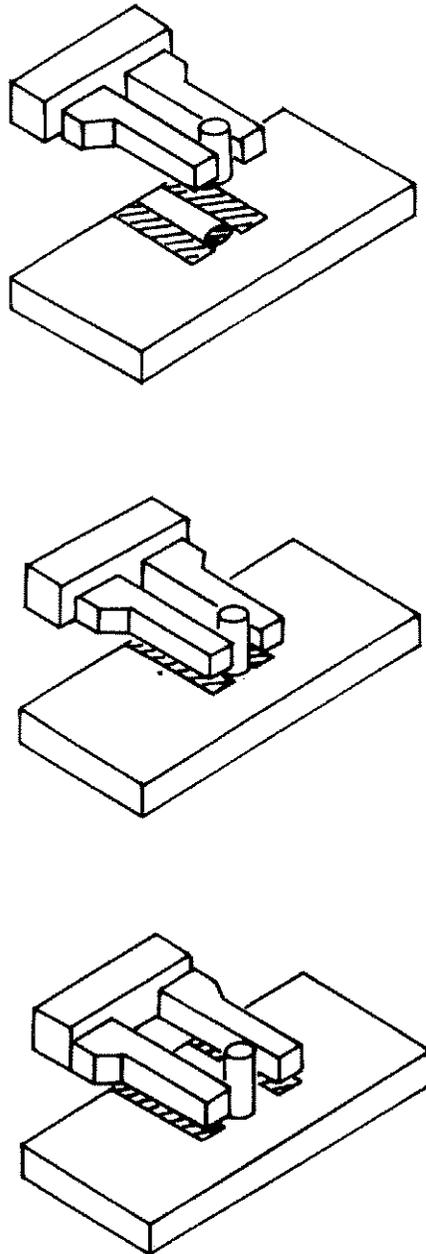


FIG. 3.13 SEQUÊNCIA DE MOVIMENTO DA GARRA NO MODO DESLIZA-ABRE

# CAPÍTULO 4

## SENSOR DE VISÃO

### 4.1 INTRODUÇÃO

O sensor de visão é um dos sensores mais ricos em quantidade de informações e a sua utilização é dependente do grau de sofisticação do *hardware* acoplado à câmera para interpretação dos sinais gerados, tais como: a resolução da imagem adquirida, a capacidade de distinção de cores, a capacidade de distinção de níveis de cinza etc., e da natureza do *software* em processar estas informações necessárias para uma determinada aplicação.

O presente trabalho de desenvolvimento do sensor de visão teve como objetivo dominar a técnica básica de digitalização e aquisição da imagem (captada por uma câmera de TV comercial de padrão *NTSC* com a resolução suficiente para análise e identificação de objetos em tempo real, feita por *software*, num microcomputador comercial tipo PC-IBM com CPU 80386 SX de 20MHz com coprocessador 80387, 4MByte de RAM e Disco Rígido de 40 Mbyte), e a utilização deste sensor no processo de agarramento de objeto por robô, e possibilitar a aplicação de estratégias de controle mais sofisticadas.

### 4.2 DESCRIÇÃO TÉCNICA.

O sensor de visão é composto de diversos dispositivos de processamento dos sinais da imagem:

1. Dispositivo para capturar a imagem;
2. Circuito de digitalização e armazenamento da imagem, normalmente denominado de digitalizador.
3. Sistema de processamento de imagem para extrair as



#### 4.2.2 Circuito Digitalizador e Armazenamento de Imagem.

O circuito digitalizador desenvolvido no presente trabalho utiliza principalmente os dispositivos a TTL e memória e está dedicado ao sinal de vídeo composto padrão NTSC. Esta escolha foi baseada no fato de que o padrão NTSC é o mais difundido no mundo e que normalmente a câmera de TV, que é o dispositivo usual de captura de imagem, emite o sinal de vídeo composto.

A descrição técnica detalhada do sinal de vídeo composto assim como o circuito de digitalização da imagem se encontram no apêndice A.

#### 4.3 SOFTWARE DO SENSOR DE VISÃO.

O software desenvolvido tem como objetivo fazer a transferência da imagem digitalizada para a memória do computador e posteriormente calcular as coordenadas do centro geométrico do objeto, identificar o objeto e detectar a sua orientação.

As coordenadas do centro geométrico do objeto, assim como a sua orientação, são as informações úteis no processo de agarramento do objeto pela garra do robô, pois estas informações dão maior flexibilidade na programação das tarefas, assim como aumenta a confiabilidade do processo de agarramento e transporte do objeto.

Considere-se por exemplo, um objeto retangular a ser agarrado pela garra de dois dedos. Se o ângulo de rotação da garra for sempre o mesmo, por exemplo,  $0^\circ$  e se os objetos estiverem inclinados em relação aos dedos da garra, podem ocorrer situações conforme a ilustrada na figura [4.2].

Esta situação, ilustrada na figura [4.2], é uma situação inadequada pois os dedos entram em contato com o objeto pelos seus vértices, conseqüentemente, devido ao agarramento instável, no decorrer do transporte, o objeto pode deslizar entre os dedos, resultando na sua queda.

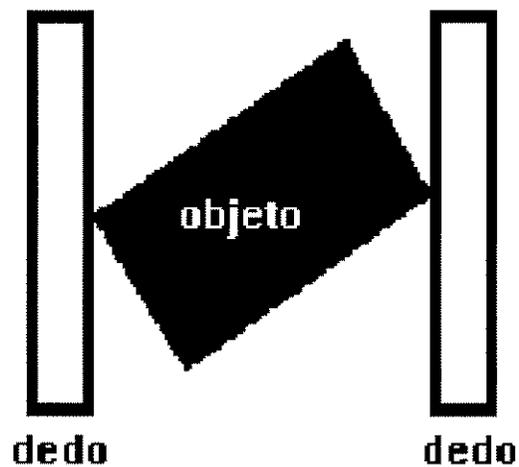


FIG. 4.2 OBJETO INCLINADO EM RELAÇÃO AOS DEDOS DA GARRA

Para evitar que aconteça uma situação como esta é que normalmente os objetos a serem agarrados são posicionados e orientados sempre da mesma maneira por meio de um mecanismo auxiliar de posicionamento e orientação, denominada de *loader*. Entretanto, o uso do *loader* prejudica a flexibilidade do robô pois quando a forma física do objeto é muito diferente daquela para a qual o *loader* foi projetado, não é mais assegurado o posicionamento e a orientação deste objeto, tornando este dispositivo de uso muito restrito.

Com a utilização do sensor de visão que fornece as informações das coordenadas do centro do objeto e a sua orientação, o robô pode posicionar e orientar a garra para que esta agarre o objeto da forma mais segura, ou seja, agarrar o objeto pelo seu menor *diâmetro* que passa pelo centro geométrico do objeto (figura [4.3]).

A possibilidade de identificar um objeto é um recurso bastante útil pois permite a classificação dos objetos. Vejamos um exemplo de aplicação onde este recurso é útil.

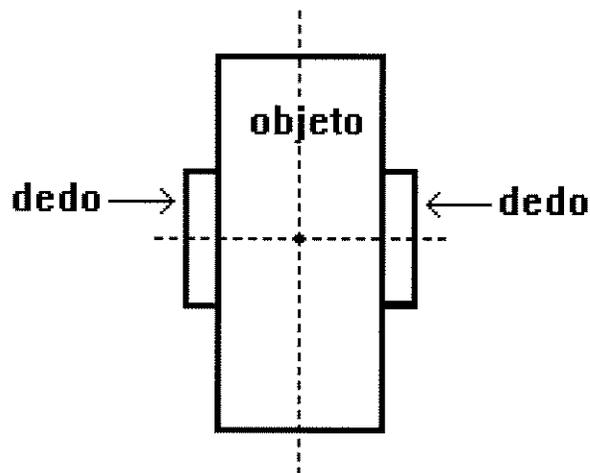


FIG. 4.3 AGARRAMENTO SEGURO DE UM OBJETO

Suponha que num determinado setor da indústria, uma correia transportadora transporta diversos tipos de objetos para serem separados no final da correia. Então o sensor de visão pode identificar os objetos conforme ilustrado na figura [4.4].

O presente trabalho foi limitado para um único objeto presente no campo visual da câmera de TV, sem sombra, através de iluminação adequada, e que o objeto possua apenas uma borda fechada, ou seja, não possua furos.

O reconhecimento das formas dos múltiplos objetos num campo visual, assim como a determinação individual das coordenadas de seus centros e as suas respectivas orientações são objetivos de estudos futuros.

A seguir são descritas as técnicas utilizadas para o processamento da imagem.

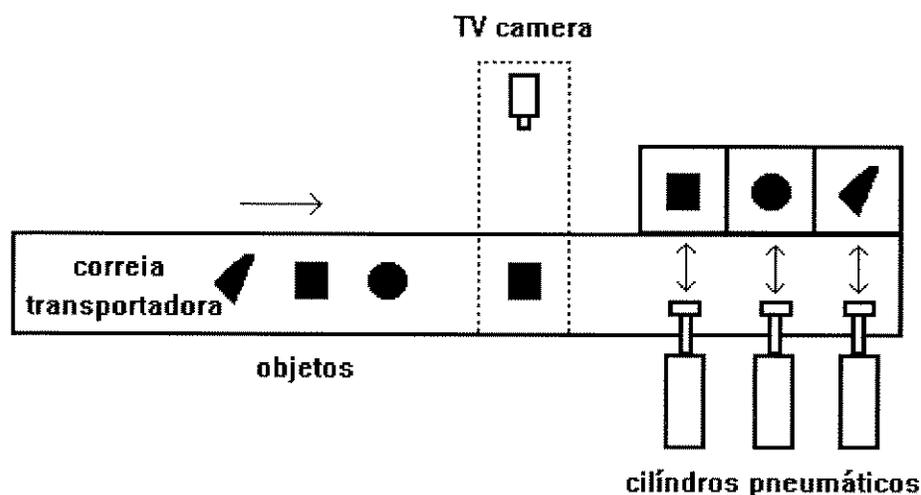


FIG. 4.4 APLICAÇÃO DO SENSOR DE VISÃO

#### 4.3.1 Cálculo do Centro Geométrico do Objeto.

Um dos objetivos deste trabalho é fazer a localização do objeto, visando obter a coordenada cartesiana real que a garra deva atingir, aumentando desta forma a flexibilidade do robô, pois sem esta ferramenta, o objeto a ser agarrado deverá estar em posição previamente determinada.

A determinação do centro geométrico do objeto é realizada exatamente com esta finalidade, e também para o cálculo da função característica do objeto, que é fundamental para o reconhecimento do mesmo.

Para que se possa calcular corretamente esse centro é necessário realizar uma correção na relação de escala nos eixos X e Y da imagem capturada. Esta correção é uma compressão vertical da imagem, visto que o objeto fica mais alongado nesta direção.

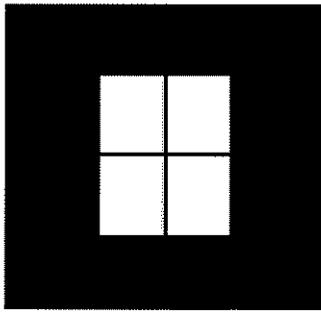
A imagem da tela de uma câmera de TV no padrão NTSC tem a dimensão proporcional de 3:4 sendo 3 para vertical e 4 para

a horizontal. Esta tela é composta de 525 linhas de varredura horizontal dividida em 262,5 linhas pares e 262,5 linhas ímpares de varredura, e no presente trabalho foram utilizadas apenas linhas pares. Por outro lado, uma varredura horizontal dará 63,5  $\mu$ s e como a amostragem é realizada com período de 0,25  $\mu$ s, uma varredura horizontal contém 254 *pixels*.

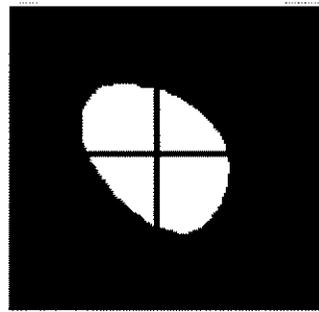
Para facilitar o raciocínio, supõe-se que a câmara esteja vendo uma figura de retângulo real de 300 mm (vertical) x 400 mm (horizontal) preenchendo completamente a tela. Esta figura é dividida em 254 *pixels* na horizontal e 262,5 *pixels* na vertical, ou seja, a relação entre a dimensão  $X_{REAL}$  e *pixel* na horizontal é de 1,57480315 mm/*pixel* e a relação  $Y_{REAL}$  e *pixel* na vertical é de 1,142857143 mm/*pixel*. Isto significa que uma figura real, após a digitalização, ocupa mais *pixels* na vertical do que na horizontal, para a mesma dimensão na horizontal e na vertical, havendo a necessidade de compressão no eixo vertical para corrigir a escala.

Após a realização da correção, faz-se a varredura horizontal da imagem adquirida (e corrigida), computando-se todos os *pixels* pretos, isto é, determina-se a área do objeto. Para determinar a coordenada Y do centro do objeto, divide-se a área encontrada em quantidade de *pixels* por dois, e repete-se a varredura horizontal da figura, desta vez identificando-se a linha onde a quantidade de *pixels* é igual a metade da área do objeto. Esta linha corresponde a coordenada Y do centro do objeto. Para determinar a coordenada X do centro do objeto, é feita uma varredura vertical localizando a coluna onde a quantidade de *pixels* acumulada é igual a metade da área do objeto.

As figuras [4.5a] e [4.5b] mostram as imagens capturadas de dois objetos com as linhas e as colunas que dividem a área do objeto pela metade, em vertical e horizontal respectivamente, e portanto a interseção destas indica as coordenadas do centro dos objetos.



[4.5a]



[4.5b]

FIG. 4.5 LOCALIZAÇÃO DO CENTRO DA FIGURA

A determinação da relação entre a quantidade de *pixels* com a dimensão real do objeto foi feita através do seguinte procedimento de calibração:

Foram colocados quatro objetos circulares de diâmetro 50 mm nos 4 vértices de um retângulo de lado  $\overline{ABCD}$  sendo  $A=C=90$  mm e  $B=D=70$  mm e foram calculadas as coordenadas dos centros dos objetos pelo sensor de visão que foram (51,88), (58,38), (117,95) e (124,44), correspondendo as coordenadas dos vértices do retângulo visto pelo sensor de visão. Portanto, os comprimentos dos lados na escala do sensor de visão são:

$$A = 66,37$$

$$B = 50,80$$

$$C = 66,27$$

$$D = 51,48$$

o que nos dá uma razão média de 1,358. Então, para converter a escala do sensor de visão para a escala real é só multiplicar por este valor.

As diferenças entre o valores de  $A$  e  $C$  e de  $B$  e  $D$  que deveriam ser nulas, deve-se ao fato do plano da lente da câmera

de TV não estar exatamente em paralelo com o plano onde estão colocados os objetos para calibração.

A relação geométrica entre o sistema de coordenadas do sensor de visão e o sistema de coordenadas cartesianas da base do robô foi determinada da seguinte maneira:

Inicialmente foi colocado um objeto circular de diâmetro de 50 mm com seu centro localizado nas coordenadas  $x_{CAL, BR}$ ,  $y_{CAL, BR}$  e  $z_{CAL, BR}$  conhecidas do braço e o sensor de visão calculou as coordenadas do centro deste objeto visto pelo sistema de coordenadas do sensor ( $x_{CAL, VIS}$ ,  $y_{CAL, VIS}$ ). Como o sistema de coordenadas do braço e o sistema de coordenadas da visão tem a relação mostrada na figura [4.6], a conversão das coordenadas de um ponto  $P(x_{VISÃO}, y_{VISÃO})$  do sistema de coordenadas do sensor de visão para sistema de coordenadas cartesianas da base é:

$$X = x_{CAL, BR} + (x_{CO, VIS} - x_{CAL, VIS}) \cdot 1,358$$

$$Y = y_{CAL, BR} + (y_{CO, VIS} - y_{CAL, VIS}) \cdot 1,358$$

onde:

$x_{CAL, BR}$ ,  $y_{CAL, BR}$  coordenadas do centro do objeto no ato de calibração visto pelo sistema de coordenadas cartesianas do braço.

$x_{CAL, VIS}$ ,  $y_{CAL, VIS}$  coordenadas do centro do objeto no ato de calibração visto pelo sistema de coordenadas do sensor de visão.

$x_{CO, VIS}$ ,  $y_{CO, VIS}$  coordenadas do centro do objeto a ser agarrado visto pelo sistema de coordenadas do sensor de visão.

$x_{CO, BR}$ ,  $y_{CO, BR}$  coordenadas do centro do objeto a ser agarrado visto pelo sistema de coordenadas cartesianas do braço.

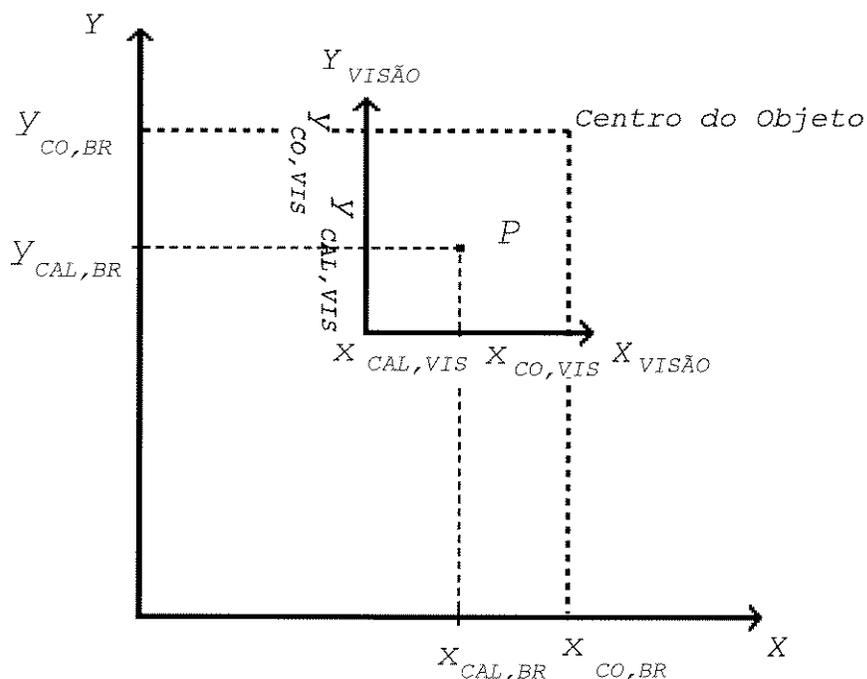


FIG.4.6 RELAÇÃO ENTRE O SISTEMA DE COORDENADA DO BRAÇO E DO SENSOR DE VISÃO

#### 4.3.2 Detecção da Borda do Objeto

A informação sobre o contorno da imagem do objeto é fundamental para poder representá-lo e, conseqüentemente, reconhecê-lo através de análise computacional.

Os *pixels* da borda são determinadas da seguinte forma: as coordenadas do centro geométrico do objeto são consideradas como as coordenadas do ponto inicial do segmento de reta que será calculado e o outro extremo do segmento serão as coordenadas referentes aos *pixels* da fronteira.

O primeiro *pixel* da borda é procurado considerando a área do objeto, já conhecida, como a área de um círculo e então calcula-se o raio  $r$  deste círculo. Este procedimento é

feito para que a procura de borda não tenha que começar na origem que foi convencionada como sendo o centro da área do objeto, e sim de um ponto provavelmente mais próximo da borda.

Calcula-se, a partir daí, as coordenadas da borda, através das relações:

$$X_B = \text{centro } X + r \cos \theta$$

$$Y_B = \text{centro } Y + r \sin \theta$$

onde  $X_B$  e  $Y_B$  são as coordenadas  $X$  e  $Y$  da borda e  $\theta$  possui valor inicial nulo. Se o *pixel* daquelas coordenadas é preto (significa que aquele ponto ainda está dentro do objeto), incrementa-se o raio até encontrar um *pixel* branco, onde caracteriza-se a borda. Se o *pixel* for branco, significa que aquele ponto já se encontra fora do objeto, decrementa-se o raio até encontrar um *pixel* preto.

Após encontrar a borda, armazena-se o raio, incrementa-se  $\theta$  e repete-se o processo, considerando agora como valor inicial do raio o mesmo valor do raio encontrado anteriormente. Esta consideração é baseada na premissa de que, como o incremento do ângulo  $\theta$  é pequeno, há grande possibilidade do raio ter um valor próximo ao raio anterior.

#### 4.3.3 Determinação da Função Característica do Objeto.

Como preparação para identificação da forma do objeto, é feita a conversão da imagem do objeto, referenciada no sistema de coordenadas cartesianas do sensor de visão, para o sistema de coordenadas polares com origem coincidente com o centro geométrico da imagem. Confecciona-se um gráfico que indica as distâncias radiais do contorno do objeto até o centro geométrico  $R(\theta)$  e que vamos denominar de contorno rastreado versus ângulo de varredura  $\theta$  (Figura [4.7]).

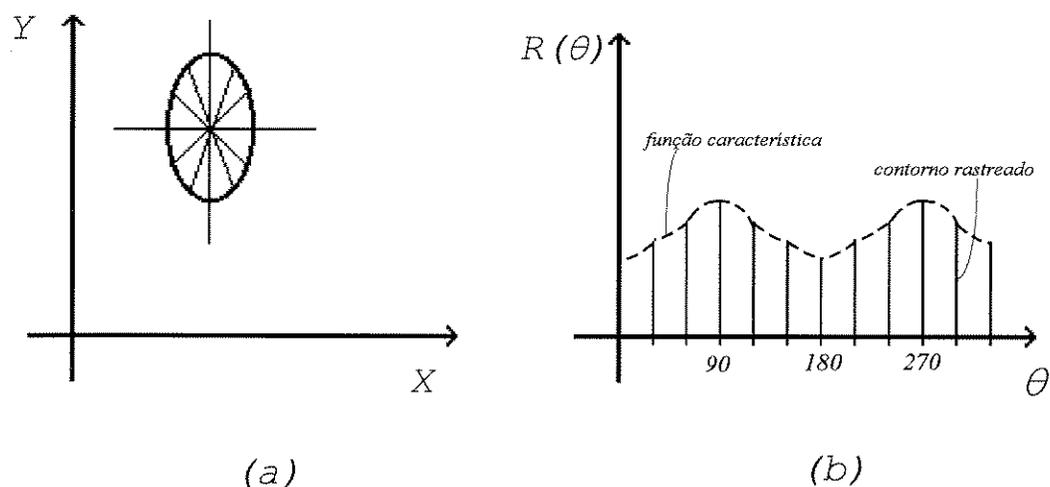


FIG. 4.7 CONTORNO RASTREADO E FUNÇÃO CARACTERÍSTICA DE UM OBJETO

Observando atentamente a figura [4.7] nota-se que se o ângulo de varredura varia além dos  $360^\circ$ , haverá uma repetição do envelope, formando-se uma função periódica em  $\theta$ , aonde o primeiro período representa a forma do objeto. Ao mudar a orientação do objeto haverá um deslocamento da função em relação a  $\theta$ , significando a mesma função defasada. Denomina-se aqui a figura [4.7b] de contorno rastreado do objeto e a sua função periódica formada pelo comprimento dos raios, de função característica do objeto (FCO), pois se o incremento  $\Delta\theta$  da varredura for infinitesimalmente pequeno, teríamos uma função contínua em  $\theta$ .

A escolha do valor de  $\Delta\theta$  depende da forma do objeto que se pretende analisar pois é conhecido pela teoria de amostragem que quando um sinal em função de tempo é amostrado com o período de amostragem  $T_s$ , a frequência máxima  $f_{MAX}$  reconstituível é de:

$$f_{MAX} = \frac{1}{2 T_s}$$

Transferindo esta teoria para o presente caso,  $\Delta\theta$  corresponde ao  $T_s$  e a componente de frequência num sinal em função de tempo corresponde à forma do objeto, como poderá ser observado nas figuras [4.8] a [4.11].

O valor de  $\Delta\theta$  foi determinado, na prática, definindo-se inicialmente as formas dos objetos que se pretende analisar e posteriormente experimentando o valor máximo de  $\Delta\theta$  que resultasse na análise sem erro. Por outro lado, como no presente trabalho foi dada a prioridade no processamento em tempo real, com o microcomputador possível de ser adquirido (CPU 386 SX de 20 MHz com coprocessador 387) o valor do  $\Delta\theta$  foi definido em  $1,40625^\circ$  levando em consideração as formas dos objetos e a velocidade de processamento do computador e testando-se na prática que este valor satisfaz para análise dos objetos em questão.

As figuras [4.8a], [4.8b] e [4.8c] ilustram a seqüência como são calculados os raios da imagem do objeto e a figura [4.8d] mostra o contorno rastreado daquele objeto. As figuras [4.9a], [4.9b], [4.9c] e [4.9d] ilustram a seqüência de varredura e o contorno rastreado do mesmo objeto da figura [4.8], colocado com uma determinada rotação. Pode-se observar que os contornos rastreados dos objetos estão defasados de acordo com a mudança na orientação do objeto.

As figuras [4.10] a [4.11] ilustram os contornos rastreados de diversos objetos, de formas distintas, para que possam ser observadas as diferenças nas suas funções características.

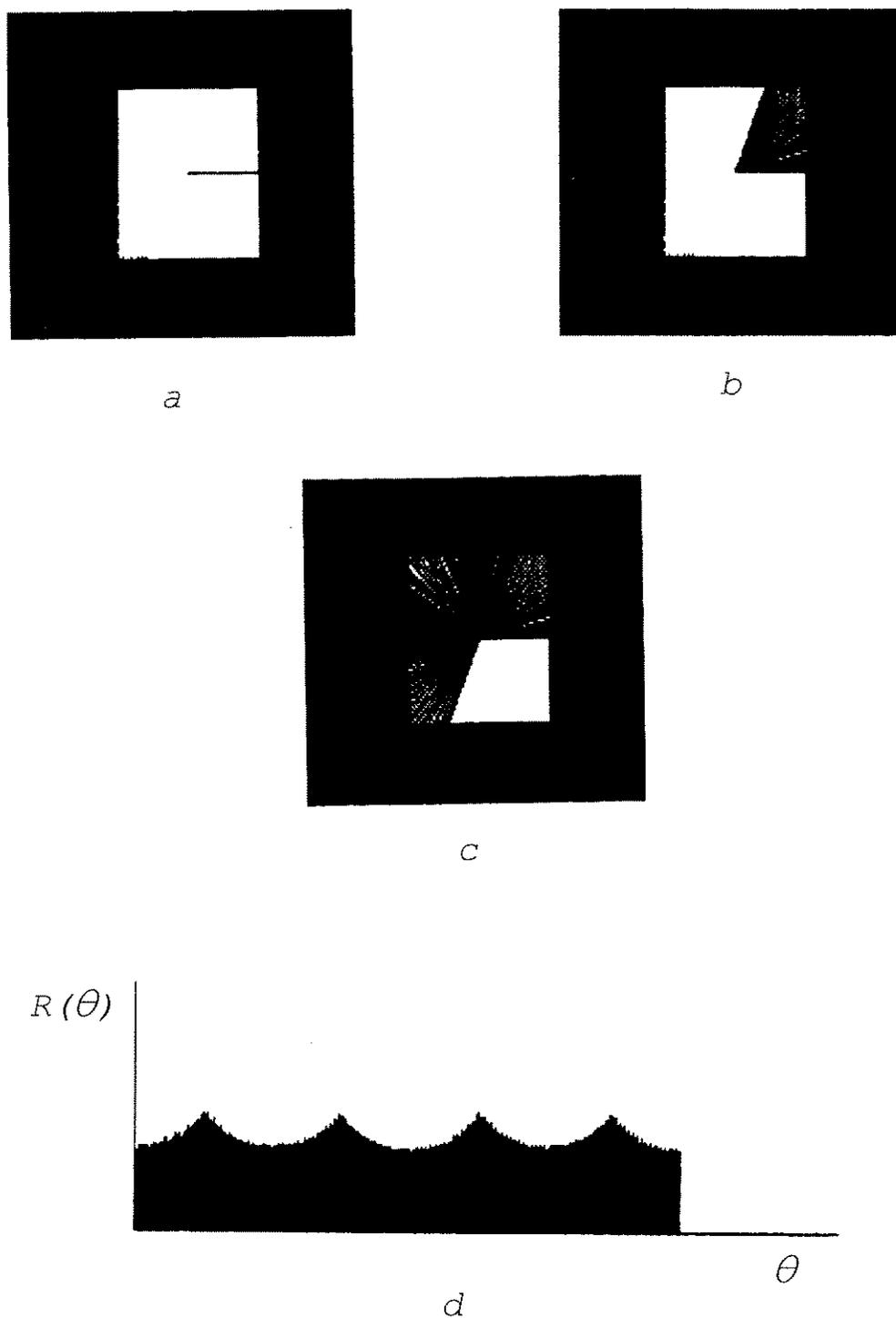


FIG. 4.8 (a), (b) e (c) SEQUÊNCIA DE CÁLCULO DOS RAIOS  
(d) CONTORNOS RASTREADOS DO OBJETO

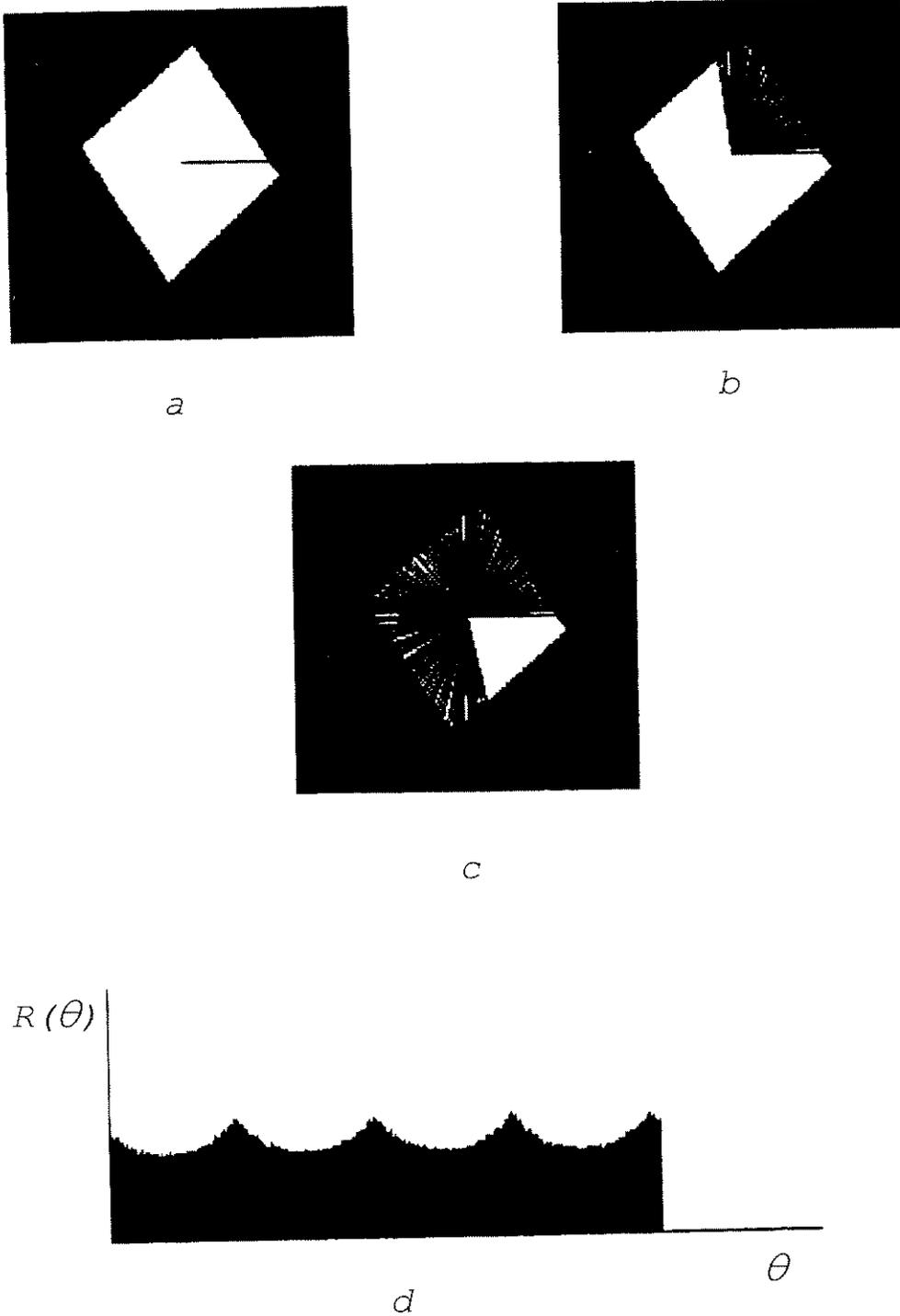


FIG. 4.9 (a), (b) e (c) SEQUÊNCIA DE CÁLCULO DOS RAIOS  
(d) CONTO RASTREADO DO OBJETO

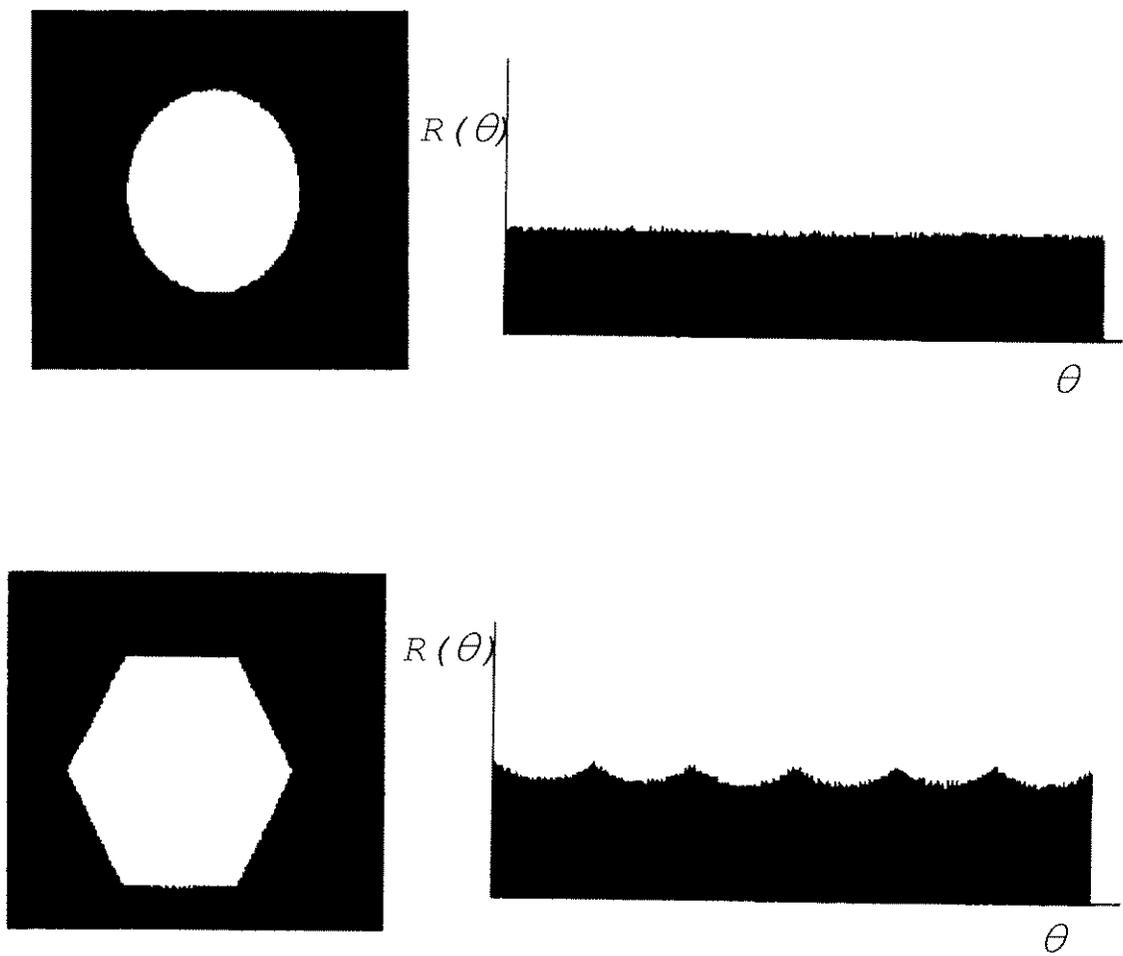


FIG. 4.10 CONTORNO RASTREADO DOS OBJETOS

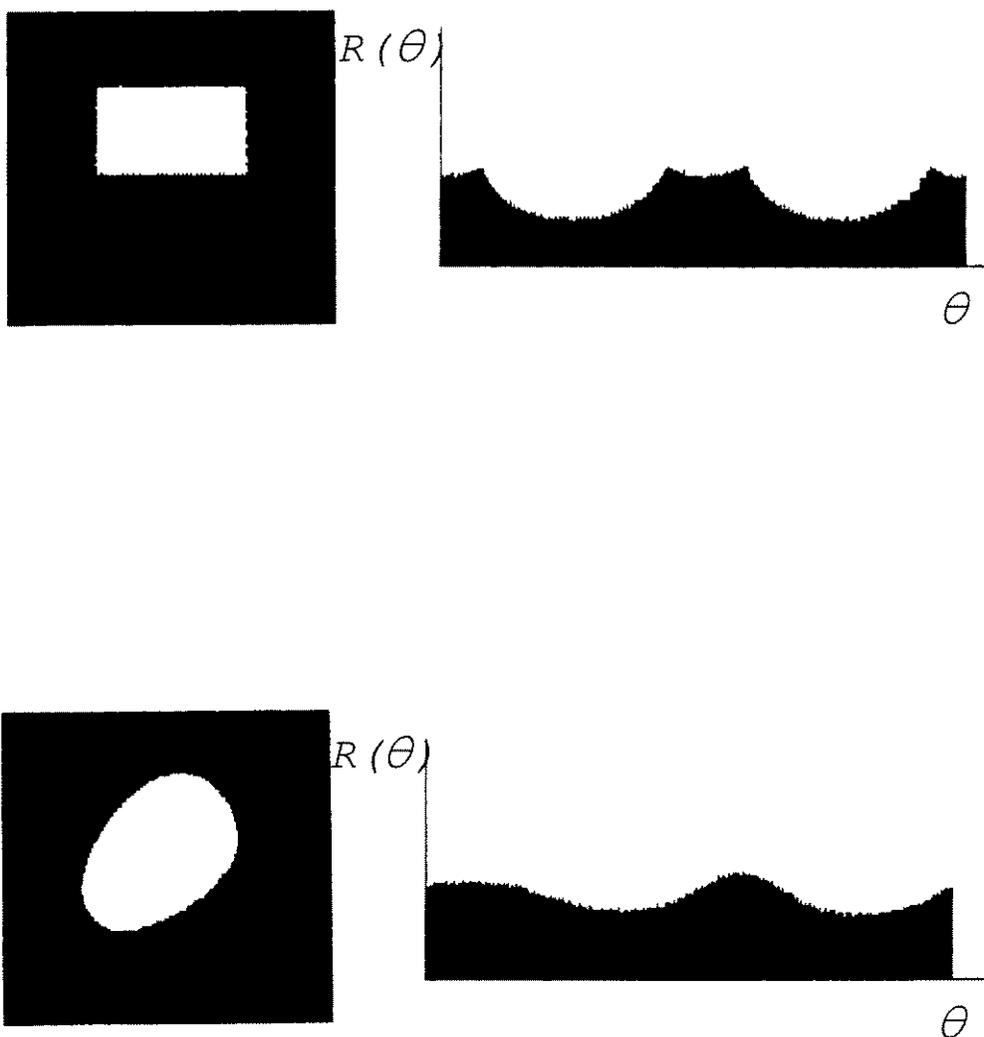


FIG. 4.11 CONTORNO RASTREADO DOS OBJETOS

#### 4.3.4 Reconhecimento do Objeto.

O reconhecimento do objeto foi feito através da comparação entre o contorno rastreado do objeto a ser reconhecido e os contornos rastreados de diversos objetos previamente construídos.

Como estes contornos rastreados podem ser considerados como um período da função amostrada, onde a própria função é definida pela forma do objeto, e que o deslocamento em  $\theta$  é dependente da orientação do objeto, como foi visto anteriormente, o reconhecimento do objeto é feito aproveitando-se a seguinte propriedade da *Transformada Discreta de Fourier* de uma função.

A *Transformada Discreta de Fourier* (*Discrete Fourier Transform - DFT*)  $X_k$  de uma seqüência  $x_m$  de  $N$  termos é definida como [20]

$$X_k = \sum_{m=0}^{N-1} x_m W^{mk}$$

onde:

$$k=0, \dots, N-1$$

$$W = e^{-j \frac{2\pi}{N}}$$

e a *DFT* desta mesma função amostrada deslocada no ângulo de varredura  $\ell$  é:

$$\left\{ X_{m+\ell} \right\} \stackrel{DFT}{\longleftrightarrow} \left\{ W^{-\ell k} X_k \right\}$$

ou seja, quando as amostras estão deslocadas de  $\ell$  em relação as amostras padrões, o módulo da *DFT* permanece inalterado havendo apenas a alteração em fase correspondente ao deslocamento das amostras.

Foi utilizado o algoritmo de *Fast Fourier Transform* (*FFT*) para calcular a *DFT* da função característica do objeto amostrada.

Desta forma, o reconhecimento do objeto é feito através da comparação do módulo da *FFT* da *FCO* amostrada com os módulos das *FFT's* dos objetos previamente registrados na memória.

Em termos práticos, o que foi realizado é o seguinte: Inicialmente é feita a digitalização e aquisição da imagem de todos os objetos que devem ser registrados com as orientações que correspondam ao ângulo de  $0^\circ$  de rotação da garra para os seus agarramentos. Em seguida, é formado um banco de dados contendo as informações dos objetos e as *Transformadas Discretas de Fourier* das *FCO* amostradas daqueles objetos.

Ao realizar a digitalização e a aquisição de um objeto a ser identificado, calcula-se a *Transformada Discreta de Fourier*, através da *FFT*. Posteriormente é comparado o módulo desta *Transformada* com o que está armazenado no banco de dados e determina-se o erro médio quadrático entre eles. Se o erro for menor que  $0,07\%$ , o objeto é considerado identificado.

Teoricamente o objeto é considerado identificado se o erro for igual a zero mas devido o ruído na aquisição da imagem o mesmo objeto pode apresentar um erro maior que zero. Foram detectados, através das aquisições repetidas da imagem de um mesmo objeto, que este erro pode atingir o valor máximo de  $0,07\%$ .

Este processo de identificação do objeto pela comparação dos módulos da *FFT* tem-se mostrado eficiente. Entretanto é um processo inevitavelmente demorado uma vez que, no pior caso, precisa comparar o módulo da *FFT* do objeto em análise com o módulo da *FFT* de todos os objetos registrados no banco de dados. Convém notar que a comparação de um objeto com outro requer no mínimo 256 subtrações, 256 potenciações de 2 e 256 adições.

Desta forma foi adotado um método de pré-seleção mais grosseiro e rápido antes de se realizar a identificação detalhada através da comparação dos módulos da *FFT* das *FCO* amostradas, acelerando assim o processo de identificação. Para esta pré-seleção foi utilizado o fator *AC* do contorno rastreado, o qual é calculado através das seguintes equações:

$$E_t = \text{Energia Total} = \sum_{i=0}^{N-1} \frac{CR_i^2}{N}$$

onde  $CR_i$  é uma seqüência de amostras da característica do objeto.

$$V_{DC} = \text{Valor DC} = \sum_{i=0}^{N-1} \frac{CR_i}{N}$$

$$E_{DC} = \text{Energia DC} = (\text{Valor DC})^2$$

$$\text{Fator AC} = \frac{E_t - E_{DC}}{E_{DC}} = \left( \frac{E_t}{E_{DC}} - 1 \right) \cdot 100 \%$$

Em outras palavras, o fator *AC* representa o objeto através das informações do contorno rastreado do objeto que estão além do raio médio.

Para montar a biblioteca foi calculado várias vezes o fator *AC* de cada objeto e determinou-se o intervalo possível deste fator para cada objeto. A tabela [4.1] ilustra os intervalos verificados experimentalmente e os armazenados na biblioteca.

OBJETO	FATOR AC (%)	
	VALOR EXPERIMENTAL	VALOR UTILIZADO
QUADRADO	[1,069; 1,185]	[1,000; 1,600]
RETÂNGULO	[7,220; 8,840]	[7,000; 9,000]
CÍRCULO	[<0,047]	[<0,050]
OCTÓGONO	[0,065; 0,083]	[0,055; 0,095]
HEXÁGONO	[0,191; 0,206]	[0,170; 0,220]
OVAL	[2,280; 2,570]	[2,000; 2,800]

TABELA 4.1 FATOR AC PARA DIVERSOS OBJETOS

Desta forma, encontra-se o fator AC da função característica amostrada do objeto que está sendo analisado e realiza-se a identificação grosseira do objeto através da verificação do fator AC do objeto em análise com os que estão registrados no banco de dados; e após esta identificação calcula-se o erro médio quadrático entre o módulo da *FFT* do objeto localizado grosseiramente, diminuindo assim o tempo de processamento.

Esta identificação grosseira pode ser feita por outros métodos, tal como através da diferença percentual entre o valor *RMS* do contorno rastreado (*CR*) e o raio do círculo de área idêntica à da imagem do objeto, ou seja, através das seguintes equações:

$$\text{Fator de distinção} = \frac{CR_{RMS} - r}{r} \times 100\%$$

$$CR_{RMS} = \sqrt{\sum_{i=1}^{N-1} \frac{CR_i^2}{N}}$$

onde: *r* é o raio do círculo.

Uma outra forma de identificação grosseira pode ser feita pela quantidade dos vértices da imagem.

#### 4.3.5 Normalização

Foi especificada como uma das características deste sensor de visão a identificação de objetos de tamanhos diferentes mas de mesmo formato, tais como dois objetos de formato circular com raios diferentes ou dois retângulos de tamanhos diferentes com a mesma proporção entre o lado menor e lado maior.

Para atingir esta característica o contorno rastreado do objeto é normalizado antes de se calcular a *Transformada Discreta de Fourier*.

#### 4.3.6 Determinação da Orientação do Objeto.

Para que o robô consiga agarrar um objeto de forma segura, por exemplo, pelo menor *diâmetro* do objeto, é necessário que se tenha a informação da orientação do mesmo.

O cálculo do ângulo de inclinação do menor *diâmetro* do objeto é um procedimento simples uma vez que os seus raios já foram calculados anteriormente.

Por outro lado, sabendo que a correlação entre duas funções determina o grau de semelhança entre elas, de quanto uma função está defasada em relação a outra e como os objetos estão sendo representados por funções discretas, a correlação discreta pode ser utilizada para determinar de quanto o objeto está defasado através das suas funções características.

A correlação discreta entre funções pode ser calculada através da *Transformada Discreta Inversa de Fourier* (*DFT* inversa) que já tem o seu algoritmo implementado, bastando passar como parâmetro  $inv = -1$ . Desta forma, multiplica-se a transformada armazenada na biblioteca pelo conjugado da transformada da função característica amostrada do objeto cuja orientação deseja-se calcular. A correlação é a *Transformada Discreta Inversa de Fourier* deste produto.

O resultado da correlação representa de quanto a orientação do objeto está defasada da sua orientação quando foi armazenado no banco de dados. Este defasamento é identificado verificando-se qual a posição do primeiro pico da correlação, onde cada posição está relacionada por  $1,40625^\circ$  que corresponde a  $\Delta\theta$ , ou seja,  $360^\circ/256$ . As figuras [4.12] e [4.13] mostram o resultado da correlação para objeto não rotacionado e o mesmo objeto rotacionado, respectivamente.

Comparando estes dois métodos, pode-se notar que cada um tem as suas vantagens e desvantagens. O método de determinação da orientação pela técnica de correlação é muito útil quando um objeto, por alguma razão, não pode ser agarrado pelo seu menor *diâmetro*; isto é possível de acontecer num processo industrial onde a próxima máquina para onde o objeto é carregado, deve receber o objeto agarrando-o pelo seu menor *diâmetro* tendo, portanto, uma orientação definida a priori com a qual a garra do robô deve agarrá-lo. Num caso como este, ao construir um banco de dados, este objeto é colocado da tal forma que o ângulo da garra para aquela orientação seja de  $0^\circ$ .

A desvantagem deste método é que, quando, por alguma razão, houver falha na identificação do objeto, não se terá condições de calcular a sua orientação.

Assim, no presente sensor de visão, foram implementados esses dois métodos de cálculo de orientação.

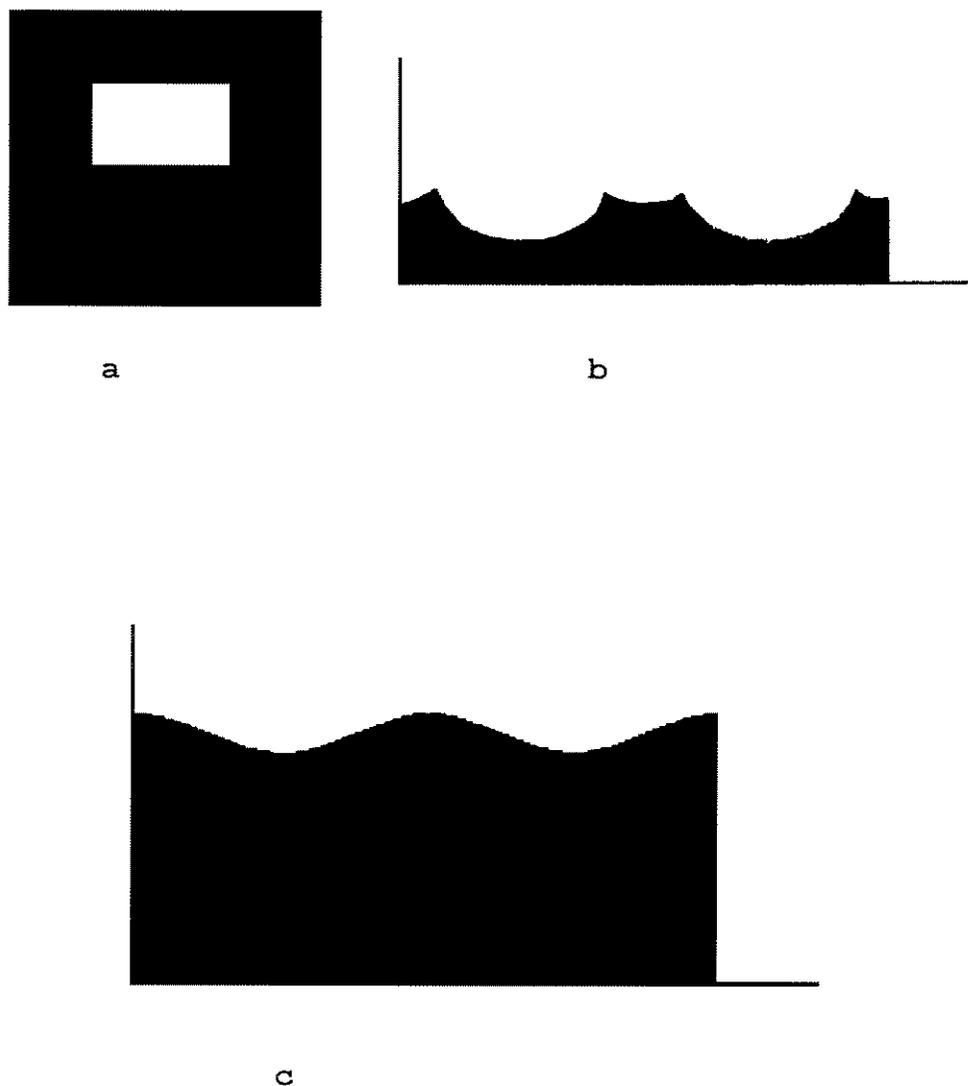
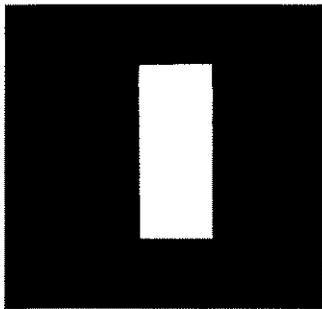
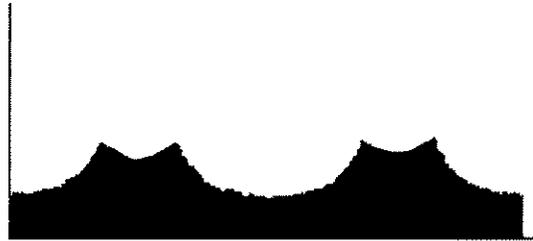


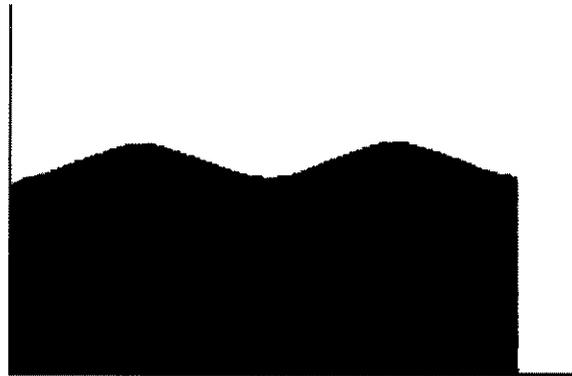
FIG. 4.12 a) IMAGEM DO OBJETO NÃO DEFASADO  
b) HISTOGRAMA DO OBJETO  
c) RESULTADO DA CORRELAÇÃO  
DEFASAMENTO DE 0°



a



b



c

FIG 4.13 a) IMAGEM DO OBJETO DEFASADO  
b) HISTOGRAMA DO OBJETO  
c) RESULTADO DA CORRELAÇÃO  
DEFASAMENTO DE  $90^\circ$

#### 4.4 RESULTADOS OBTIDOS.

O sistema foi testado em laboratório e apresentou resultados satisfatórios. A imagem capturada pela TV câmera é mostrada na tela do computador e processada em tempo real como era desejado no projeto.

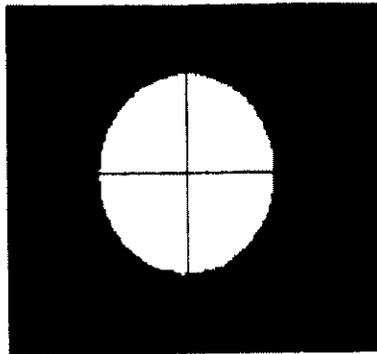
Para evitar que a sombra do objeto cause a aquisição da imagem errada, foi construída uma mesa de luz sobre a qual o objeto é colocado para monitoramento.

Foi montada uma biblioteca com a informação de seis diferentes formas de objetos diferentes:

4. círculo;
5. hexágono;
6. octógono;
7. quadrado;
8. retângulo;
9. forma irregular com formato oval.

O *software* de reconhecimento se mostrou bastante eficiente. Todos os objetos foram identificados, tendo sido determinado também suas coordenadas de centro e a orientação deles em relação à informação armazenada.

As figuras [4.14] a [4.24] mostram o *hardcopy* da tela do computador para diversos objetos.

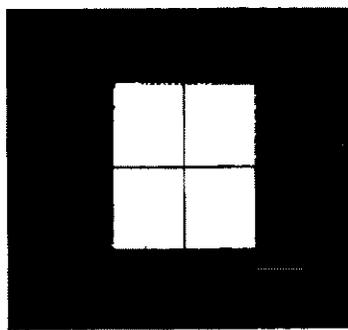


CENTRO DO OBJETO

CENTRO-X=76    CENTRO-Y=61

O OBJETO EH UM CIRCULO

FIG. 4.14 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

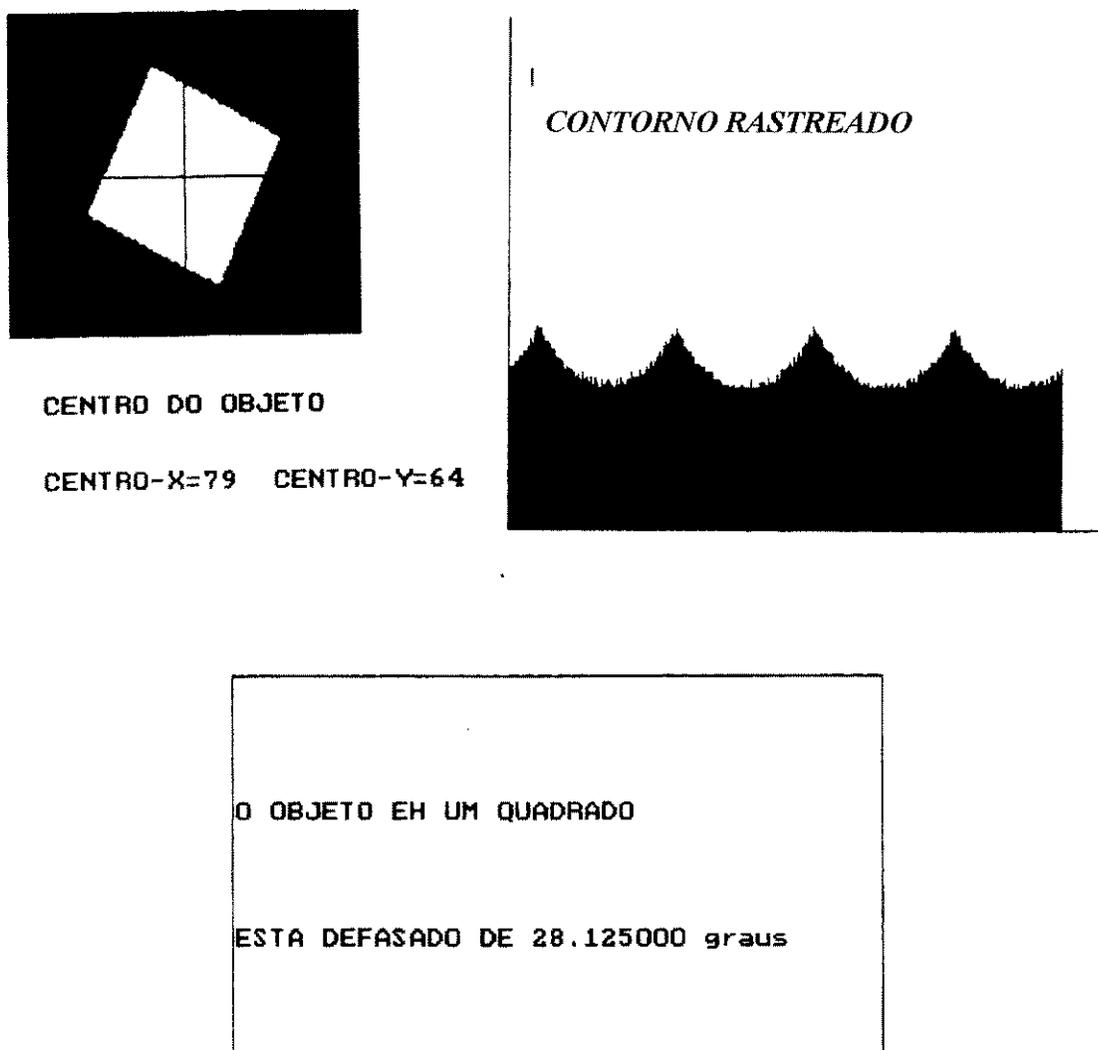
CENTRO-X=81 CENTRO-Y=62



O OBJETO EH UM QUADRADO

ESTA DEFASADO DE 0.000000 graus

FIG. 4.15 *HARDCOPY* DA TELA DO COMPUTADOR

FIG. 4.16 *HARDCOPY* DA TELA DO COMPUTADOR

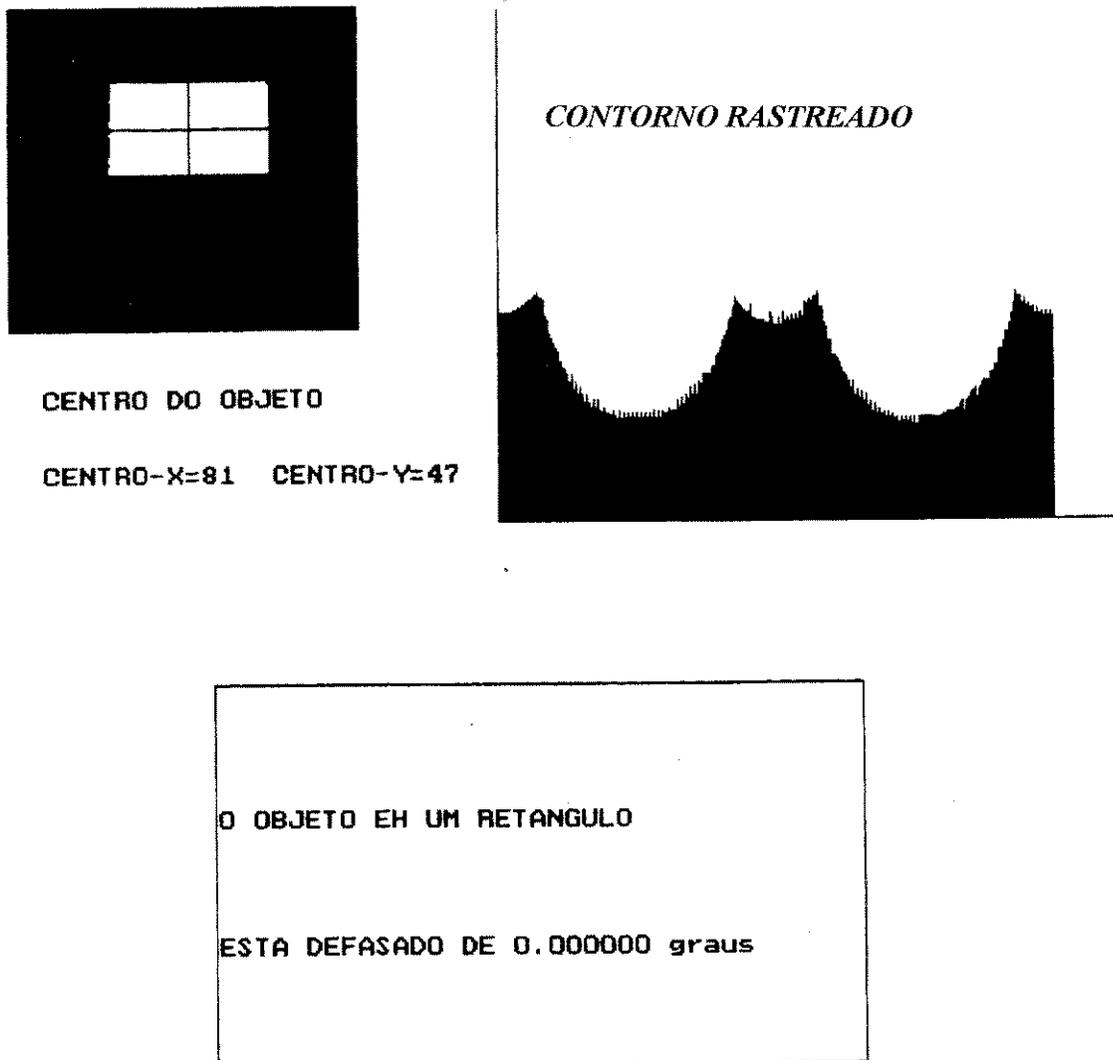
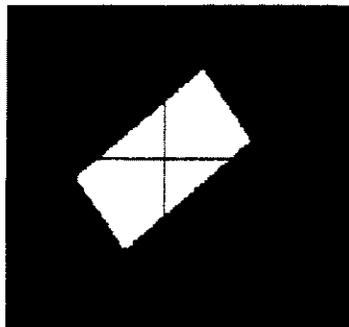


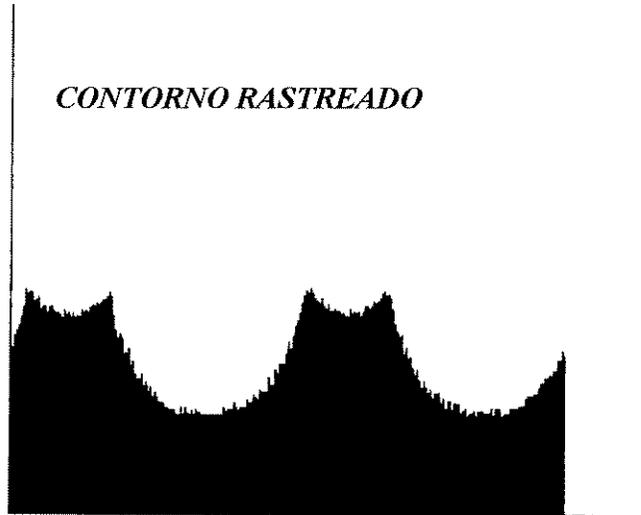
FIG. 4.17 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

CENTRO-X=72 CENTRO-Y=60

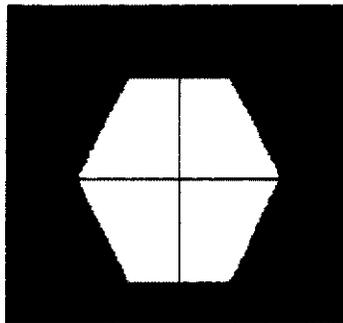
*CONTORNO RASTREADO*



O OBJETO EH UM RETANGULO

ESTA DEFASADO DE 144.843750 graus

FIG. 4.18 *HARDCOPY* DA TELA DO COMPUTADOR



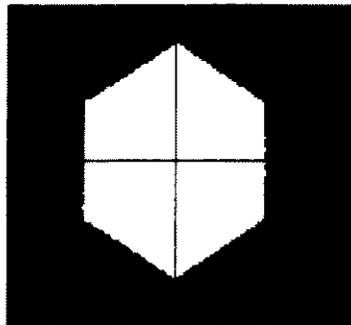
CENTRO DO OBJETO

CENTRO-X=80 CENTRO-Y=69



O OBJETO EH UM HEXAGONO  
  
ESTA DEFASADO DE 0.000000 graus

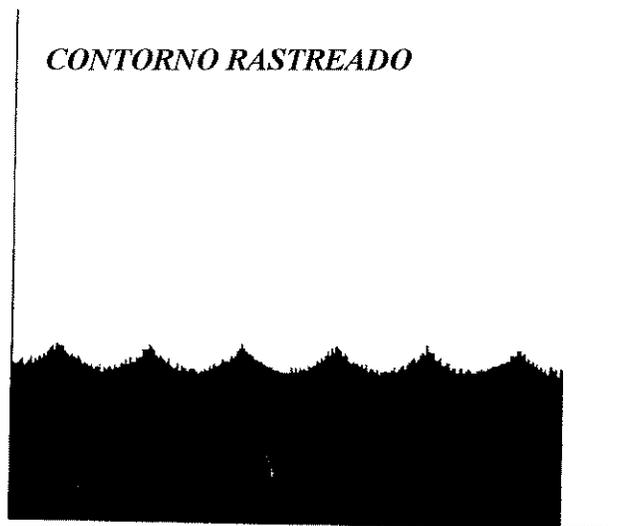
FIG. 4.19 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

CENTRO-X=76    CENTRO-Y=61

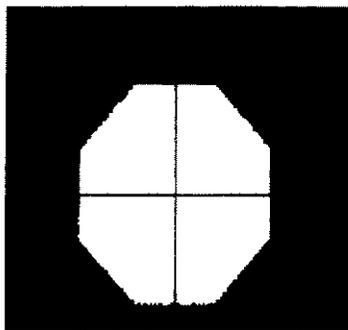
*CONTORNO RASTREADO*



O OBJETO EH UM HEXAGONO

ESTA DEFASADO DE 32.343750 graus

FIG. 4.20 *HARDCOPY* DA TELA DO COMPUTADOR



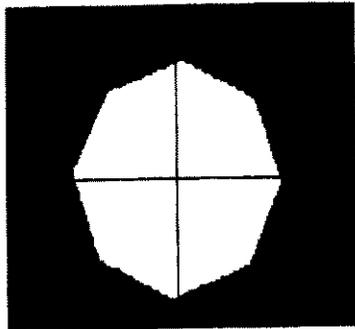
CENTRO DO OBJETO

CENTRO-X=77 CENTRO-Y=73



O OBJETO EH UM OCTOGONO  
ESTA DEFASADO DE 0.000000 graus

FIG. 4.21 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

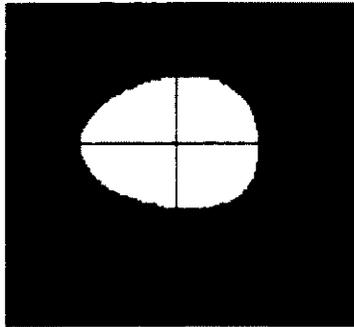
CENTRO-X=77 CENTRO-Y=68



O OBJETO EH UM OCTOGONO

*ESTA DEFASADO DE 25.312500 graus*

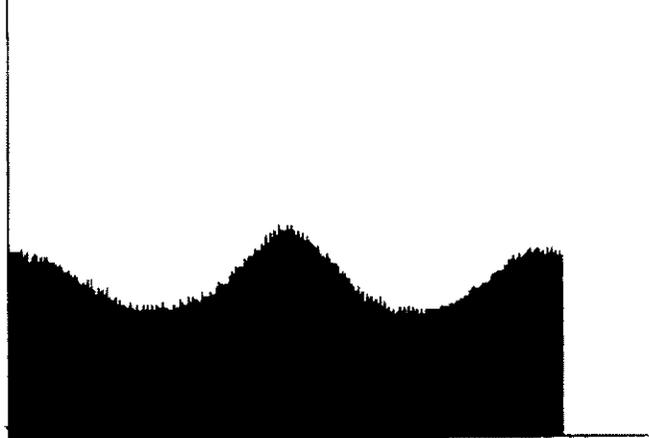
FIG. 4.22 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

CENTRO-X=76 CENTRO-Y=55

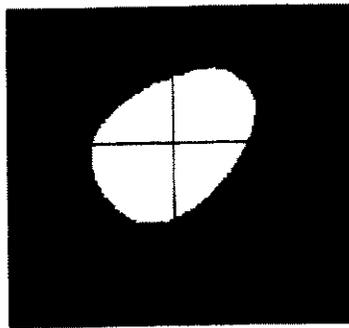
CONTORNO RASTREADO



O OBJETO EH UM OVO

*ESTA DEFASADO DE 0.000000 graus*

FIG. 4.23 *HARDCOPY* DA TELA DO COMPUTADOR



CENTRO DO OBJETO

CENTRO-X=76 CENTRO-Y=54

*CONTORNO RASTREADO*



O OBJETO EH UM OVO

ESTA DEFASADO DE 153.281250 graus

FIG. 4.24 *HARDCOPY* DA TELA DO COMPUTADOR

#### 4.5 FUTURA APLICAÇÃO DO SENSOR DE VISÃO

Uma das aplicações futuras do sensor de visão no controle de movimento do robô será a monitoração direta da posição da ponta do robô, sem ter que passar pelos mecanismos e sensores das juntas que introduzem erros, obtendo diretamente a posição da ponta do robô. São os seguintes os possíveis erros introduzidos na detecção da posição da ponta do robô através dos sensores das juntas:

10. Erros de medidas em cada junta;
11. Erro de modelagem direta;
12. Erro de calibração e alinhamento do mecanismo;
13. Erro do modelo computacional.

Uma experiência similar foi divulgada por FURUTA [8] utilizando um feixe de raio laser para monitorar o erro de rastreamento da ponta do robô.

Para esta aplicação, devem ser resolvidos alguns problemas:

- a. O sensor de visão deve ter, no mínimo 2 câmeras e 2 sistemas de processamento, pois a ponta do braço é posicionado no espaço tridimensional;
- b. A resolução do sensor de visão deve ser alta pois a precisão da detecção da posição da ponta do braço está diretamente relacionada com a resolução do sensor;
- c. Como a monitoração da posição da ponta do braço deve ser feita em tempo real, é necessário um computador com alta capacidade de processamento;
- d. É necessário algum arranjo na ponta do braço para que o sensor de visão possa distinguir facilmente aquele ponto.

Resolvidos esses problemas, o sistema de controle do robô passará, da configuração atual, figura [4.25], para a configuração mostrada na figura [4.26].

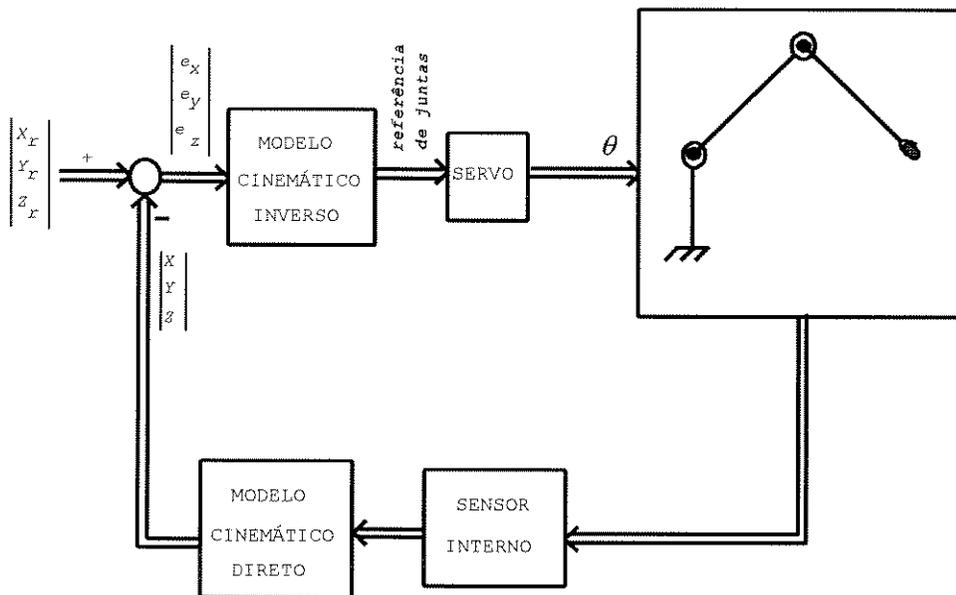


FIG. 4.25 SISTEMA DE CONTROLE SEM O SENSOR DE VISÃO

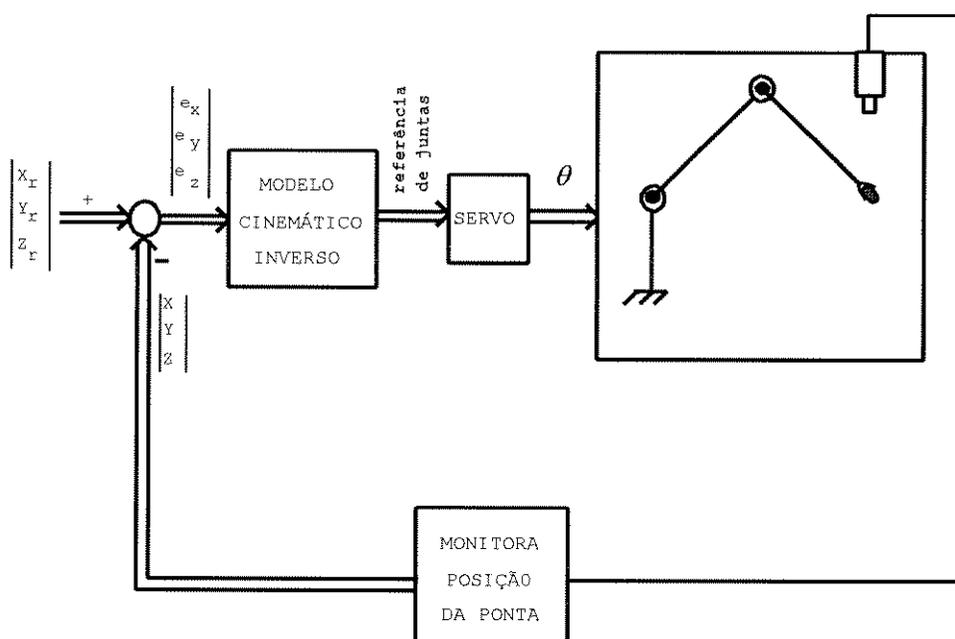


FIG. 4.26 SISTEMA DE CONTROLE COM O SENSOR DE VISÃO

## CAPÍTULO 5

# SISTEMA HIERÁRQUICO DE CONTROLE DE POSIÇÃO E ORIENTAÇÃO

### 5.1 INTRODUÇÃO.

O sistema de controle de um robô pode ser visto como a estrutura hierárquica [30] e [33] conforme a ilustração da figura [5.1], onde o controle das juntas representa o mais baixo nível hierárquico e o mais alto nível é o controle supervisor de tarefas.

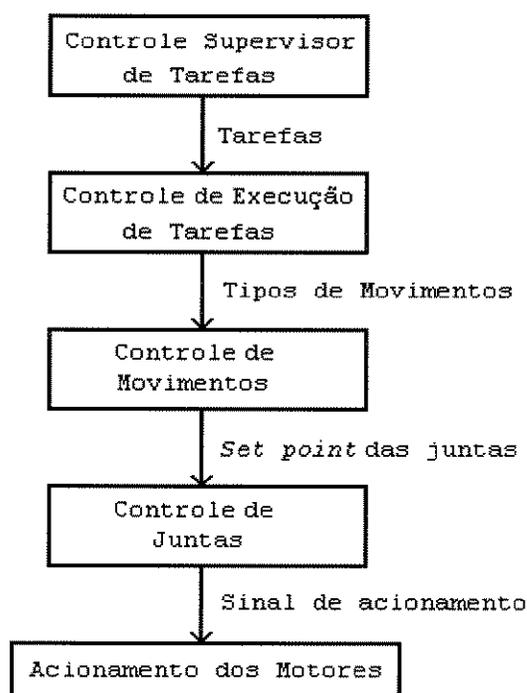


FIG. 5.1 ESTRUTURA HIERÁRQUICA DE CONTROLE DO ROBÔ

### 5.1.1 Controle Supervisor de Tarefas

Este nível de controle é responsável pelo controle de intertravamento entre o robô e os equipamentos periféricos tais como máquinas operatrizes a CNC (Comando Numérico Computadorizado).

A ação deste nível de controle é analisar qual ou quais equipamentos periféricos devem iniciar as suas tarefas, baseado nos sinais que indicam os estados atuais dos equipamentos periféricos, decidindo inclusive qual é a tarefa que o robô deve executar, seguindo a programação de seqüenciamento. Como resultado, o controle supervisor envia sinais de comando aos equipamentos periféricos e indica ao controle de execução da tarefa, a tarefa que o robô deve executar.

Como este nível de controle tem mais característica de controle seqüencial, para a automação de processos discretos complexos onde a análise dos processos a serem disparados é mais demorada, pode-se utilizar um equipamento dedicado para esta finalidade, tal como um Controlador Lógico Programável (CLP) ou o controlador seqüencial baseado no Mark Flow Graph [34].

### 5.1.2 Controle de Execução de Tarefas

Este nível de controle é responsável pela execução de uma determinada tarefa indicada pelo controle supervisor. Normalmente esta tarefa é composta de diversos tipos de movimentos tais como a trajetória ponto-a-ponto ou outras trajetórias que a garra deve rastrear, e alguns movimentos são definidos por sensores externos, tais como o sensor de visão que determina a posição e a orientação do objeto a ser agarrado.

Em termos concretos, o controle de execução recebe do controle supervisor a indicação da tarefa a ser executada e envia seqüencialmente ao controlador de movimentos as coordenadas do ponto inicial e final, a trajetória e a velocidade de cada movimento.

### 5.1.3 Controle de Movimentos

Este nível de controle é responsável pela execução de movimentos que foram indicados pelo controle de execução de tarefas. Em outras palavras, este nível de controle executa periodicamente cálculos necessários de *set point* de velocidade ou de posição para cada uma das juntas para que a garra rastreie a trajetória indicada. Estes cálculos são baseados no algoritmo de controle de trajetória que necessita de informações sobre a posição das juntas para verificar se a garra está rastreando corretamente a trajetória e no caso de ocorrer o desvio da trajetória, os *set points* devem ser calculados para corrigir este desvio.

Estes *set points* calculados são enviados ao controle de juntas.

### 5.1.4 Controle de Juntas

Este nível de controle é responsável pelo controle de posição ou de velocidade de cada uma das juntas. Cada uma das juntas tem o seu *set point* de posição ou de velocidade enviado pelo controle de movimento e este nível de controle realiza o controle individual das juntas. Portanto é o nível de controle que, para robôs de alta velocidade e precisão, muitas vezes são implementados em multiprocessamento utilizando uma CPU para cada junta.

No presente trabalho foi desenvolvido um sistema de controle com 3 níveis hierárquicos que são Controle de Execução de Tarefas, Controle de Movimentos e Controle de Juntas. Foi desenvolvido um sistema de controle seqüencial baseado no *Mark Flow Graph* mas a integração com o sistema de controle do robô é objetivo de estudo futuro, pois envolve a reestruturação da linguagem de programação e conseqüentemente a alteração do interpretador de linguagem.

## 5.2 HARDWARE

O *hardware* do sistema de controle está baseado em

microcomputador compatível ao PC-IBM com o microprocessador 80386 DX com o *clock* de 40 MHz e coprocessador 80387, memória RAM de 4 Mbytes, memória cache de 128 Kbytes, disco rígido de 117 Mbytes, acionador de disco flexível de 3,5 polegadas de 1,44 Mbytes e unidade de vídeo de padrão SVGA monocromático, com cartões de interfaceamento desenvolvidos no presente trabalho com 6 canais de conversor D/A de 10 bits, 6 canais de contador *up/down* de 16 bits, 2 canais de comunicação paralela com outros microcomputadores e uma entrada digital de 16 bits.

As razões da escolha do microcomputador compatível ao PC-IBM são devido principalmente:

1. a esta máquina apresentar capacidade computacional satisfatória para processar o algoritmo de controle em tempo real;
2. a possibilidade de se desenvolver placa de interfaceamento com sensores internos e circuitos de acionamento do robô, pois a arquitetura é amplamente divulgada;
3. a possibilidade de se desenvolver software de controle em linguagem de alto nível, dispensando, desta maneira, um sistema de desenvolvimento dispendioso;
4. a facilidade em se migrar do protótipo para o produto acabado, pois atualmente diversas empresas fabricam esta máquina na versão industrial;
5. a esta máquina estar evoluindo em sua capacidade computacional com a utilização de CPU mais veloz, tais como 80486 DX-2 66 MHz ou 80486 DX-4 100 Mhz, que possibilita o uso da mesma placa de interfaceamento colocada no seu *slot* de expansão e, portanto, poder, futuramente, implementar algoritmos de controle mais complexos.

A figura [5.2] ilustra o diagrama em blocos deste controlador mostrando inclusive como está interligado com o sistema mecânico.

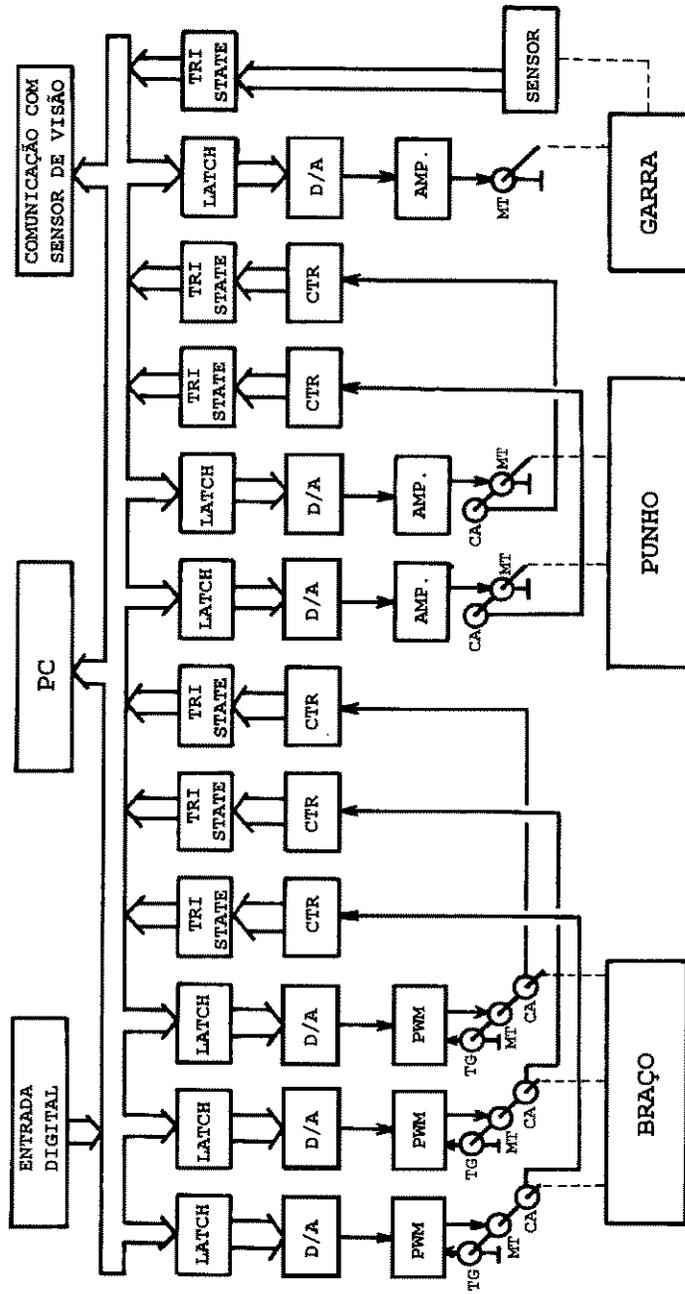


FIG. 5.2 HARDWARE DO CONTROLADOR DO ROBÔ

A descrição técnica dos circuitos de interfaceamento se encontra no apêndice B.

### 5.3 SOFTWARE.

Um dos objetivos do desenvolvimento deste projeto é que ele sirva como planta física para convalidação prática de novas estratégias de controle de robôs, visto que no Brasil existem vários estudos teóricos que não foram convalidados na prática.

A convalidação prática dos algoritmos de controle desenvolvidos teoricamente não é uma tarefa simples, mesmo que hajam recursos financeiros disponíveis para aquisição de robôs comerciais, pois estes são máquinas fechadas onde os fabricantes, em nome dos segredos industriais, não fornecem informações técnicas detalhadas. Assim, ao tentar utilizar um robô comercial para esta finalidade, torna-se necessário um investimento adicional tanto em recursos humanos com conhecimento especializados, como em recursos financeiros (na preparação deste robô comercial para que possa ser experimentado os novos algoritmos de controle). Neste processo de preparação, muitas vezes torna-se necessário abandonar o *hardware* e *software* de controle original do robô comercial e refazer o sistema inteiro, aproveitando-se somente a sua parte mecânica.

Assim, ao se conceber o *software* para o presente robô, foi dada uma grande importância neste aspecto e foi implementado *software* específico para esta finalidade de implantação e verificação das novas estratégias de controle. Este *software* foi denominado de módulo de análise.

Outro objetivo deste projeto é que ele seja um protótipo de um robô para aplicações industriais. Em outras palavras, este robô deve ter a facilidade de programação das tarefas, de verificação dos programas das tarefas e de execução automática das tarefas num ambiente amigável, que se comunica facilmente com o usuário mesmo que não seja um especialista em robótica. Estes recursos foram denominados, respectivamente,

de módulo de programação, módulo de execução passo a passo e módulo de execução automática.

Esses diversos módulos entram em ação através de comando dado pelo operador via interface homem máquina (IHM) bastante simples, que foi denominada de seletor de módulos.

No decorrer do desenvolvimento do *software*, foi detectada a necessidade de um módulo de frenagem com a função de manter as juntas do robô numa determinada posição quando livre da ação do controle para que desta maneira sejam vencidas as perturbações causadas pelo *drift* dos acionadores dos motores. Este *drift*, segundo o manual do fabricante, é característica desse *driver* e deve ser ajustado pelo usuário. Este módulo foi denominado de *servolock*.

A interligação desses módulos está representada na ilustração da figura [5.3].

Nas seções seguintes são descritos em detalhes as funções de cada módulo.

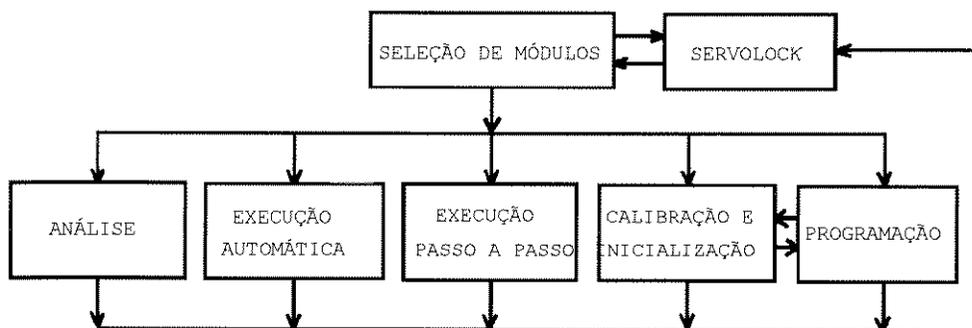


FIG. 5.3 MÓDULOS DE SOFTWARE DO CONTROLADOR

### 5.3.1 Módulo de Análise.

Para verificar a estabilidade, velocidade e suavidade de movimentos do robô, a percepção humana é muitas vezes insuficiente, pois trata-se de cinco graus de liberdade em movimentos simultâneos.

Por outro lado, a movimentação individual de cada uma das juntas é insuficiente para a avaliação global, na medida que uma junta em movimento interfere dinamicamente no movimento das outras juntas. Para contornar este problema, o presente módulo conta com um sistema de aquisição de dados e recursos gráficos que permitem a análise do comportamento do sistema em qualquer que seja a situação.

Como o sistema de controle é baseado em microcomputador, foi inserido um *software* de armazenamento de dados de movimentos dentro do *software* de controle de movimento, para que se possa analisar posteriormente os movimentos realizados. Os dados que são adquiridos e armazenados em forma de arquivo no disco são a posição, em milímetro na junta da base deslizante e em graus nas outras juntas, e as referências de velocidades, em mm/s na junta da base deslizante e graus por segundo nas outras juntas, em cada instante de amostragem de controle.

A quantidade de dados que pode ser armazenada é limitada a 2000 conjuntos, ou seja, 5 juntas X 2 variáveis X 2000, que são suficientes para a movimentação da ponta do braço de uma posição a outra. Estes 2000 conjuntos de valores são suficientes porque o período de amostragem é de 5 ms e portanto 2000 amostragens correspondem a 10 segundos, que é tempo suficiente para que todas as juntas alcancem seus *set points*.

Foi desenvolvido também um *software* para traçar gráficos desses dados armazenados em função do tempo, através do qual é realizada a avaliação dos movimentos.

Este recurso gráfico desenvolvido é dotado de recurso de *ZOOM* para ampliar as partes desejadas e possibilitar a

visualização de pequenas variações de movimentos. Assim, ao visualizar o comportamento global de movimento do robô na tela do computador, pode-se selecionar a faixa de valores que seja de interesse para avaliação, tanto no eixo horizontal como no eixo vertical, e realizar ampliação para que se possa visualizar melhor o comportamento do robô naquela faixa.

As figuras [5.4] e [5.5] ilustram o comportamento da base linear, sendo que na figura [5.4] tem a visão geral e na figura [5.5] a ampliação de uma faixa demarcada por linhas tracejadas na figura [5.4].

### 5.3.2 Módulo de Calibração e Inicialização.

Um dos recursos indispensáveis num robô industrial é a movimentação individual das suas juntas para conduzir manualmente e continuamente o robô através de comando manual.

Este recurso é utilizado para posicionar manualmente a ponta do braço na posição desejada com diversas finalidades, tais como, a verificação da melhor posição para o agarramento do objeto pelo robô ou para programação das coordenadas dos pontos de trabalho ou de passagem.

Foram atribuídas 3 teclas para movimentação de cada junta sendo uma para incrementar a posição da junta, outra para decrementar e a terceira tecla para parar o movimento.

Quando as juntas são movimentadas manualmente através destas teclas, são mostrados o valor atual da posição das juntas na tela.

Foi designada uma tecla especialmente para as juntas retornarem à posição inicial que corresponde ao valor zero de sensores de posição. Esta posição é denominada de *HOME POSITION*.

### 5.3.3 Módulo de Programação.

Programar um robô para uma determinada tarefa significa armazenar na memória do computador de controle um conjunto de informações que represente os movimentos do elemento final de

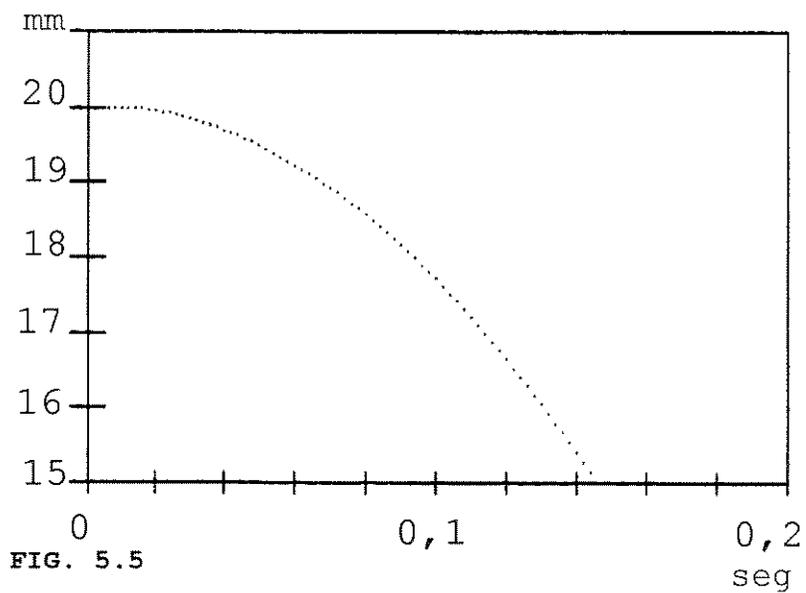
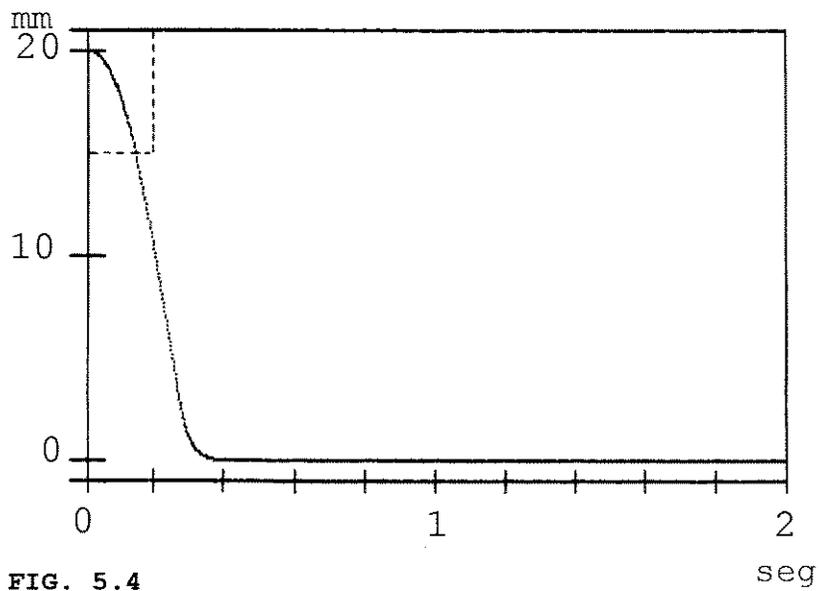


FIG. 5.4 VISÃO GERAL DO DESLOCAMENTO DA BASE LINEAR  
FIG. 5.5 AMPLIAÇÃO DA PARTE DEMARCADA DA FIGURA 5.4

ação para execução da tarefa. Estas informações são basicamente: as posições e orientações do elemento final de ação, que aqui denominamos de ponto de trabalho e ponto de passagem; a seqüência para alcançar esses pontos de trabalho e de passagens; tipo de trajetória do elemento final de ação para navegar de um ponto para outro; a velocidade de navegação; e a forma de atuação do elemento final de ação (por exemplo, no caso da garra pode ser, abre garra ou fecha garra).

A definição de cada ponto de trabalho e de passagem pode ser feita em sistema de coordenadas das juntas ou em sistema de coordenadas cartesianas universal ou da base do robô.

Por exemplo, as coordenadas dos pontos de trabalho A, B e C ilustrados na figura [5.6] podem ser definidas de duas maneiras. Uma delas é definir em coordenadas cartesianas do centro do punho, ângulo de mergulho e ângulo de rotação da garra, que seria:

$$\text{Ponto A} = X_1, Y_1, Z_1, \theta_{\text{MERG1}}, \theta_{\text{ROTGR1}}$$

$$\text{Ponto B} = X_2, Y_2, Z_2, \theta_{\text{MERG2}}, \theta_{\text{ROTGR2}}$$

$$\text{Ponto C} = X_3, Y_3, Z_3, \theta_{\text{MERG3}}, \theta_{\text{ROTGR3}}$$

A outra forma é defini-los através das coordenadas das juntas, que seria:

$$\text{Ponto A} = X_1, \theta_{\text{OMBRO1}}, \theta_{\text{COTOVELO1}}, \theta_{\text{MERG1}}, \theta_{\text{ROTGR1}}$$

$$\text{Ponto B} = X_2, \theta_{\text{OMBRO2}}, \theta_{\text{COTOVELO2}}, \theta_{\text{MERG2}}, \theta_{\text{ROTGR2}}$$

$$\text{Ponto C} = X_3, \theta_{\text{OMBRO3}}, \theta_{\text{COTOVELO3}}, \theta_{\text{MERG3}}, \theta_{\text{ROTGR3}}$$



espaço de trabalho atingível pelo robô, e em segundo lugar, pelo método denominado de *ON LINE*, onde as coordenadas dos pontos de trabalho e de passagem são definidas utilizando-se o próprio robô pela técnica de aprendizagem, ou seja, o programador conduz o elemento final de ação às posições e orientações dos diversos pontos desejados, de passagem e de trabalho e armazena no controlador essas posições em coordenadas de juntas ou coordenadas cartesianas do centro do punho e ângulo de mergulho e de rotação da garra.

Aparentemente o método *OFF LINE* é mais vantajoso que o método *ON LINE* pois pode-se realizar a programação dos pontos sem ocupar o robô. Entretanto, para isso é necessário que se faça uma análise prévia das possibilidades de execução de trajetórias livres de obstáculos, através de uma documentação detalhada do ambiente de trabalho, pois se esta documentação for insuficiente podem ocorrer situações de risco, tais como a colisão do robô em alguns obstáculos, especificação de pontos não atingíveis ou pontos singulares, etc.

Neste sentido, podemos afirmar que o método *OFF LINE* é mais analítico e o método *ON LINE* é mais intuitivo e direto, não requerendo análise prévia.

A nova tendência na programação do robô é o emprego de métodos que utilizam as vantagens desses dois métodos, *OFF LINE* e *ON LINE*, através de técnicas de computação gráfica. Nesses métodos, o ambiente de trabalho e o robô são mostrados na tela do computador através de uma simulação do seu modelo cinemático, com a animação vista na tela do computador.

O ambiente e o robô representados na tela podem ser observados por diversos ângulos de visão como se o programador estivesse no ambiente de trabalho do robô, com liberdade de observação dos ângulos de visão mais importantes para a especificação dos movimentos, simulando a situação real, inclusive com recursos de observação em *ZOOM* para poder observar maiores detalhes.

Após a conclusão da especificação de movimentos do robô, é feita a simulação para verificar se não há colisão entre o robô e algum obstáculo e também para se certificar que o robô executará corretamente a tarefa especificada.

No presente trabalho foi adotado um método de programação com a seguinte filosofia: É desejável que a programação seja fácil ao usuário de tal forma que este sintase como se estivesse descrevendo uma tarefa ao homem com baixa qualificação técnica, ou seja, ensinando a tarefa de forma direta e detalhada.

Baseado nesta filosofia, foi definida uma linguagem textual de programação composta de um conjunto mínimo de comandos com o qual seja possível descrever as tarefas de forma simples e direta. Nesta linguagem, os pontos de trabalho e de passagem são referenciados pelos seus nomes e as respectivas coordenadas são definidas através do método *ON LINE* ou *OFF LINE* descritos anteriormente.

A função desta linguagem de programação da tarefa é descrever a seqüência de rastreamento dos pontos, o tipo de trajetória, a velocidade do movimento e forma de atuação do elemento final de ação.

A sintaxe adotada nesta linguagem é simples e direta, para possibilitar que as pessoas sem o conhecimento de informática possam programar os movimentos, bastando ter conhecimento das trajetórias em coordenadas cartesianas a serem realizadas, para que o robô execute uma determinada tarefa. Em outras palavras, esta linguagem foi concebida objetivando, como usuário, os operadores das máquinas que conhecem as tarefas, mas não necessariamente programação de computadores.

Nesta linguagem os comandos são executados de cima para baixo, terminando a execução ao encontrar o comando END.

Foram adotados os seguintes comandos:

PTP	Trajeto�ria ponto-a-ponto
RET	Trajeto�ria retil�nea
CIR	Trajeto�ria circular
GARRA	Comando para garra
ABRE	Comando para abrir a garra
FECHA	Comando para fechar a garra
ESPERA	Comando para espera conforme o estado da entrada digital
SA�DA	Comando para emitir sinal digital
VIS	Utilizado no lugar de nome do ponto e as coordenadas do ponto destino ser�o fornecidas pelo sensor de vis�o
END	Final de programa

A sintaxe desta linguagem   seguinte:

*<trajeto ria>, <nome(s) do(s) ponto(s)>, <velocidade>*

onde: *<trajeto ria>*   a especifica o do tipo da trajeto ria que pode ser:

*PTP - para trajeto ria ponto-a-ponto*

*RET - para trajeto ria retil nea*

*CIR - para trajeto ria circular*

*<nome(s) do(s) ponto(s)>*   definido da seguinte maneira:

- a. Para trajeto ria ponto-a-ponto e para trajeto ria retil nea   citado apenas um nome do ponto destino;
- b. Para trajeto ria circular s o colocados os dois pontos sendo o primeiro o ponto intermedi rio pertencente ao c rculo e o segundo   o ponto de parada. Isso   devido ao m todo de defini o da trajeto ria circular que foi adotado no presente projeto, onde um c rculo   definido

através de três pontos pertencentes à trajetória, por ser mais prático.

<velocidade> tem significados diferentes dependendo do tipo da trajetória, como é explanado a seguir:

- a. Para trajetória ponto-a-ponto, o valor da velocidade deve estar entre 0 e 510, onde a relação entre este valor e a velocidade é a seguinte: para a base linear, o valor 510 corresponde a 633 mm/s e para juntas rotativas o mesmo valor corresponde a 143°/s;
- b. Para trajetória retilínea o valor da velocidade significa a velocidade na direção do movimento em mm/s tendo o valor máximo de 200 mm/s;
- c. Para trajetória circular o valor da velocidade significa a velocidade tangencial em mm/s tendo o valor máximo de 100 mm/s.

Esta definição de velocidade é importante para a execução da tarefa, principalmente nas trajetórias retilíneas e circulares quando o robô executa tarefas onde a velocidade de movimento é de importância fundamental, tais como: solda a arco, aplicação de adesivos, etc.

Quando o comando é para a garra, é citado o comando GARRA seguida de espaço e do comando ABRE ou FECHA.

O comando ESPERA é para aguardar que uma determinada entrada fique no estado indicado para continuar a execução do programa. A indicação da entrada é de E01 a E16. Por exemplo, na instrução

```
ESPERA E02=1
```

o robô fica parado no ponto anterior até que a entrada E02 tenha valor 1.

É utilizado VIS como o nome do ponto especial. Este nome é utilizado quando as coordenadas do ponto destino são fornecidas pelo sensor de visão. Quando o interpretador de

comando encontra este nome, o controlador do robô envia o comando de processamento para o sensor de visão e o braço espera parado no ponto anterior àquele comando, até que o sensor de visão forneça as coordenadas do ponto destino.

O comando SAIDA é para emitir um sinal digital do controlador para equipamentos periféricos. Por exemplo, o comando

*SAIDA S03=1*

é para emitir sinal digital 1 na saída S03. Este comando SAIDA, associado com o comando ESPERA serve para sincronizar o movimento do robô com os equipamentos periféricos.

As figuras [5.7] e [5.8] ilustram um exemplo. Neste exemplo, a garra, inicialmente na posição inicial, abre os dedos e se movimenta até o ponto PT1 numa trajetória ponto-a-ponto. Alcançando este ponto, o controlador envia comando de processamento para o sensor de visão e aguarda até a chegada da informação de coordenadas do ponto destino VIS, fornecida por sensor de visão, e então movimenta a garra até o ponto VIS numa trajetória ponto-a-ponto. Chegando naquele ponto, a garra fecha os dedos e se movimenta na direção do ponto PT2 numa trajetória retilínea com velocidade 100 mm/s. Chegando no ponto PT2, o robô espera até que a entrada E04 tenha o valor 1. Quando a entrada E04 tiver o valor 1, a garra é movimentada na direção do ponto PT3 numa trajetória retilínea e velocidade 80 mm/s e posteriormente alcança o ponto PT5 passando pelo ponto PT4, em trajetória circular com a velocidade tangencial 80 mm/s. Chegando no ponto PT5, a garra abre os dedos e se movimenta na direção PT1 em trajetória retilínea com velocidade de 80 mm/s.

O método de programação das coordenadas dos pontos de trabalho e de passagem adotado no presente projeto é basicamente o método *ON LINE*, e é realizado movendo-se o robô manualmente por meio do teclado, utilizando-se o módulo de

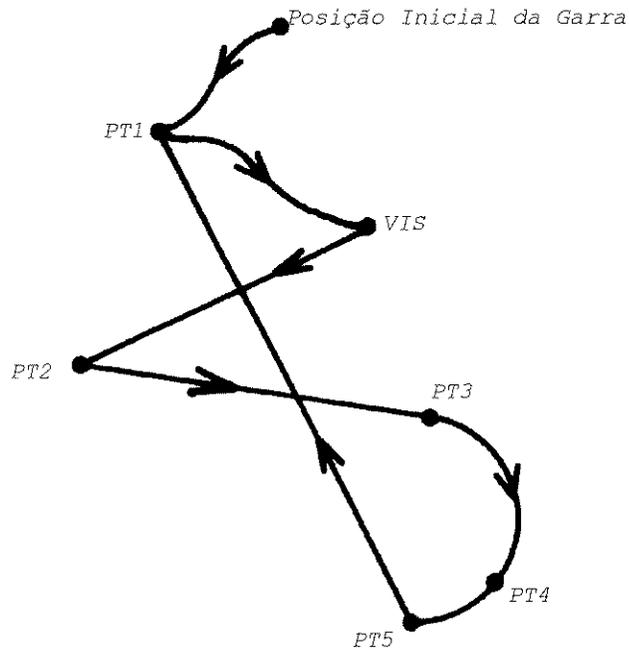


FIG. 5.7 NOME DOS PONTOS E TIPOS DE MOVIMENTAÇÃO DO ROBÔ

```

GARRA ABRE
PTP,PT1,160
PTP,VIS,160
GARRA FECHA
RET,PT2,100
ESPERA E04=1
RET,PT3,080
CIR,PT4,PT5,080
GARRA ABRE
RET,PT1,080
END

```

FIG. 5.8 PROGRAMA DE MOVIMENTO DO BRAÇO MOSTRADO NA FIG.5.7

calibração e inicialização, gravando-se os valores de  $X$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  e  $\theta_5$  daqueles pontos de trabalho e de passagem, no arquivo de coordenadas pós pontos, onde:

- $X$  é o deslocamento da base linear;
- $\theta_2$  deslocamento angular da junta do ombro;
- $\theta_3$  deslocamento angular da junta do cotovelo;
- $\theta_4$  e  $\theta_5$  são os deslocamentos angulares das juntas do punho para os movimentos de mergulho e rotação respectivamente;

$$\theta_{\text{MERGULHO}} = \theta_3 + \theta_4 \quad (5.1)$$

$$\theta_{\text{ROTGAR}} = \frac{\theta_5 - \theta_4}{2} \quad (5.2)$$

Em outras palavras, programar coordenadas dos pontos significa atribuir um nome e os valores de posição das suas juntas para um determinado ponto e criar um arquivo de dados de diversos pontos.

Os dois arquivos, arquivo de coordenadas dos pontos e arquivo de descrição de movimentos, são utilizados pelo submódulo de compilação que monta um arquivo de execução da tarefa. Este arquivo de execução é basicamente o mesmo do arquivo de movimentos, diferindo apenas que no lugar do nome dos pontos estão os valores numéricos das juntas  $X$ ,  $\theta_2$ ,  $\theta_3$ ,  $\theta_4$  e  $\theta_5$ , seguido do código da trajetória e o estado da garra.

Esquemáticamente, o módulo de programação pode ser representado conforme a figura [5.9].

#### 5.3.4 Módulo de Execução Passo a Passo.

Após a conclusão da programação é necessário que haja um recurso para verificar se o programa do robô está correto. Esta verificação pode ser feita executando-se o programa passo

a passo, ou seja, parando o robô cada vez que ele atinge o ponto programado para que possa ser verificado os movimentos de um ponto a outro; sempre que se constatar erro na programação será feita a correção das coordenadas dos pontos ou da seqüência de rastreamento.

O módulo de execução passo a passo foi implementado com esta finalidade.

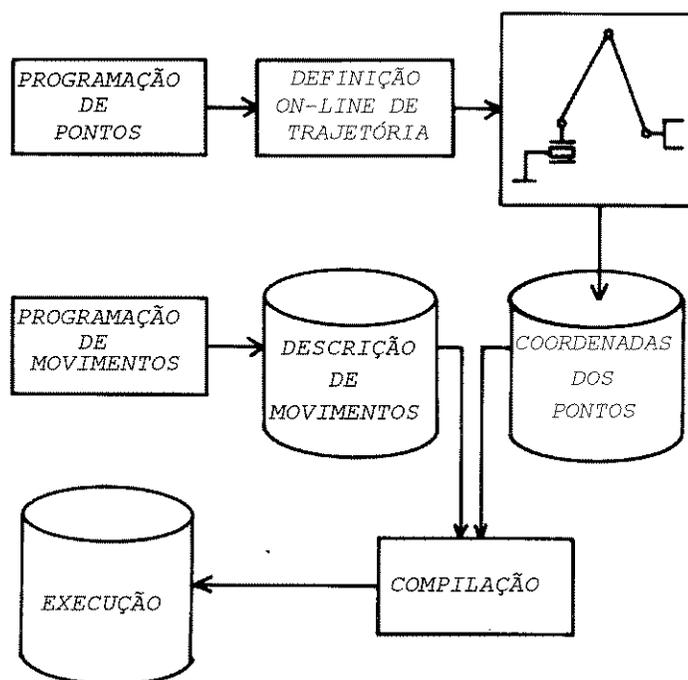


FIG. 5.16 MÓDULO DE PROGRAMAÇÃO

### 5.3.5 Módulo de Execução Automática.

O módulo de execução automática é o que executa automaticamente os movimentos do robô baseado no programa que consta no arquivo de programas executáveis.

A quantidade de repetição do movimento é indicada

inicialmente, via teclado, mas pode ser interrompida a qualquer momento pela digitação de uma tecla do controlador.

Após a parada do robô pela conclusão do programa ou pela interrupção pelo operador, o controlador entra em execução no módulo SELEÇÃO DE MÓDULOS, para assegurar a manutenção da posição do robô e a seleção de outros módulos de operação.

### 5.3.6 Algoritmo de Ajuste e Controle de Trajetória.

Foram desenvolvidos três tipos de algoritmo de controle de movimentos:

- a. *Servo Lock*;
- b. Controle de Trajetória Ponto-a-Ponto;
- c. Controle de Trajetória Contínua.

Nas seções a seguir serão descritos cada um desses algoritmos.

#### a) Controle *Servo Lock*

Este algoritmo de controle não é exatamente para movimentar o robô, mas para mantê-lo o robô numa determinada posição, vencendo as perturbações causadas principalmente pelo *drift* do circuito de acionamento dos motores a PWM que apresentam a variação do *OFF SET* devido ao *drift*.

Como este algoritmo tem a finalidade de atuar no sistema para um erro pequeno de posição, a estratégia de controle é proporcional com ganho alto e com saturação de velocidade. Este algoritmo de *Servo Lock* é utilizado também para movimentar as juntas do robô através do comando manual, para assegurar a parada suave nesse modo de movimento.

#### b) Controle de Trajetória Ponto-a-Ponto.

O controle de movimento ponto-a-ponto do robô é caracterizado pelo controle individual da posição das juntas, onde cada uma das juntas alcança o seu *set point* individualmente. Portanto, neste modo de movimento, a trajetória da ponta do robô não é definida.

A estratégia de controle adotada para este modo de operação é, basicamente, proporcional, mas com um tratamento diferente na partida, para evitar que o robô dê uma partida brusca. Esse tipo de cuidado é necessário considerando que a estrutura do robô articulado é constituída de barras rígidas apoiadas apenas por uma das extremidades e em cascata, e, portanto, a aceleração brusca é prejudicial ao mecanismo.

Este tratamento na partida é apenas para causar uma aceleração inicial controlada até que a velocidade das juntas alcancem um certo valor pre-estabelecido. Após alcançar esta velocidade, permanece com este valor até que o erro de posição atinja um certo valor. A partir desse valor de erro, a estratégia de controle é comutada por *software* para o controle proporcional convencional.

Esta estratégia de controle foi desenvolvida para se evitar o impacto inicial, alcançar a posição desejada com menor tempo, evitar o *overshoot* e ter uma parada suave.

A figura [5.10] ilustra a curva de referência de velocidade gerada por esta estratégia.

As figuras [5.11] a [5.15] ilustram as curvas de referência de velocidade e a variação de posição das juntas obtidas experimentalmente, segundo a estratégia de controle descrita anteriormente.

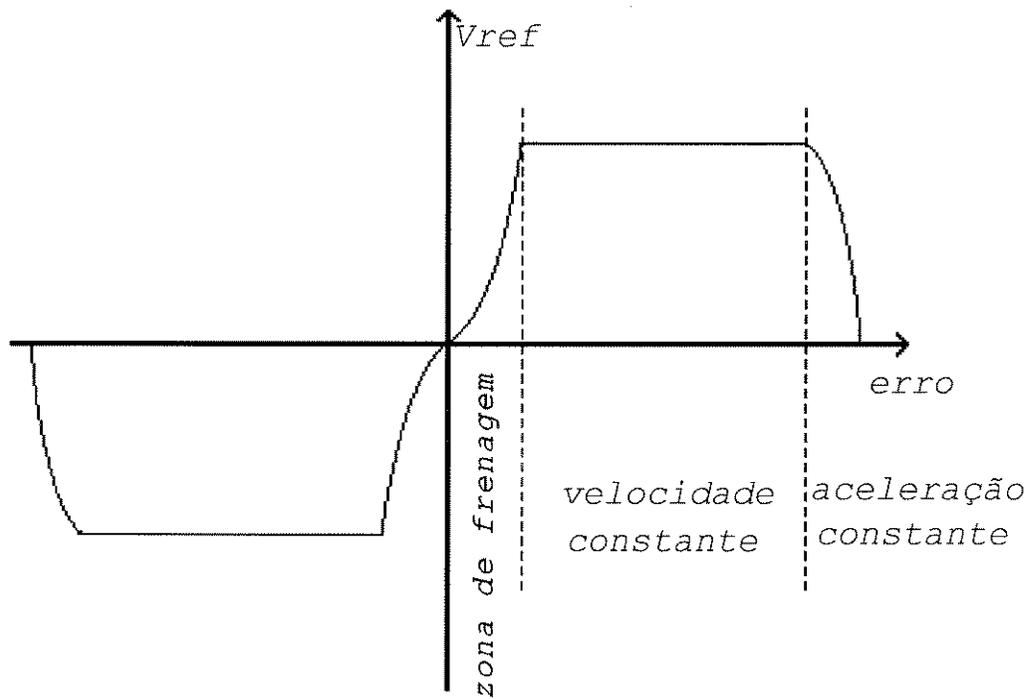


FIG. 5.10 CURVA DE VELOCIDADE DE UMA JUNTA

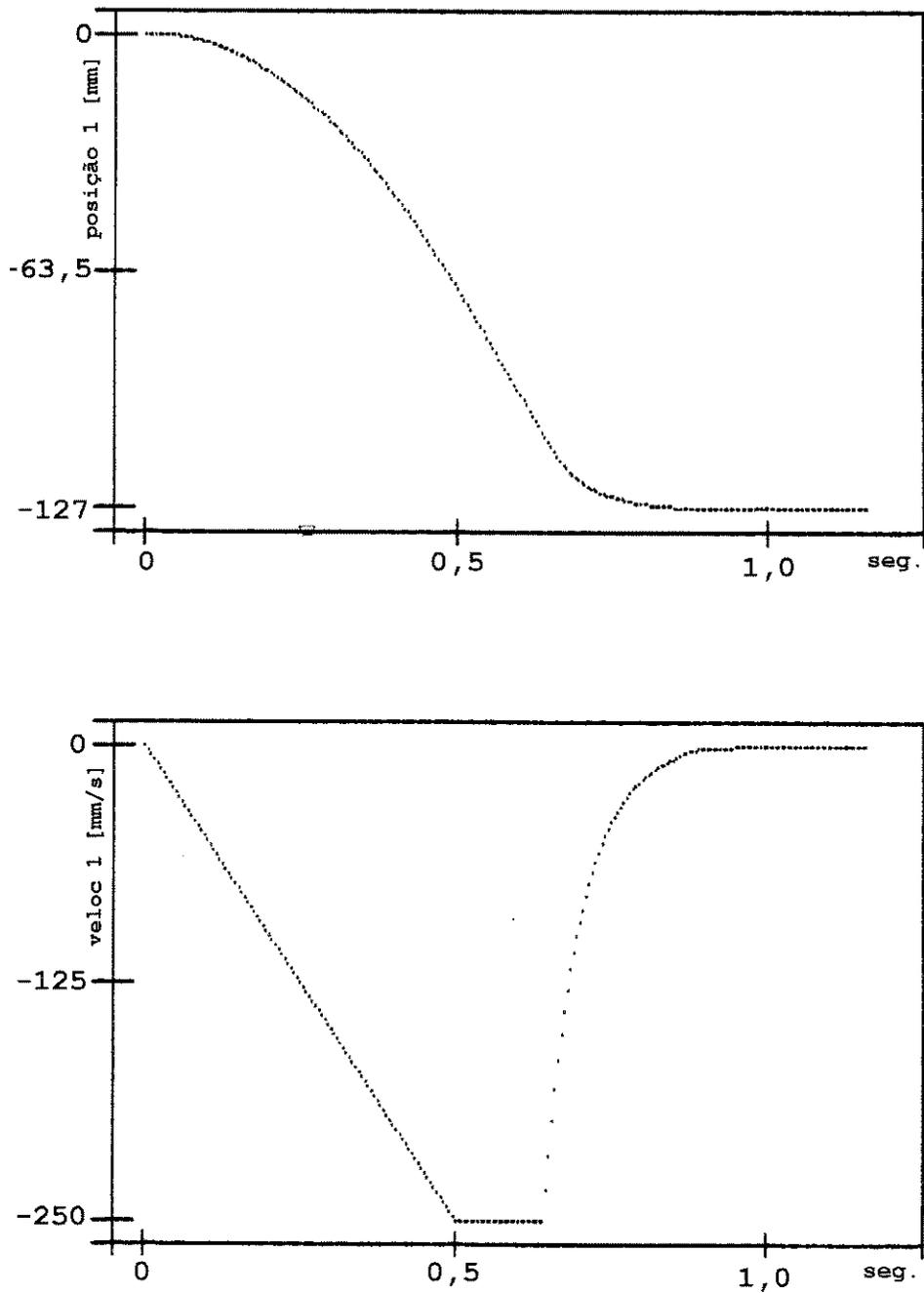


FIG.5.11 POSIÇÃO E VELOCIDADE DE REFERÊNCIA DA BASE DESLIZANTE

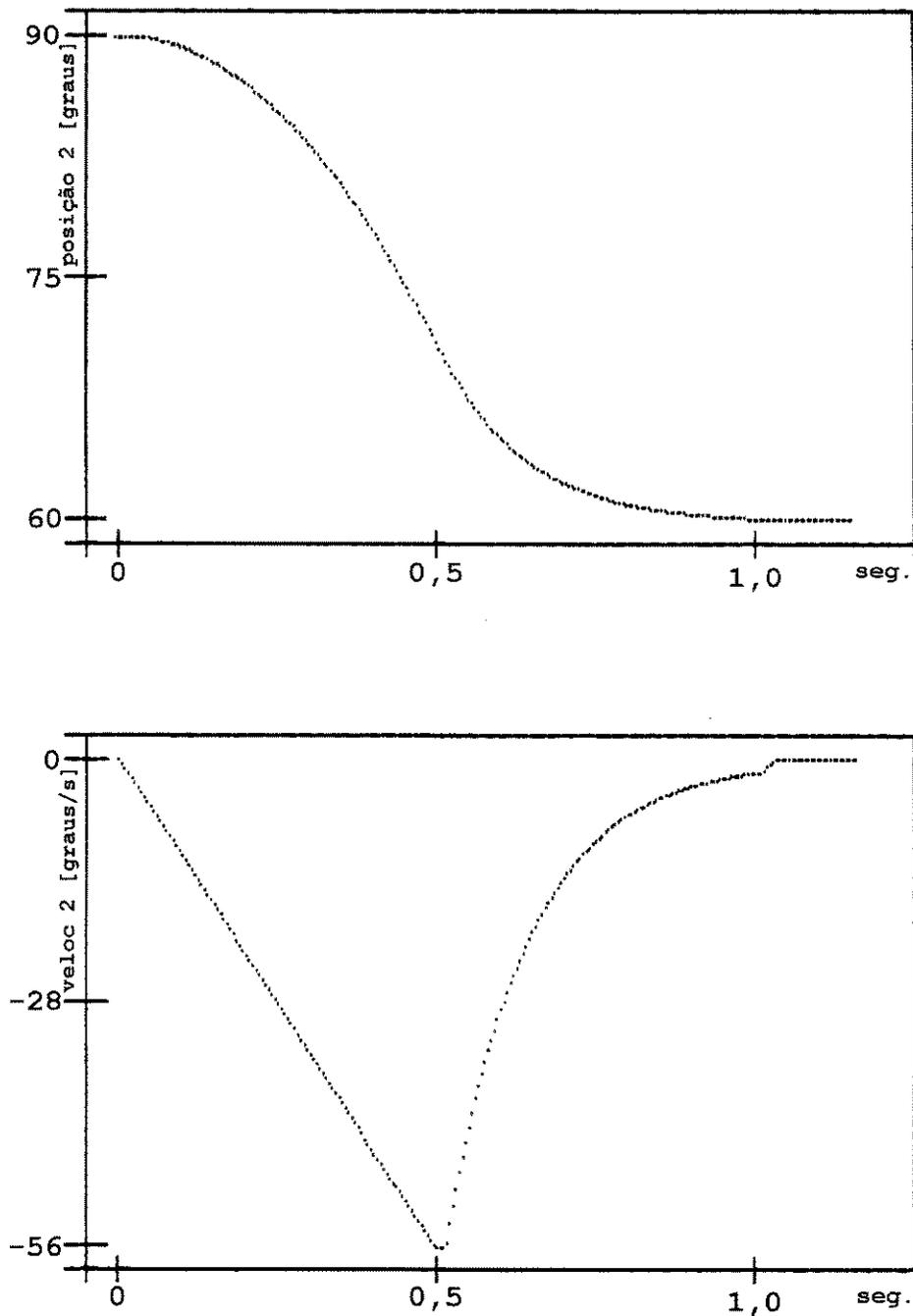


FIG. 5.12 POSIÇÃO E VELOCIDADE DE REFERÊNCIA PARA A JUNTA  $\theta_2$

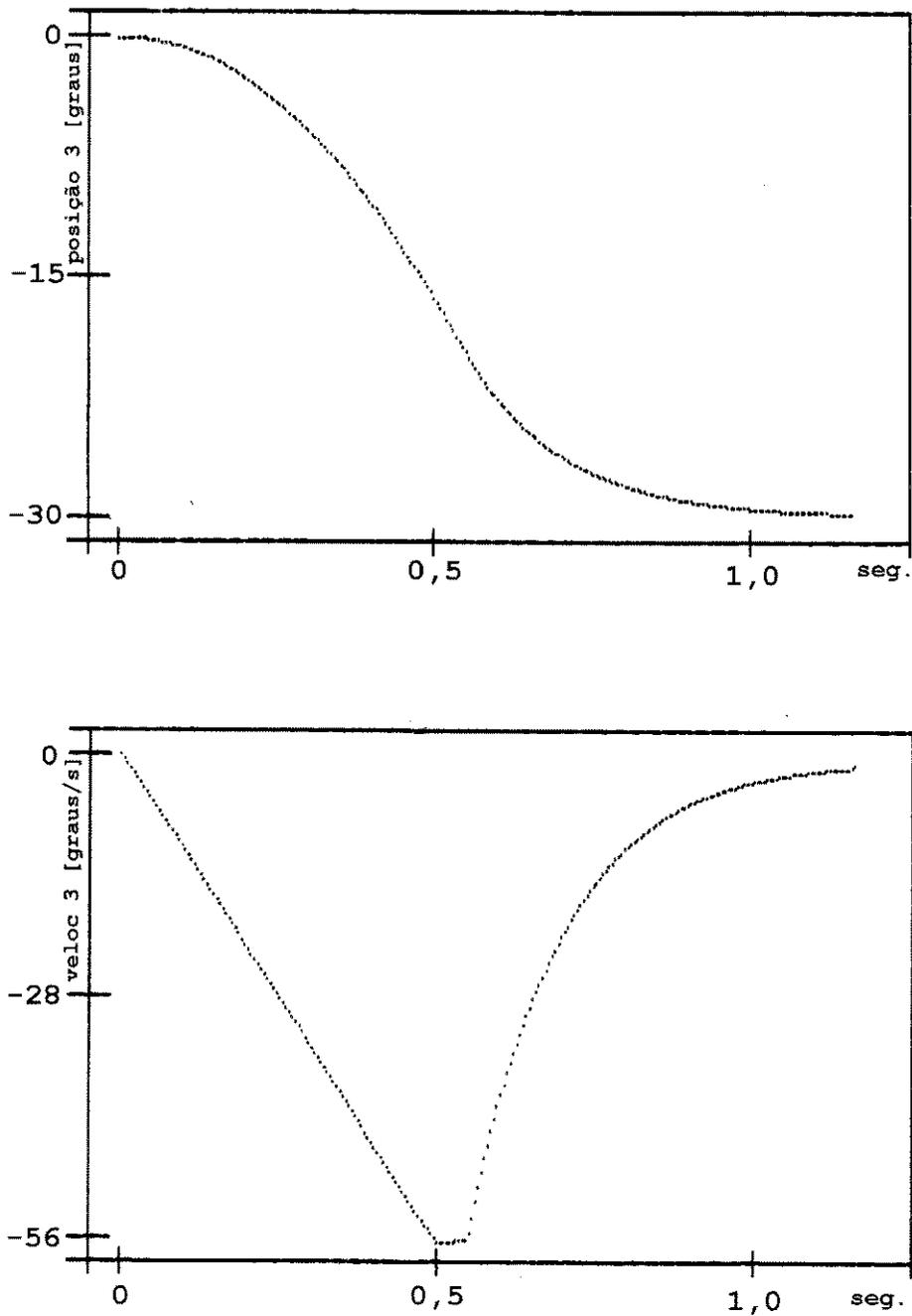


FIG. 5.13 POSIÇÃO E VELOCIDADE DE REFERÊNCIA PARA A JUNTA  $\theta_3$

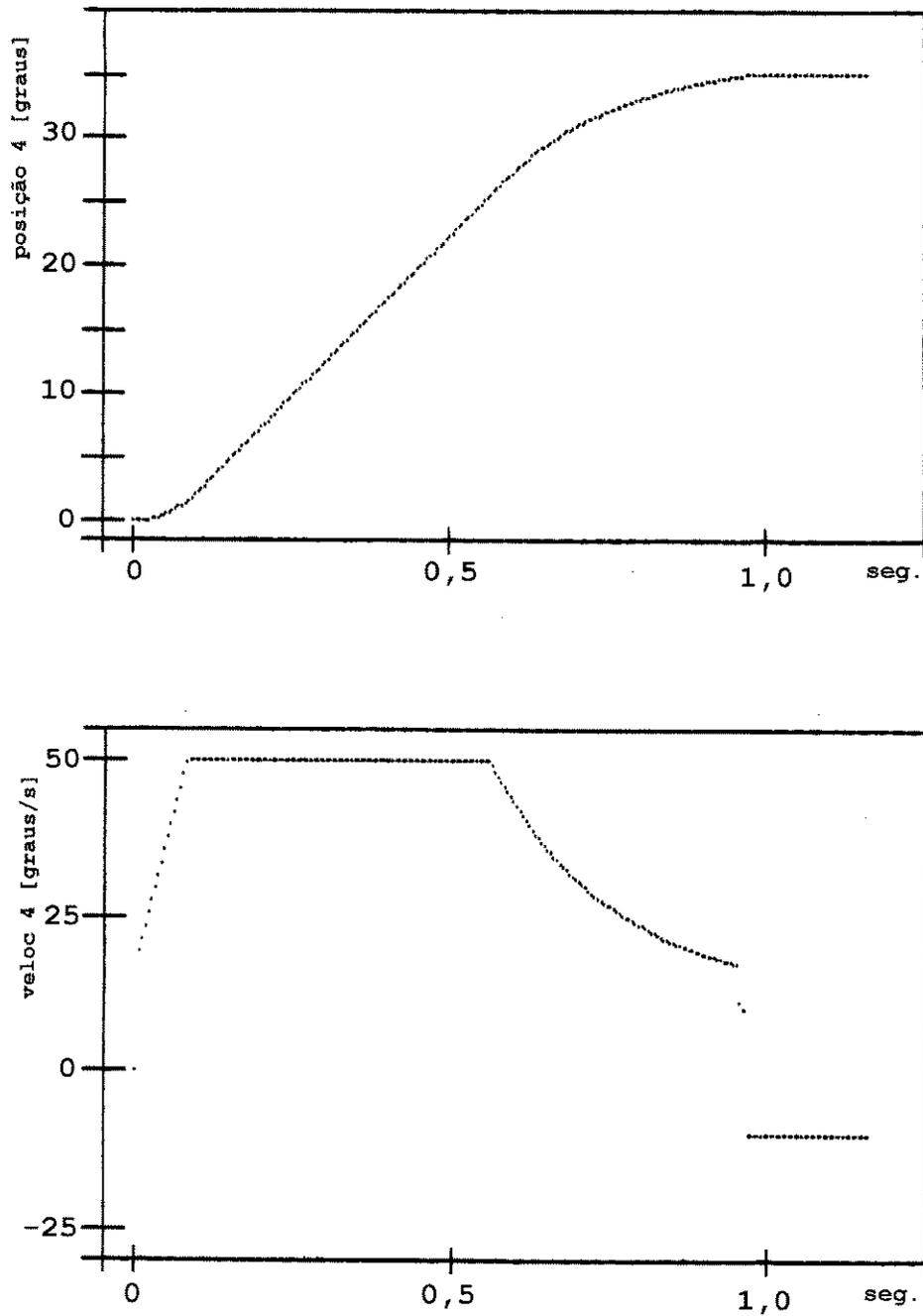


FIG. 5.14 POSIÇÃO E VELOCIDADE DE REFERÊNCIA PARA A JUNTA  $\theta_4$

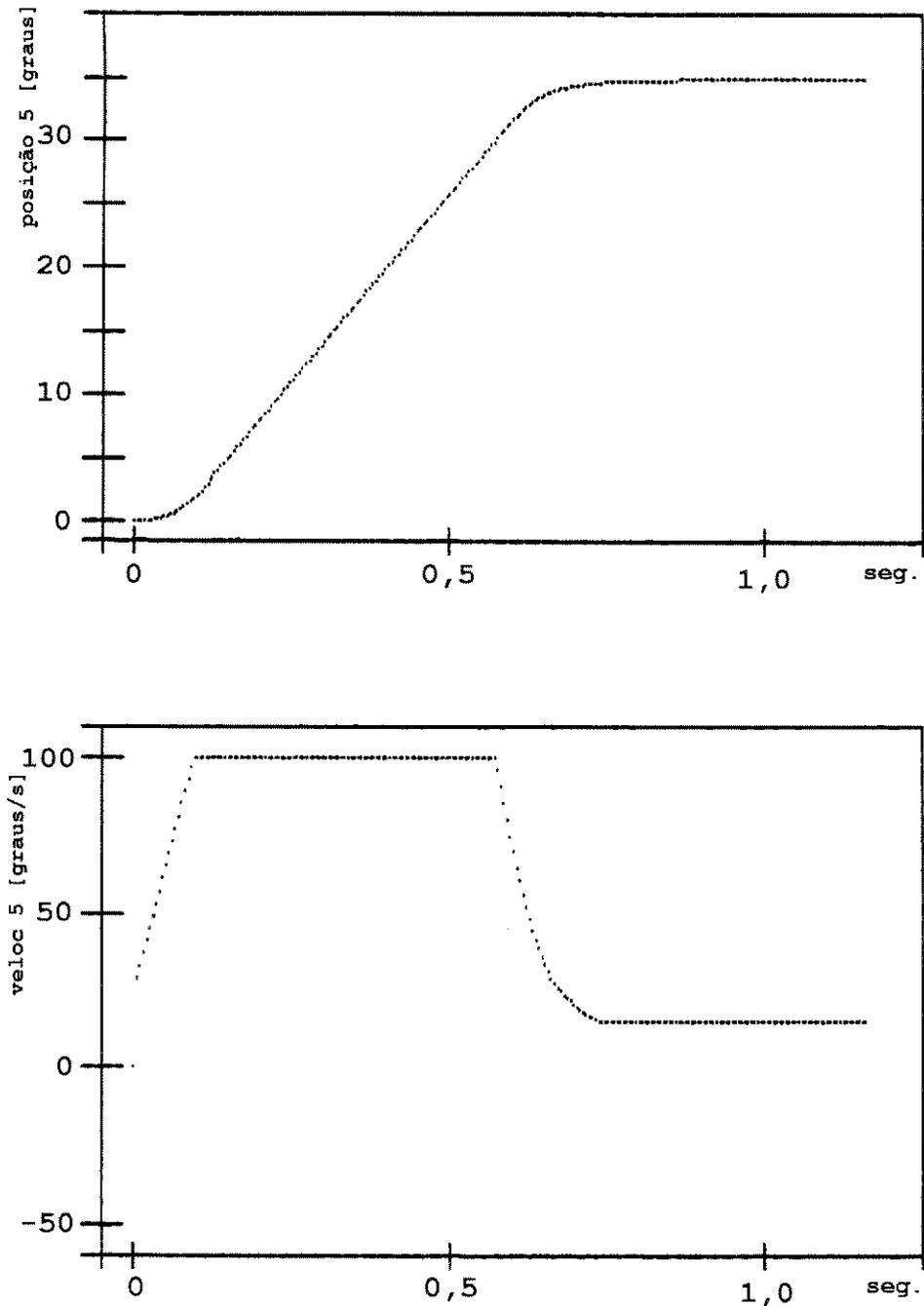


FIG. 5.15 POSIÇÃO E VELOCIDADE DE REFERÊNCIA PARA A JUNTA  $\theta_5$

### c) Controle de Trajetória Contínua.

- i. Trajetória Retilínea;
- ii. Trajetória Circular.

#### i) Trajetória Retilínea.

Em algumas aplicações de um robô industrial, torna-se necessário realizar o controle de trajetória da ponta do braço, tais como em solda a arco, aplicação de adesivos, pintura, polimento, etc., e mesmo em algumas atividades simples de agarramento de um objeto, é desejável que se tenha o controle da trajetória da ponta do braço.

Assim, foi realizado um estudo visando a implementação do controle de trajetória para realizar o agarramento de objeto visto por câmera de TV.

O controle de trajetória num robô articulado é um problema relativamente complexo. Assim se torna necessário desenvolver o raciocínio por etapas bem definidas.

Inicialmente é definido um novo sistema de coordenadas cartesianas  $(X_s, Y_s, Z_s)$  no espaço de trabalho, onde o eixo  $X_s$  é a própria trajetória retilínea a ser rastreada pela ponta do braço, como é ilustrado na figura [5.16]. Como pode ser visto nesta figura, os eixos  $X_s$  e  $Z_s$  são definidos girando os eixos XYZ em torno do eixo Y num ângulo de  $90^\circ$ , e deslocando a origem para a coordenada  $(X_0, Y_0, Z_0)$ .

Sabendo-se que a matriz  $L$  (3X3) de transformação rotativa é definida como:

$$L = \begin{bmatrix} \cos \phi \cos \theta & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \sin \phi \cos \theta & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi \\ -\sin \theta & \cos \theta \sin \psi & \cos \theta \cos \psi \end{bmatrix}$$

(5.3)

onde:

$\phi$  - ângulo de rotação em torno do eixo X;

$\theta$  - ângulo de rotação em torno do eixo Y;

$\psi$  - ângulo de rotação em torno do eixo Z.

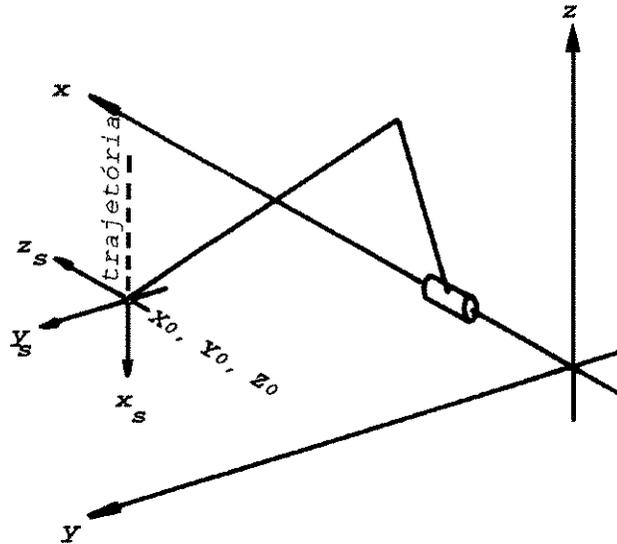


FIG. 5.16 COORDENADAS  $(X_{s'}, Y_{s'}, Z_{s'})$

Logo, para o presente caso, tem-se:

$$L = [\vec{n} \quad \vec{s} \quad \vec{a}] \quad (5.4)$$

onde:

$$\vec{n} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \vec{s} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \vec{a} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad (5.5)$$

Para transformar as coordenadas  $(X_{ATUAL}, Y_{ATUAL}, Z_{ATUAL})$  da posição atual da ponta do braço em novo sistema de coordenadas  $(X_s, Y_s, Z_s)$ , são feitas as seguintes operações de transformações homogêneas:

$$\begin{bmatrix} X_s \\ Y_s \\ Z_s \\ 1 \end{bmatrix} = \begin{bmatrix} & & \cdot & -pn \\ & L & \cdot & -po \\ & & \cdot & -pa \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{ATUAL} \\ Y_{ATUAL} \\ Z_{ATUAL} \\ 1 \end{bmatrix} \quad (5.6)$$

onde:

$$pn = \vec{n}^T \cdot \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \Rightarrow pn = -Z_0 \quad (5.7)$$

$$ps = \vec{s}^T \cdot \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \Rightarrow ps = Y_0 \quad (5.8)$$

$$pa = \vec{a}^T \cdot \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \end{bmatrix} \Rightarrow pa = X_0 \quad (5.9)$$

portanto:

$$\begin{aligned}
 X_{SATUAL} &= -Z_{ATUAL} + Z_0 \\
 Y_{SATUAL} &= Y_{ATUAL} - Y_0 \\
 Z_{SATUAL} &= X_{ATUAL} - X_0
 \end{aligned}
 \tag{5.10}$$

Pode-se notar que, como a trajetória retilínea a ser rastreada é coincidente com o eixo  $X_s$ , se a ponta do braço estiver sobre a trajetória, o valor do  $Y_{SATUAL} = 0$  e  $Z_{SATUAL} = 0$ . Em outras palavras, o próprio valor de  $Y_{SATUAL}$  e de  $Z_{SATUAL}$  indicam o erro de rastreamento de trajetória. Então, o algoritmo para manter a ponta do braço sobre a trajetória será:

$$\begin{bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{bmatrix} = \begin{bmatrix} V \\ -K_{REC} Y_{SATUAL} \\ -K_{REC} Z_{SATUAL} \end{bmatrix}
 \tag{5.11}$$

onde:

$V$  é o módulo da velocidade na direção da trajetória;

$K_{REC}$  é o ganho de recuperação da trajetória.

Após a determinação do  $\dot{X}_s$ ,  $\dot{Y}_s$  e  $\dot{Z}_s$ , esses valores deverão ser convertidos pelo modelo cinemático inverso através do jacobiano inverso e matriz de transformação homogênea, das coordenadas de trajetória para as coordenadas da base, em valores de referência de velocidade dos motores que acionam o eixo  $X$ ,  $\theta_{OMBRO}$  e  $\theta_{COTOVELO}$ . Esta conversão é feita da seguinte maneira:

1. Inicialmente as velocidades  $\dot{X}_s$ ,  $\dot{Y}_s$  e  $\dot{Z}_s$ , são transformadas em velocidades  $\dot{X}$ ,  $\dot{Y}$  e  $\dot{Z}$  através da equação de transformação homogênea de coordenadas de trajetória  $(X_s, Y_s, Z_s)$  para coordenadas de base  $(X, Y, Z)$ .

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = \begin{bmatrix} L \\ \cdot \\ \cdot \end{bmatrix} \cdot \begin{bmatrix} \dot{X}_S \\ \dot{Y}_S \\ \dot{Z}_S \end{bmatrix} \quad (5.12)$$

2. Calculado  $\dot{x}$ ,  $\dot{y}$  e  $\dot{z}$ , são determinados  $\dot{\theta}_{\text{OMBRO}}$  e  $\dot{\theta}_{\text{COTOVELO}}$  utilizando-se da inversa da matriz jacobiano referente às três juntas de posicionamento, que é quadrada e portanto admite inversa em todos os pontos do volume de trabalho que não sejam singulares. A singularidade da matriz jacobiano acontece quando  $\theta_{\text{OMBRO}} = \theta_{\text{COTOVELO}}$ , ou seja, quando o braço se encontra totalmente esticado que é uma situação que não acontece no presente robô devido a sua estrutura mecânica.

$$J^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{L_3 \cdot \cos \theta_{\text{COTOVELO}}}{\Delta} & \frac{L_3 \cdot \text{sen} \theta_{\text{COTOVELO}}}{\Delta} \\ 0 & -\frac{L_2 \cdot \cos \theta_{\text{OMBRO}}}{\Delta} & -\frac{L_2 \cdot \text{sen} \theta_{\text{OMBRO}}}{\Delta} \end{bmatrix} \quad (5.13)$$

onde  $\Delta = L_2 \cdot L_3 \cdot \text{sen}(\theta_{\text{COTOVELO}} - \theta_{\text{OMBRO}})$

Portanto, os valores de  $\dot{x}$ ,  $\dot{\theta}_{\text{OMBRO}}$  e  $\dot{\theta}_{\text{cotovelo}}$  são dados pelo vetor:

$$\begin{bmatrix} \dot{X} \\ \dot{\theta}_{OMBRO} \\ \dot{\theta}_{COTOVELO} \end{bmatrix} = J^{-1} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (5.14)$$

Resultando em

$$\dot{\theta}_{OMBRO} = \frac{L_3 \cdot \cos(\theta_{COTOVELO}) \cdot \dot{Y} + \text{sen}(\theta_{COTOVELO}) \cdot \dot{Z}}{\Delta} \quad (5.15)$$

$$\dot{\theta}_{COTOVELO} = \frac{-L_2 \cdot \cos(\theta_{OMBRO}) \cdot \dot{Y} + \text{sen}(\theta_{OMBRO}) \cdot \dot{Z}}{\Delta} \quad (5.16)$$

onde:  $\Delta \neq 0$

Baseado nos valores de  $\dot{X}$ ,  $\dot{\theta}_{OMBRO}$  e  $\dot{\theta}_{COTOVELO}$ , e  $\alpha_{COTOVELO}$ , calcula-se o valor a ser colocado nos respectivos canais do conversor D/A utilizando-se as seguintes equações:

$$\text{valorDA}_{OMBRO} = \frac{33339,02459 \cdot \text{sen}\alpha_{OMBRO}}{\sqrt{36500 - 32557,6412 \cdot \text{cos}\alpha_{OMBRO}}} \cdot \dot{\theta}_{OMBRO} \quad (5.17)$$

$$\text{valorDA}_{COTOVELO} = \frac{33339,02459 \cdot \text{sen}\alpha_{COTOVELO}}{\sqrt{36500 - 32557,6412 \cdot \text{cos}\alpha_{COTOVELO}}} \cdot \dot{\theta}_{COTOVELO} \quad (5.18)$$

$$\text{valorDA}_x = \frac{\dot{X}}{1,240234375} \quad (5.19)$$

Obs: O cálculo do  $\alpha$  está descrito no capítulo 2.

ii) Trajetória Circular.

No presente trabalho não foi desenvolvido o algoritmo de controle de trajetória circular para este robô porém o mesmo se encontra em desenvolvimento baseado na estratégia *Autonomous Trajectory Generating Servomechanism*, ATGS [11] com estudos realizados posteriormente a [32] e [41], na pesquisa da Tese de Doutorado de Antonieta do Lago Vieira, no DSCE-FEE-UNICAMP sob a orientação do Prof. Dr. Alvaro Geraldo Badan Palhares.

O princípio de geração da trajetória da ponta do robô é descrito a seguir:

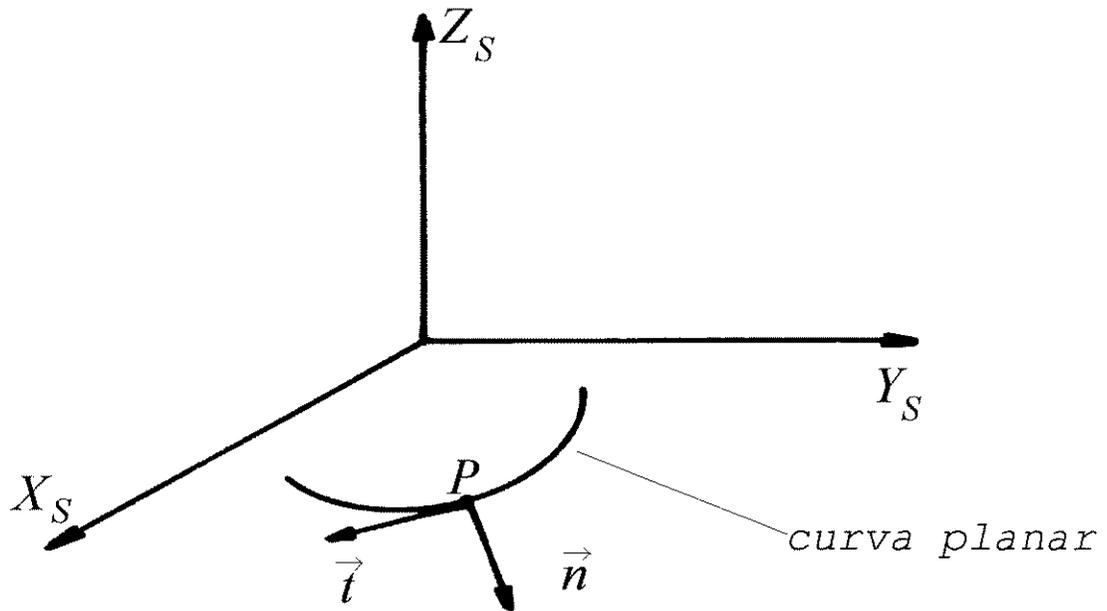
Suponha que a ponta do braço rastreie uma curva planar num plano arbitrário  $X_s Y_s$ , conforme a ilustração da figura [5.17]. Esta curva pode ser descrita como:

$$\begin{aligned} f(X_s, Y_s) &= 0 \\ Z_s &= 0 \end{aligned} \quad (5.20)$$

Um vetor normal a esta curva é dado por:

$$\vec{n} = \frac{\nabla f}{|\nabla f|} = \frac{\left( \frac{\partial f}{\partial X_s}, \frac{\partial f}{\partial Y_s} \right)^T}{|\nabla f|} \quad (5.21)$$

onde  $\nabla f$  é o vetor gradiente de  $f$ .

FIG. 5.17 CURVA PLANAR EM  $X_S, Y_S$ 

Denominando os termos  $\frac{\partial f}{\partial X_S}$  e  $\frac{\partial f}{\partial Y_S}$ , respectivamente de  $fX_S$  e  $fY_S$ , tem-se:

$$\vec{n} = \frac{(fX_S, fY_S)^T}{|\nabla f|} \quad (5.22)$$

onde

$$|\nabla f| = \sqrt{fX_S^2 + fY_S^2} \quad (5.23)$$

Um vetor tangente ao vetor  $\vec{n}$  é dado por.

$$\vec{t} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \vec{n} = \frac{(fX_s, fY_s)^T}{|\nabla f|} \quad (5.24)$$

A direção do vetor  $\vec{t}$  é a direção do vetor velocidade tangencial  $\vec{v}$  e a equação do movimento que gera a curva planar é.

$$\begin{bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{bmatrix} = \begin{bmatrix} \frac{V \cdot fY_s}{|\nabla f|} \\ \frac{V \cdot fX_s}{|\nabla f|} \\ 0 \end{bmatrix} \quad (5.25)$$

onde  $V$  é o módulo do vetor velocidade tangencial.

A equação (5.25) gera a curva planar mas é necessário que haja uma estratégia para recuperar a trajetória, caso ocorra um desvio. A estratégia para recuperação da trajetória é seguinte:

Definindo uma função penalidade  $J$  conforme a equação (5.26) tem-se:

$$J = \frac{f^2(X_s, Y_s) + Z_s^2}{2} \quad (5.26)$$

O gradiente desta função determina a direção do erro.

$$\nabla J = \begin{bmatrix} \frac{\partial f}{\partial X_s} \\ \frac{\partial f}{\partial Y_s} \\ \frac{\partial f}{\partial Z_s} \end{bmatrix} \triangleq \begin{bmatrix} f \cdot f X_s \\ f \cdot f Y_s \\ Z_s \end{bmatrix} \quad (5.27)$$

$$V_r = \begin{bmatrix} \frac{-K \cdot f \cdot f X_s}{|\nabla f|^2} \\ -K \cdot f \cdot f Y_s \\ \frac{|\nabla f|^2}{-K \cdot Z_s} \end{bmatrix} \quad (5.28)$$

onde  $K$  é o ganho de recuperação e  $|\nabla f|^2$  normaliza o vetor.

Adicionando-se  $\vec{V}_r$  à equação (5.25), tem-se a equação de movimento para geração e recuperação da curva:

$$\begin{aligned} \dot{X}_s &= \frac{V \cdot f Y_s}{|\nabla f|} - \frac{K \cdot f \cdot f X_s}{|\nabla f|^2} \\ \dot{Y}_s &= -\frac{V \cdot f X_s}{|\nabla f|} - \frac{K \cdot f \cdot f Y_s}{|\nabla f|^2} \\ \dot{Z}_s &= -K \cdot Z_s \end{aligned} \quad (5.29)$$

No caso do círculo centrado na origem do sistema de coordenadas  $X_s Y_s Z_s$  a função que o representa é

$$f = \sqrt{X_s^2 + Y_s^2} - R \quad (5.30)$$

Portanto

$$fX_S = \frac{X_S}{\sqrt{X_S^2 + Y_S^2}} \quad (5.31)$$

$$fY_S = \frac{Y_S}{\sqrt{X_S^2 + Y_S^2}}$$

$$VF = 1$$

e a velocidade de recuperação é

$$\dot{X}_{rs} = -K \left( \sqrt{X_S^2 + Y_S^2} - R \right) \frac{X_S}{\sqrt{X_S^2 + Y_S^2}} \quad (5.32)$$

$$\dot{Y}_{rs} = -K \left( \sqrt{X_S^2 + Y_S^2} - R \right) \frac{Y_S}{\sqrt{X_S^2 + Y_S^2}}$$

$$\dot{Z}_{rs} = -K Z_S$$

Portanto a equação dos componentes  $\dot{X}_S \dot{Y}_S \dot{Z}_S$  para geração e recuperação da trajetória circular é

$$\dot{X}_S = \frac{V \cdot Y_S - K \left( \sqrt{X_S^2 + Y_S^2} - R \right) X_S}{\sqrt{X_S^2 + Y_S^2}} \quad (5.33)$$

$$\dot{Y}_S = \frac{V \cdot X_S - K \left( \sqrt{X_S^2 + Y_S^2} - R \right) Y_S}{\sqrt{X_S^2 + Y_S^2}}$$

$$\dot{Z}_S = -K Z_S$$

A transformação dos  $\dot{X}_s, \dot{Y}_s, \dot{Z}_s$  para velocidade das juntas é realizada da forma semelhante ao caso da trajetória retilínea.

## 5.4 EXPERIMENTOS REALIZADOS

### 5.4.1 Rastreamento de Trajetória Retilínea Realizados.

Foi realizado experimento de controle de trajetória retilínea na vertical, baseado no algoritmo descrito anteriormente.

A trajetória a ser rastreada pela garra é uma reta vertical de comprimento 200 mm. A velocidade tangencial aumenta gradativamente em forma de rampa em função de tempo até alcançar 80 mm/s e mantém esta velocidade até que se esteja a 20 mm do final na direção do movimento, ou seja, até percorrer 180 mm. A partir desse valor de erro a velocidade tangencial é calculada para uma frenagem gradativa no movimento, para que haja a parada suave.

A figura [5.18] ilustra as rotinas utilizadas para o controle de trajetória. Para melhor entendimento, será descrito a seguir os detalhes de algumas dessas rotinas mais complexas.

1. Cálculo do  $X$ ,  $\theta_{\text{OMBRO}}$  e  $\theta_{\text{COTOVELO}}$ .

$$X = 0,000635.CA1$$

$$\theta_{\text{OMBRO}} = 2,312740538 - \alpha_{\text{OMBRO}}$$

$$\theta_{\text{COTOVELO}} = 0,741947268 - \alpha_{\text{COTOVELO}}$$

onde CA1 é o valor lido do contador do eixo 1 e os cálculos do  $\alpha_{\text{OMBRO}}$  e  $\alpha_{\text{COTOVELO}}$  estão no capítulo 2.

2. Cálculo do  $X$ ,  $Y$ ,  $Z$  da ponta do braço

$$X = 0,000635.CA1$$

$$Y = 400.\cos\theta_{\text{OMBRO}} + 600.\cos\theta_{\text{COTOVELO}}$$

$$Z = 400.\sen\theta_{\text{OMBRO}} + 600.\sen\theta_{\text{COTOVELO}}$$

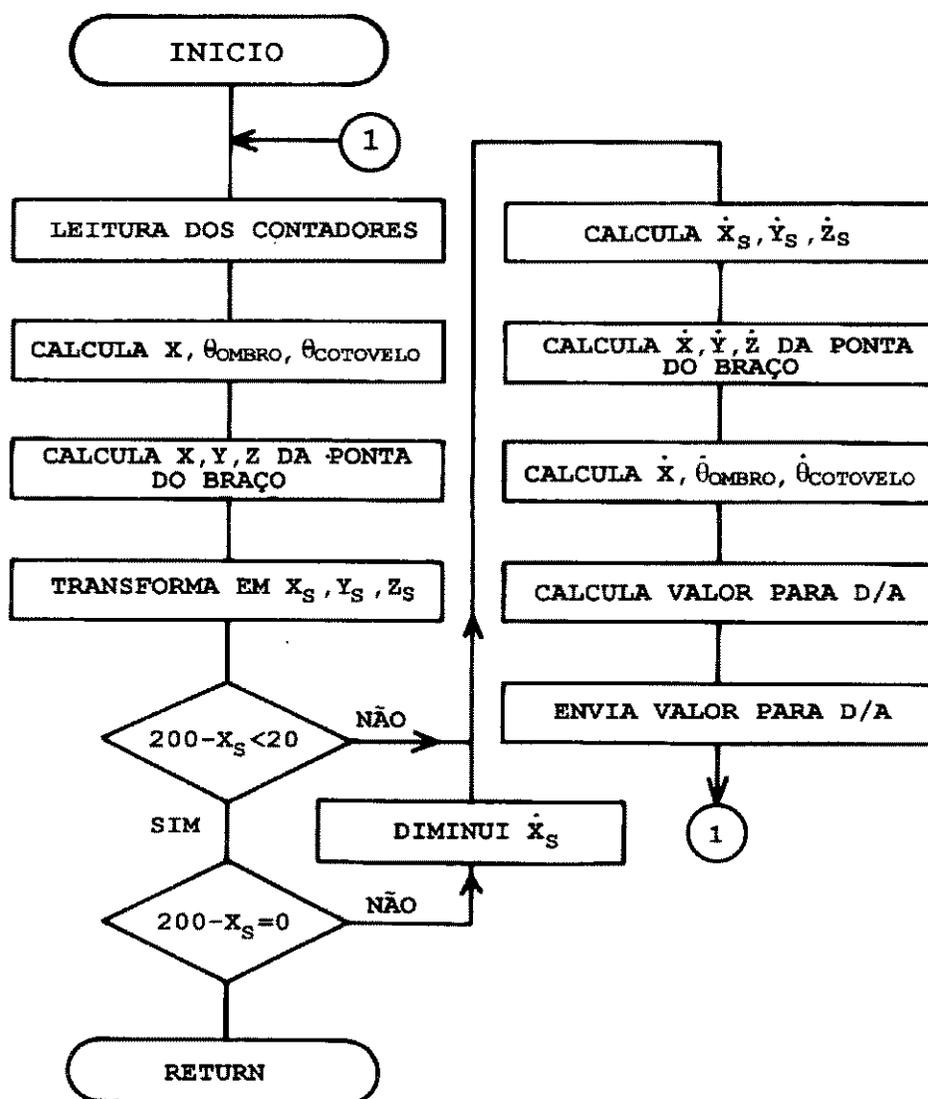


FIG. 5.18 ROTINAS DE CONTROLE DE TRAJETÓRIA

3. Transformação de  $X, Y, Z$  em  $X_s, Y_s, Z_s$

Como no presente caso a trajetória é uma reta vertical de cima para baixo

$$X_s = -Z + Z_0$$

$$Y_s = Y - Y_0$$

$$Z_s = X - X_0$$

4. Cálculo do  $\dot{X}_s, \dot{Y}_s, \dot{Z}_s$  onde  $V$  é o módulo da velocidade tangencial

$$\begin{bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{bmatrix} = \begin{bmatrix} V \\ -K_{REC} \cdot Y_s \\ -K_{REC} \cdot Z_s \end{bmatrix}$$

5. Cálculo do  $\dot{X}, \dot{Y}, \dot{Z}$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = [L^T] \cdot \begin{bmatrix} \dot{X}_s \\ \dot{Y}_s \\ \dot{Z}_s \end{bmatrix}$$

6. Cálculo do  $\dot{X}, \dot{\theta}_{OMBRO}$  e  $\dot{\theta}_{COTOVELO}$

$$\begin{bmatrix} \dot{X} \\ \dot{\theta}_{OMBRO} \\ \dot{\theta}_{COTOVELO} \end{bmatrix} = [J^{-1}] \cdot \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix}$$

7. Cálculo dos valores a serem colocados no conversor D/A dos respectivos eixos. Veja as equações (5.17), (5.18) e (5.19)

A figura [5.19] mostra o erro de rastreamento na direção  $Y_s$ , medido através dos sensores internos do robô, onde

pode-se observar que este erro é menor que 0,3 mm, no valor de pico e o seu valor médio está em torno de 0,1 mm, dentro das aplicações mais usuais nas indústrias, o que é um resultado bastante satisfatório.

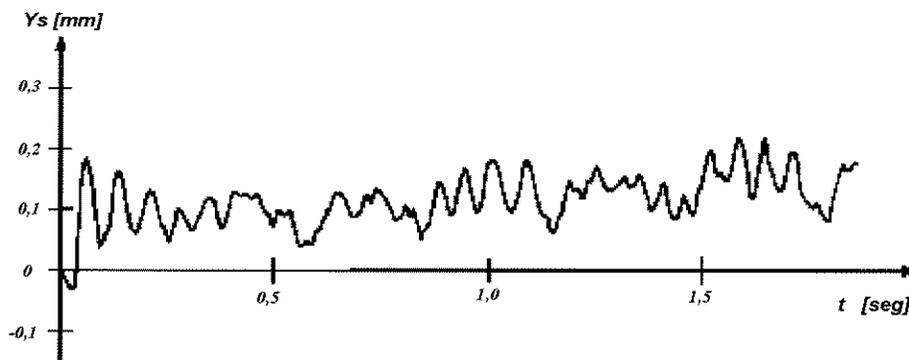


FIG. 5.19 GRÁFICO DO ERRO NA DIREÇÃO  $Y_s$

#### 5.4.2 Agarramento do Objeto Visto pelo Sensor de Visão.

Conforme o capítulo 4, a câmera do sensor de visão está situada em posição perpendicular à mesa de trabalho e fornece as coordenadas  $XY$  do centro do objeto e o ângulo de inclinação do eixo menor que passa pelo centro do objeto. Em outras palavras, o sensor de visão fornece as coordenadas  $XY$  da ponta do braço e o ângulo  $\theta_{ROTGR}$ , sendo que  $\theta_{MERG}$  é fixado em  $-90^\circ$  (figura [5.20]).

Como o valor de  $\theta_{ROTGR}$  necessário para que a garra agarre o objeto pelo menor eixo é fornecido por sensor de visão, o controlador do braço deve calcular o valor de  $\theta_4$  e  $\theta_5$  que alcancem  $\theta_{MERG} = 90^\circ$  e  $\theta_{ROTGR}$ , conforme algoritmo para o calculo dos ângulos do cotovelo e do ombro a partir das coordenadas  $YZ$  da ponta, desenvolvido anteriormente .



considerável.

Desta forma, foi adotado o seguinte procedimento para agarrar objeto visto pelo sensor de visão:

1. Inicialmente o controlador do braço recebe as informações das coordenadas XY do centro do objeto e ângulo  $\theta_{ROTGR}$ .
2. A ponta do braço é posicionada nas coordenadas XY fornecidas pelo sensor de visão, sendo a coordenada Z correspondente a 200 mm acima da mesa, com ângulo de mergulho de  $-90^\circ$  graus e  $\theta_{ROTGR}$  fornecido por sensor de visão.
3. Após alcançado este ponto, inicia-se o controle de trajetória vertical descendente da ponta do braço e para cada iteração são calculados os novos valores de referência de  $\theta_4$  e  $\theta_5$  para manter  $\theta_{MERG}$  e  $\theta_{ROTGR}$ , e realiza-se o controle de posição para  $\theta_4$  e  $\theta_5$ .

O algoritmo acima descrito foi testado na prática, obtendo-se o resultado esperado.

## **CAPÍTULO 6**

### **CONCLUSÃO**

#### **6.1 ESTRUTURA MECÂNICA DE POSICIONAMENTO E ORIENTAÇÃO**

A estrutura mecânica desenvolvida no presente trabalho teve como principal preocupação a sua simplicidade construtiva e sua robustez, de tal forma que a sua reprodução seja possibilitada com equipamentos simples de usinagem e com componentes facilmente acessíveis. Foi escolhido o mecanismo de paralelogramo para as juntas do braço, utilizando o sistema de fuso-castanha como redutor confiável e de baixo custo. O mecanismo desenvolvido apresenta boa confiabilidade e simplicidade na construção conforme o objetivo inicial.

Os servomotores e acionadores eletrônicos foram selecionados conforme a necessidade do torque de acionamento, realizando-se um estudo baseado no modelo dinâmico do robô e posteriormente consultando-se catálogos de servomotores e seus respectivos acionadores eletrônicos de boa qualidade, para assegurar a resposta rápida e fiel ao comando.

O sensor de posição (codificador angular) é conectado no eixo do motor de cada uma das juntas, aumentando assim a sua resolução pela mesma taxa de redução do redutor.

Estes fatos têm facilitado o controle de movimento, obtendo como resultado, o rastreamento da trajetória retilínea pela ponta do braço, com precisão satisfatória para aplicações industriais.

#### **6.2 GARRA COM SENSORES**

A integração dos sensores na garra do robô industrial demonstrou, na prática, um efeito significativo. Com a sua

composição mecânica bastante simples, a garra conseguiu segurar e carregar um objeto frágil, um ovo de galinha, sem quebrá-lo.

A atuação do sensor de deslizamento no controle de agarramento e no controle de quando e como deve ser largado um objeto foi bastante eficiente, possibilitando uma forma de largar objeto deixando-o deslizar lentamente entre os dedos, ou largando imediatamente o objeto assim que o mesmo toque no patamar onde deve ser largado.

Quanto ao sensor de força, foram desenvolvidos dois tipos: um deles baseado em *strain gage* com circuitos analógicos e um outro com dispositivo digital. A comparação entre esses sensores tem demonstrado a maior imunidade do sensor digital em relação ao ruído eletromagnético.

### 6.3 SENSOR DE VISÃO

O sensor de visão desenvolvido alcançou os objetivos iniciais de se projetar e construir um digitalizador simples e de baixo custo que realize a digitalização da imagem capturada por câmera de TV, com resoluções suficientes para extrair desta as informações necessárias, tais como as coordenadas do centro e a orientação e a identificação do objeto, que são suficientes para a melhoria do desempenho da maioria das aplicações industriais simples.

Algumas melhorias podem ser feitas ao sistema de sensor de visão:

1. A resolução do sistema, que é de 160X200 *pixels* pode ser aumentada, com relativa facilidade, para 320X200 *pixels* ou para 320X400 *pixels*, da seguinte maneira: Para aumentar a resolução horizontal basta alterar o módulo de alguns contadores e o tempo de aquisição permanecerá inalterado. Para aumentar a resolução vertical seria necessário utilizar a informação do campo par (o sistema atual utiliza somente o campo ímpar) do sistema de vídeo entrelaçado e portanto será necessário dobrar o tempo de

aquisição, o que não deve acarretar problema para a maioria das aplicações industriais. Entretanto, no presente trabalho foi definida a resolução em 160X200 por causa do tempo de processamento do computador.

2. O sistema trabalha com apenas dois níveis para cada *pixel*. Com a utilização do conversor A/D de alta velocidade, ao invés do comparador simples, poder-se-ia trabalhar com vários níveis de cinza que possibilitam o processamento para se distinguir o contorno real do falso contorno [7]. Entretanto, este problema de falso contorno foi resolvido com a técnica de iluminação adequada.
3. O sistema pode evoluir ainda, através da implementação de *software* para processamento da imagem multi-objetos ou objetos com mais de uma borda fechada, ou seja, identificação e determinação de parâmetros de mais de um objeto por tela, ou objetos com furos.
4. Finalmente, a incorporação de mais uma câmera de TV, ou mesmo apenas uma câmera de TV com a utilização de um sistema de iluminação estruturada [7], faria com que o sistema tivesse informação de profundidade, permitindo assim uma análise tridimensional [8].

No presente trabalho foram desenvolvidos um manipulador robótico para fins didáticos e um sensor de visão, e foi realizada a integração entre esses dois sistemas.

A utilização do sensor de visão no processo de agarramento de objeto por robô industrial aumenta a flexibilidade da programação e a confiabilidade deste processo.

#### 6.4 SISTEMA HIERÁRQUICO DE CONTROLE DE POSIÇÃO E ORIENTAÇÃO

O microcomputador comercial, compatível ao IBM-PC-AT/386 de 40 MHZ, utilizado como computador de controle do manipulador, no presente trabalho, tem-se mostrado, no decorrer do desenvolvimento, como um equipamento adequado, pela sua comodidade e confiabilidade. Uma das vantagens deste equipamento como computador de controle é o fato de poder

migrar facilmente para um outro equipamento mais veloz da mesma família, quando a sua capacidade for insuficiente para processar um determinado algoritmo de controle em tempo real. Os algoritmos de controle utilizados no presente trabalho são processados em 5 ms neste equipamento.

Foi adotada a filosofia de controle hierárquico e de autonomia em cada nível de hierarquia e com isso foi possível realizar desenvolvimento por etapas bem definidas, assegurando a confiabilidade de cada nível hierárquico de controle e conseqüentemente, a confiabilidade global do sistema. Além disso a adoção desta filosofia facilita a implantação de outras estratégias de controle que está em andamento em outro trabalho.

O recurso de armazenamento de dados de movimentos e recurso gráfico têm-se mostrado ser uma ferramenta muito importante para desenvolvimento do software de controle de movimento, pois estes recursos permitem, facilmente, a visualização de dados de movimentos tais como a posição e a velocidade de cada uma das juntas, em função do tempo, oferecendo, desta maneira, um ambiente confortável para desenvolvimento.

O método de programação de tarefas do robô, em duas etapas, desenvolvido no presente trabalho, tem-se mostrado, na prática, ser um método fácil e flexível pois a programação das coordenadas dos pontos de trabalho e de passagem e a programação da seqüência dos pontos a serem alcançados são realizadas distintamente.

## APÊNDICE A

### *HARDWARE* DO SENSOR DE VISÃO

#### A.1 SINAL DE VÍDEO COMPOSTO.

O sinal de vídeo composto [10] é assim denominado por ser um sinal com a composição das seguintes informações:

1. informação da imagem propriamente dita;
2. pulsos de apagamento;
3. pulsos de sincronismo.

A informação da imagem de uma linha da varredura é localizada entre os pulsos de apagamento e pulsos de sincronismo e a informação da imagem está em nível analógico de tensão ou corrente.

O pulso de apagamento tem a finalidade de fazer com que as linhas de retraço (verticais e horizontais) não sejam visualizadas, mudando a amplitude do sinal para o nível correspondente ao preto. Os pulsos de apagamento horizontal tem a função de apagar os retraços da esquerda para a direita em cada linha de varredura horizontal. As linhas de varredura são apagadas pelos pulsos de apagamento vertical, quando o feixe de elétrons volta de baixo para o topo da imagem em cada campo.

Os pulsos de sincronismo são transmitidos juntamente com as informações de vídeo, para manter a transmissão e a recepção sincronizadas, permitindo desta forma a reprodução da imagem na posição correta num quadro.

Os pulsos de sincronismo ocorrem durante o retraço quando nenhuma informação está sendo transmitida. O sincronismo horizontal acontece ao fim de cada linha e marca o início do retraço horizontal. Já o sincronismo vertical ocorre no final de cada campo e marca o início do retraço vertical.

Na figura [A.1] é mostrado o sinal de vídeo composto em função do tempo, correspondente a duas linhas horizontais de imagem.

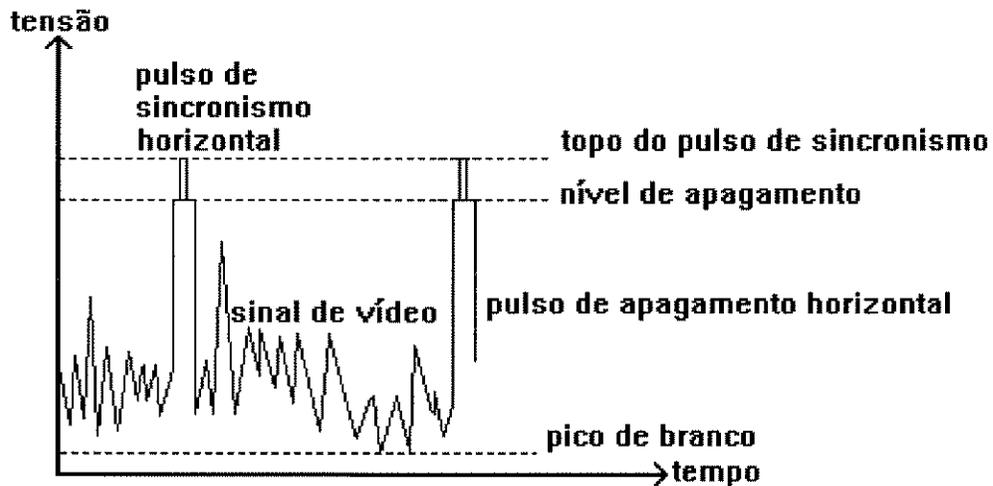


FIG. A.1 SINAL DE VÍDEO COMPOSTO

## A.2 DIGITALIZAÇÃO DA INFORMAÇÃO DA IMAGEM.

Após a verificação da composição do sinal de vídeo composto para compor uma imagem, nota-se que a digitalização da imagem deve ser realizada linha por linha. Para isso, é necessário que haja a extração do sinal que contém a informação da imagem de uma linha e como uma linha de varredura horizontal tem uma duração de  $63,5 \mu\text{s}$ , deverá ser feita uma amostragem periódica cujo período de amostragem depende da resolução desejada da imagem digitalizada. Por exemplo, se é desejada uma resolução digital de 400 elementos de imagem ou *pixels* na horizontal, o período de amostragem deve ser de  $0,15875 \mu\text{s}$  ( $63,5 \mu\text{s}/400$ ).

A informação da intensidade luminosa de um *pixel* está contida na amplitude do sinal de imagem. Para digitalizar esta informação utiliza-se um conversor analógico-digital (A/D) transformando o nível de tensão ou corrente analógico em

informações digitais. A única restrição nesta conversão A/D é o tempo de conversão que é dependente do método de conversão (dupla rampa, aproximação sucessiva, *flash*, etc.) e da sua resolução (4 bits, 8 bits, 10 bits, etc.). Um conversor A/D de 8 bits comercialmente acessível que utiliza o método *half flash* tem um tempo de conversão de 1 $\mu$ s, portanto daria para se ter uma resolução máxima de 63 *pixels* numa linha horizontal. Desta forma, a obtenção do *nível de cinza* (denominação da informação da intensidade luminosa de um *pixel*) e a resolução horizontal (quantidade de *pixel* contida em uma linha) está na dependência do conversor analógico digital comercialmente acessível.

Teoricamente é possível confeccionar um conversor A/D de alta velocidade, utilizando-se diversos comparadores paralelos com diferentes níveis de tensão de referência, mas para uma resolução de 8 bits, que corresponde a 256 níveis, tornaria necessário o uso de 256 circuitos comparadores com 256 linhas de saída digital e um circuito conversor de 256 linhas para 8 bits codificados em binário, o que acabaria impondo grande dificuldade na implementação prática. Atualmente existe um conversor A/D de alta velocidade dedicado para a utilização da conversão de sinal de vídeo, mas é praticamente impossível adquiri-lo, pois os fabricantes normalmente fornecem esses *chips* já embarcados na placa de digitalização de imagem.

A digitalização da imagem do objeto em vários níveis de cinza permite a recomposição fiel das informações dos objetos e quanto maior for a quantidade de níveis de cinza, maior serão os detalhes do objeto que podem ser adquiridos. A mesma afirmação é válida quanto a resolução da imagem digitalizada. Em outras palavras, a quantidade de níveis de cinza e a resolução necessária dependem do grau de informação que se deseja extrair da imagem do objeto.

O efeito da pequena quantidade de níveis de cinza, menor que 32 níveis, provoca o problema de *falso contorno*, ou seja, a diferença na luminosidade, causada pela curvatura da superfície do objeto, é digitalizada erroneamente como área

branca ou preta, confundindo-se com o fundo, informando ao sistema de processamento um falso contorno do objeto.

A baixa resolução, por sua vez, muda a forma dos contornos do objeto. Uma reta inclinada, por exemplo, pode ser interpretado como uma junção de pequenos retângulos em forma de escada, ou um contorno suave de um objeto pode ser digitalizado em um contorno composto de quadrados, como é ilustrado na figura [A.2].

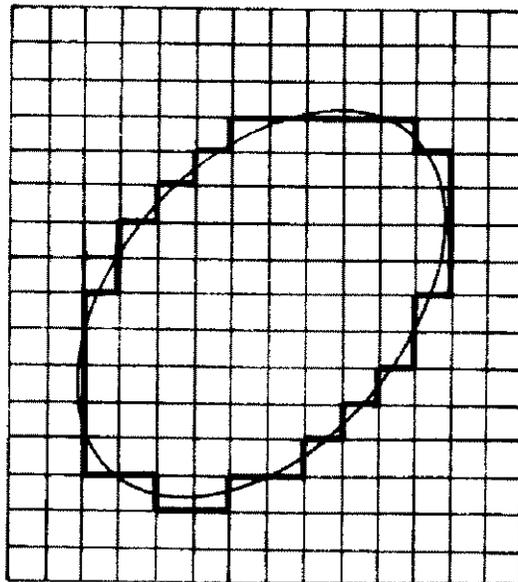


FIG. A.2 DIGITALIZAÇÃO EM BAIXA RESOLUÇÃO

Por outro lado, a técnica de iluminação é um fator muito importante na aquisição da imagem do objeto e no reconhecimento da natureza da superfície. Portanto, deve ser

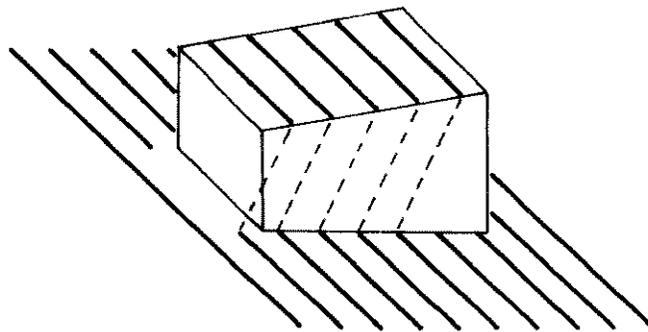
escolhido um sistema de iluminação adequado dependendo da informação que se pretende extrair.

Basicamente existem quatro tipos de iluminação, conforme classifica Fu [7]. São eles:

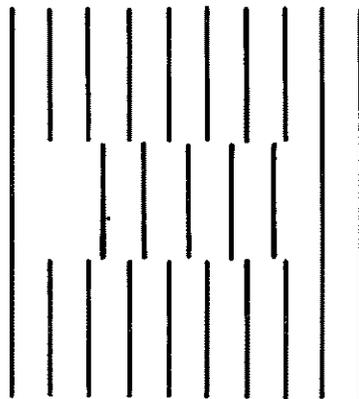
- a. Uma iluminação difusa, utilizada para objetos com superfícies suaves, onde é pretendido verificar as características das superfícies;
- b. A iluminação em contra-luz, ou seja, quando o objeto é colocado entre a fonte de luz de feixes paralelos e a câmera de TV, produzindo uma imagem binária (preto e branco), adequada para detectar apenas o contorno do objeto;
- c. Uma iluminação estruturada, composta de retas paralelas luminosas como está ilustrado na figura [A.3a], fornece a imagem mostrada na figura [A.3b] que contém a informação da altura do objeto, além do seu contorno, ou seja, contém informações tridimensionais;
- d. A iluminação direta de feixes paralelos, inclinados em relação a uma superfície, é adequada para detectar a rugosidade da superfície através das minúsculas sombras causadas pela superfície rugosa.

No presente trabalho foi definida a digitalização em dois níveis, ou seja, cada *pixel* amostrado é considerado como um ponto preto ou um ponto branco, utilizando-se para isso um circuito comparador de alta velocidade (figura [A.4a]). Como efeito, é conseguida uma imagem digitalizada de altíssimo contraste, o que facilita o processamento para extrair certas informações de interesse, nela contidas.

Por outro lado, desde que o objeto, cuja imagem é digitalizada, seja estático, é possível obter diversos níveis de cinza, variando-se a tensão de referência do comparador, através do circuito ilustrado na figura [A.4b]. O mesmo objeto seria digitalizado diversas vezes e em cada digitalização o valor digital que define o nível de referência



a



b

FIG. A.3 a) OBJETO COM ILUMINAÇÃO ESTRUTURADA  
b) IMAGEM ADQUIRIDA

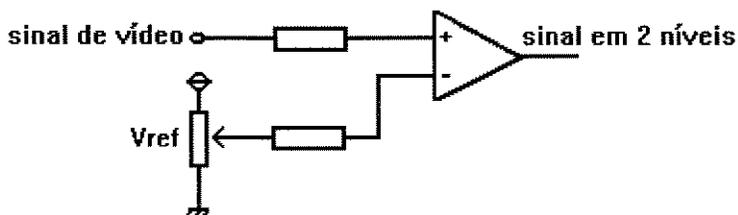


FIG. A.4a CIRCUITO COMPARADOR COM A TENSÃO DE REFERÊNCIA FIXA

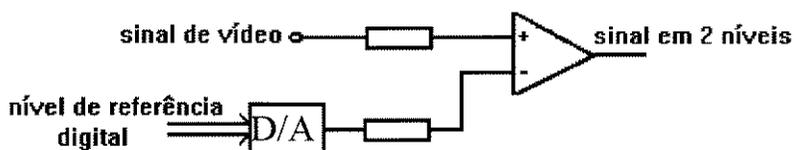


FIG. A.4b CIRCUITO COMPARADOR COM A TENSÃO COMANDADA DIGITALMENTE

do comparador é variado e com isso, cada digitalização corresponderá a um nível de cinza. Nesta solução simples, comparada com a utilização do conversor A/D de alta velocidade, naturalmente há um preço a pagar que é o tempo de digitalização, na medida que neste caso a digitalização deve

ser feita em múltiplas vezes. No entanto, esta é uma solução possível e imposta pela dificuldade de se adquirir conversores A/D de alta velocidade.

A resolução vertical é definida pela quantidade de linhas horizontais de uma tela em padrão *NTSC* que é de 525 linhas sendo 262,5 linhas compondo o campo par e 262,5 linhas para o campo ímpar (este método de compor uma tela de imagem é denominado de sistema entrelaçado).

No presente trabalho, o sensor de imagem tem as seguintes características:

1. A dimensão da janela em relação a tela normal de imagem do padrão *NTSC* é definida como ilustrado na figura [A.5];
2. A resolução desta janela é de 160X200 *pixels*, podendo facilmente ser ampliado para 320X200 *pixels*. Apesar da vantagem da alta resolução, manteve-se em 160x200 *pixels* por falta de capacidade de processamento em tempo real do computador;
3. A informação de luminosidade de cada *pixel* é em 2 níveis, sendo facilmente ampliado em diversos níveis de cinza através da utilização do conversor D/A para variar a tensão de referência do comparador associada à múltipla digitalização.

### A.3 CIRCUITO DE EXTRAÇÃO DO SINAL DE VÍDEO.

Como foi citado anteriormente, o dispositivo de captura da imagem emite o sinal de vídeo composto de onde deve ser extraído o sinal da imagem propriamente dita.

Tecnicamente, extrair o sinal da imagem significa desenvolver circuitos que identifiquem o início da primeira linha horizontal de um campo, esperar a quantidade de linhas horizontais que não fazem parte da janela ilustrada na figura [4.6] e reconhecer o início da primeira linha pertencente à janela. No nosso caso a primeira linha da janela é a 33<sup>a</sup> linha.

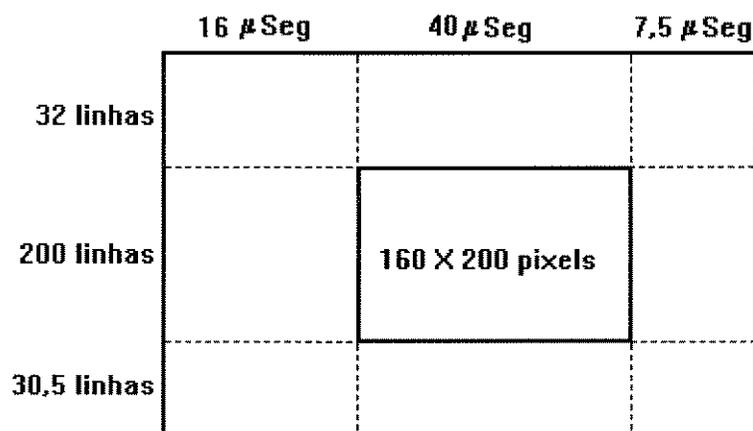


FIG. A.5 DIMENSÃO DA JANELA

Ao detectar o início de uma linha pertencente à janela, deve-se esperar 16  $\mu$ s para iniciar o armazenamento da informação digitalizada, o que deve permanecer por 40  $\mu$ s. O período de amostragem para digitalização é de 250 ns para resultar em 160 *pixels*/linha. Este procedimento é repetido 200 vezes quando termina a janela, para concluir a digitalização e armazenamento de um quadro.

O procedimento acima descrito deixa clara a necessidade de detectar e isolar o sinal de sincronismo horizontal, que define o início de uma linha horizontal, o sinal de sincronismo vertical, que define o início de um quadro da imagem, e o sinal de distinção de campo, que faz a distinção do campo par e do campo ímpar. Esta separação de sincronismo foi realizada utilizando-se um circuito integrado LM1881 da *National Semiconductor* que é dedicado para esta finalidade.

Quando se injeta o sinal de vídeo composto na entrada deste circuito integrado, obtém-se na saída o sinal de sincronismo vertical e horizontal e o sinal de distinção de campo, todos compatíveis a nível TTL.

Baseado nesses sinais de sincronismo vertical e horizontal e de distinção de campo, é realizado o armazenamento do sinal de vídeo digitalizado na memória de 8 bits X 4 KByte.

A seguir é descrito o comportamento do circuito das figuras [A.6] e [A.7].

O *flip flop* FF4 recebe sinal de *START* do computador e sua saída Q torna-se ALTO liberando FF3. O FF3 liberado espera a chegada do sinal de início do 1º campo, que virá do circuito separador de sincronismo.

A saída Q do *flip flop* FF1 torna-se ALTO, ao iniciar o 1º campo, liberando FF2 e o contador CT1. O CT1 é responsável por esperar 32 pulsos de sincronismo horizontal, e após esses 32 pulsos a saída desse contador envia o sinal ALTO na entrada de *clock* do FF2, que já estava liberado no início do 1º campo. Desta forma a saída do FF2 torna-se ALTO levando a saída do FF3 para ALTO. A partir deste instante, realiza-se a aquisição de dados durante 200 pulsos de sincronismo horizontal. A contagem desses 200 pulsos que indica o final da aquisição, é feita de forma indireta. Aproveitando a evolução do contador que fornece o endereço da memória que armazena a informação da imagem digitalizada, a saída do decodificador envia o sinal de fim de aquisição ao *flip-flop* FF4 quando o endereço da memória atingir o valor FA1 (HEX) que corresponde a 4001 (DEC), zera a sua saída e ao mesmo tempo desativa FF3 com o seu *CLEAR* mantendo-se ativado, concluindo a aquisição de dados da imagem.

Em outras palavras, são feitas aquisições de 200 linhas horizontais de imagem.

A aquisição de dados da imagem contida em uma linha é feita da seguinte maneira:

A informação da imagem contida em uma linha é recolhida em 160 *pixels*.

Para cada sinal de sincronismo horizontal, FF7 torna-se ALTO, liberando CT2 que emite *clock* de 8 MHz.

Em seguida, a saída do FF6 torna-se ALTO após esperar 16  $\mu$ s (correspondente a 128 pulsos de 8 MHz) causada por CT3 e CT4.

Com a saída Q do FF6 em ALTO o *clock* de 8 MHz é liberado e injetado no contador CT6, que foi liberado pela





saída Q do FF6. A saída do CT6 é uma frequência de 4MHz que é injetada no *clock* do Registrador de Deslocamento, em cuja entrada de dados é injetado o sinal de vídeo após o comparador. Em outras palavras, o Registrador de Deslocamento faz a aquisição do sinal de vídeo digitalizado para cada 250 ns, deslocando sucessivamente a informação obtida para próximo bit.

Após entrar 8 bits de informação no Registrador de Deslocamento, a saída paralela deste é habilitada pelo circuito IC12-A que ao mesmo tempo envia o pulso de escrita na memória.

O circuito IC12-B é responsável por enviar pulsos que incrementa o contador de endereço da memória sempre que for escrita uma informação de 8 bits na memória.

Quando o endereço da memória atinge o valor FA1H, o circuito decodificador emite o sinal de final de aquisição no FF4, que faz com que a sua saída seja BAIXO. Esta saída BAIXO do FF4 tranca FF3 (*CLEAR* = 0) e a saída Q deste tranca FF7, que por sua vez tranca CT2, que impede a passagem do sinal de *clock*.

A saída do IC22 (LS08 - *AND*) tranca também o FF6 e sua saída Q tranca CT3 e CT4 e também o CT5.

Em outras palavras, no final da aquisição são recuperados os estados iniciais dos diversos *flip flop* e dos contadores, inclusive o contador de endereços da memória.

Entretanto, é necessário no início, ou seja, logo após energizar o circuito, escrever no endereço \$309H qualquer valor que tenha o bit LSB nulo (*Do* = 0) para que a porta *OR* (IC19-B), cuja saída está ligada no *clock* do contador, libere este contador a fim de que IC12-B possa comandar o incremento de endereço da memória. Também é necessário que mantenha a saída do *flip flop* do IC18 - pino 9 em BAIXO enquanto estiver em aquisição.

O início da digitalização e aquisição da imagem digitalizada é comandado pelo computador hospedeiro, escrevendo qualquer valor no endereço \$308H. A conclusão da aquisição é

monitorada por computador, verificando o bit LSB do endereço \$309H. Se  $D_0 = 1$  a digitalização e aquisição foram concluídas.

Após a conclusão da digitalização e aquisição da imagem, a imagem adquirida está armazenada na memória desta placa. Estes dados deverão ser transferidos para a memória do computador hospedeiro para o processamento.

A transferência desses dados são realizados através do *tri-state* alocado no endereço \$308H do I/O. Os endereços da memória da placa de aquisição são definidos novamente por contadores compostos por IC15 a IC17, que inicialmente devem ser zerados pelo *flip flop* do IC18, que é ativado por bit LSB do endereço \$30AH do computador hospedeiro.

Após este procedimento de zerar o endereço da memória da placa, o computador hospedeiro realiza a leitura da *tri-state* localizado no endereço \$308H e posteriormente envia um pulso no *clock* do *flip flop* IC18 através do endereço \$309H para incrementar o contador que define o endereço da memória da placa. Este procedimento - leitura de dados do endereço \$308H e incremento de endereço - será repetido até completar a leitura completa de uma imagem ou seja  $\frac{(160 \times 200)}{8} = 4000$  vezes.

Ao analisar este circuito de digitalização e aquisição de imagem, nota-se que o interfaceamento com o computador hospedeiro é extremamente simples, podendo ser utilizado qualquer computador como hospedeiro. Esta facilidade no interfaceamento foi um dos objetivos ao se desenvolver este circuito, pois o mesmo pode ser interfaceado com um computador baseado em DSP (*Digital Signal Processor*), caracterizado pelo seu baixo custo e alta velocidade, além de permitir a compactação do sensor de visão.

A figura [A.8] ilustra a localização dos *pixels* na tela e a sua função imagem correspondente. A alocação da função imagem na memória da placa digitalizadora está indicada na tabela [A.1].

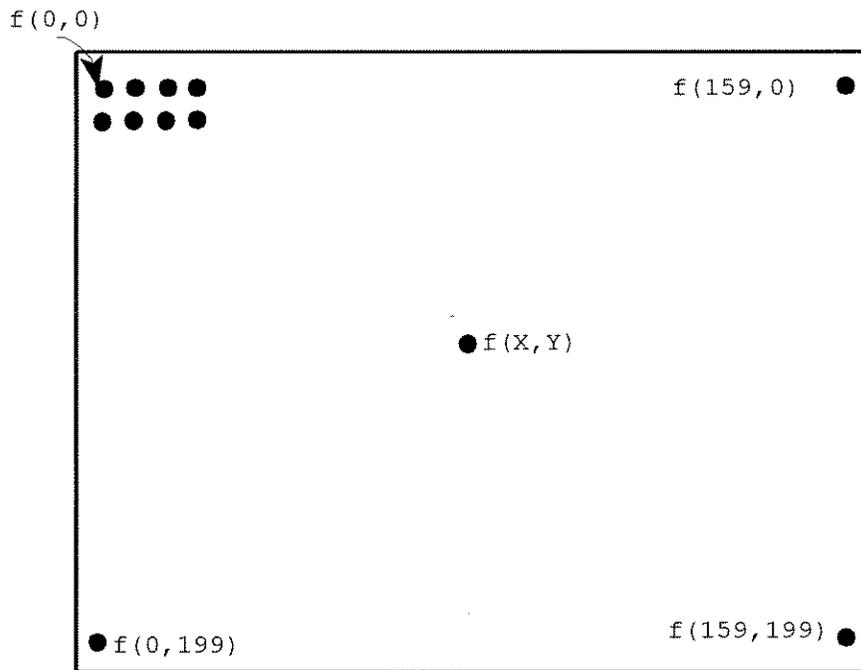


FIG. A.8 LOCALIZAÇÃO DOS PIXELS

END. (DEC)	BIT 0	BIT 1	...	BIT 7
000	$f(0,0)$	$f(1,0)$	...	$f(7,0)$
001	$f(8,0)$	$f(9,0)$	...	$f(15,0)$
.	.	.		.
.	.	.		.
.	.	.		.
019	$f(152,0)$	$f(153,0)$	...	$f(159,0)$
020	$f(0,1)$	$f(1,1)$	...	$f(7,1)$
.	.	.		.
.	.	.		.
.	.	.		.
4000	$f(152,199)$	$f(153,199)$	...	$f(159,199)$

TABELA A.1 ALOCAÇÃO DA FUNÇÃO IMAGEM NA MEMÓRIA

## APÊNDICE B

### ***HARDWARE* DO CONTROLADOR DO ROBÔ**

#### **B.1 DESCRIÇÃO DE CIRCUITOS DE INTERFACEAMENTO.**

O circuito da placa de interfaceamento é dividido em 4 partes que são:

- a. Decodificador de endereços
- b. Contador *up/down*
- c. Conversor D/A
- d. Entrada digital de 16 bits

A seguir são descritos alguns detalhes desses circuitos.

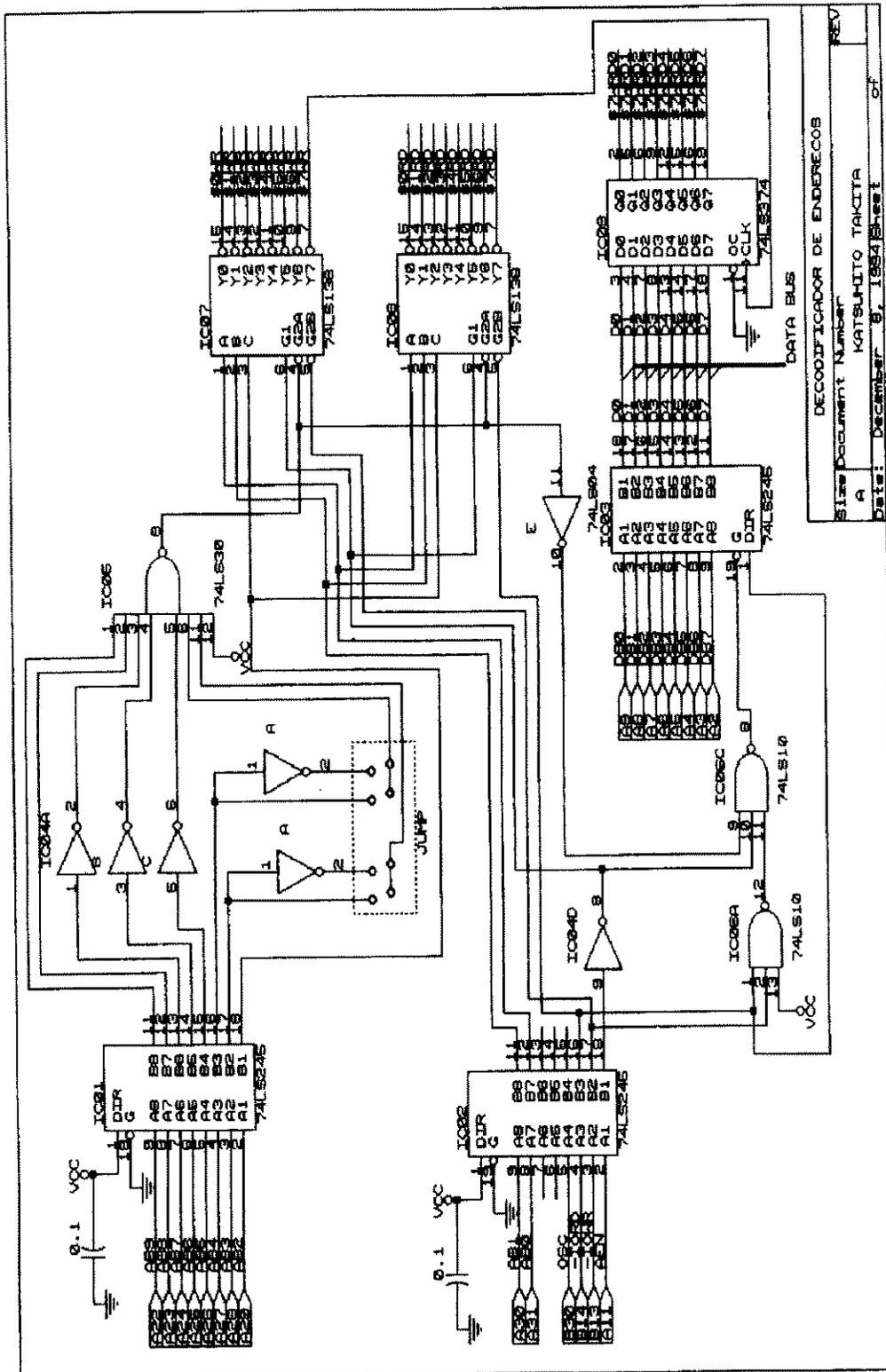
##### **B.1.1 Decodificador de Endereços.**

A figura [B.1] ilustra o diagrama esquemático deste circuito.

Neste circuito, foram utilizados *buffers* em todas as linhas de sinal do microcomputador, pois na descrição técnica dos *slots* do microcomputador consta que cada linha de sinal do *slot* consegue acionar seguramente apenas um dispositivo compatível a TTL tipo LS.

Foram decodificados apenas  $A_0$  a  $A_9$  do barramento de endereços baseado na informação de que não é necessário decodificar outros bits superiores.

Como esta placa seria confeccionada em circuito impresso, foram colocados *jumpeamento* nos bits AB3 e AB4 que possibilitam a mudança dos endereços da placa. Os conjuntos de endereços possíveis são 300(Hex) a 307(Hex), 308(Hex) a 30F(Hex), 310(Hex) a 317(Hex) e 318(Hex) a 31F(Hex). Estes endereços foram escolhidos baseado na informação fornecida pela



DECODIFICADOR DE ENDEREÇOS  
 Silve Document Number  
 A  
 KATSUNITO TAKITA  
 Date: December 9, 1984 Sheet 01

FIG. B.1 DECODIFICADOR DE ENDEREÇO

IBM de que os endereços 300(Hex) a 31F(Hex) estão reservados para o usuário.

Foram decodificados separadamente os endereços de leitura e escrita com 2 demultiplexadores, gerando assim 8 endereços para leitura e 8 endereços para a escrita, indicados como \$0RD a \$7RD e \$0WR a \$7WR no circuito. Entretanto, como foi utilizado duplo *latch* em cada conversor D/A de 10 bits, ocupando o total de 6 endereços e foi necessário circuito de *latch* para saídas dos contadores *up/down*, que será explicado posteriormente, além de 3 endereços para possibilitar a inicialização individual dos 3 contadores, os 8 endereços para a escrita foram insuficientes.

Para resolver este problema, foi adotado o método de endereçamento em duas etapas para alguns dispositivos. Neste método de endereçamento é colocado um *latch*, alocado num dos endereços em cujas entradas está o barramento de dados e as saídas geram endereços indiretos. Naturalmente estes endereços indiretos não servem para escrita de dados por não estarem sincronizados com o sinal de controle de escrita mas servem como sinal para zerar ou segurar as saídas dos contadores.

Foram utilizados, nesta placa, apenas os 8 bits do barramento de dados.

### B.1.2 Contador *UP/DOWN*.

O contador *up/down*, figura [B.2], é utilizado para contar os pulsos gerados por sensor de posição, que é um codificador angular incremental de duas fases. Neste codificador, as duas fases estão defasadas de 90° entre si e dependendo do sentido de rotação do seu eixo, o avanço de uma fase em relação à outra é trocado, como é ilustrado na figura [B.3]. Observa-se nesta figura que quando o eixo do codificador gira no sentido horário a fase A está adiantada em relação a fase B e o inverso acontece quando muda o sentido de rotação do eixo.



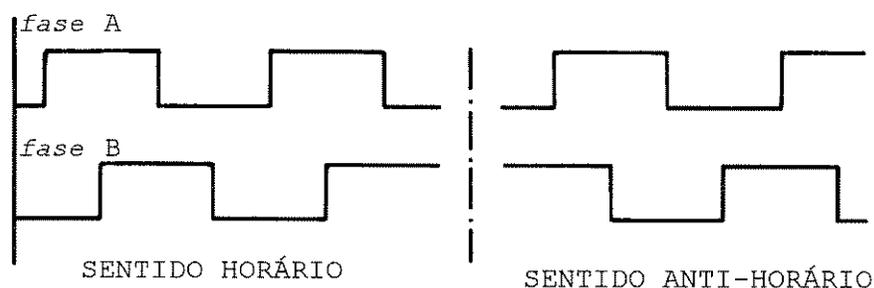


FIG. B.3 SINAIS DE 2 FASES EMITIDOS PELO CODIFICADOR ANGULAR

Desta forma, o sentido de contagem de pulsos do contador *up/down* é dependente desse defasamento. O circuito composto de IC12, IC21, IC7 e IC60 cria sinais de *clock* da ordem de 1 MHz defasados de  $90^\circ$  que são injetados no *CLOCK* do IC13 e *ENABLE* do IC14, respectivamente e no IC13 são injetadas as duas fases (FASE A e B) emitidas pelo codificador angular.

As saídas 1 e 2 do demultiplexador 4 por 16 (IC14) emitem sinais dependendo do sentido de rotação do codificador angular. Quando a fase A é atrasada em relação a fase B (sentido anti-horário), apenas a saída 2 emite pulsos, incrementando o contador. A saída 1 emite pulsos somente quando a fase A estiver adiantada em relação à fase B (sentido horário), decrementando o contador.

Ao observar conjuntamente as fases A e B na figura [B.4] nota-se que existem quatro mudanças de estado num ciclo, significando que é possível multiplicar a quantidade de pulsos.

As portas *NAND* de quatro entradas, em cujas entradas estão as saídas do demultiplexador, realiza esta multiplicação por 4.

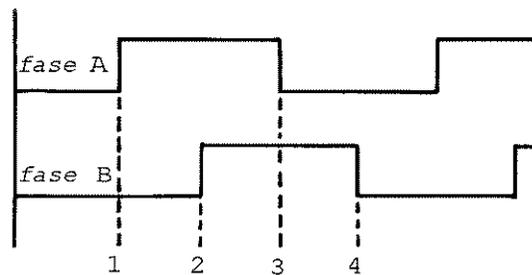


FIG. B.4 MUDANÇAS DE ESTADOS NUM CICLO DO CODIFICADOR ANGULAR

Os pulsos de incremento ou de decremento assim gerados, que interpretam o sentido de rotação do codificador angular e multiplicam a sua resolução, são injetados nas entradas *UP* e *DOWN*, respectivamente, do contador que está cascadeado com outros três contadores, compondo assim um contador de 16 bits.

Os dois *latches* colocados nas saídas dos contadores são para evitar o erro de leitura numa situação quando estes contadores estão na iminência de receber um novo pulso e que haja a mudança de contagem entre a leitura de primeiros 8 bits com os 8 bits seguintes.

Como pode ser observado na figura [B.2], o *clock* dos dois *latches* são acionados simultaneamente, para realizar posteriormente a leitura individual.

Atualmente existe um chip que realiza a função semelhante a deste circuito desenvolvido no presente trabalho, como, por exemplo, o PCC140 da *COSMO SYSTEM*, mas na época que

este circuito foi desenvolvido, tal chip não estava disponível no mercado nacional.

### B.1.3 CONVERSOR D/A.

Os conversores D/A de 10 bits foram implementados, utilizando-se o circuito integrado da *Analog Devices* AD7520. O circuito referente a um canal de conversor D/A é ilustrado na figura [B.5].

Como foram utilizados apenas 8 bits do barramento de dados do microcomputador, foram colocados *latch* duplo na entrada desse conversor para possibilitar a entrada simultânea de 10 bits com barramento de 8 bits.

Desta forma, para escrever um valor numérico de 10 bits na entrada do conversor é obrigatório que se escreva inicialmente os 2 bits mais significativos, que ficam retidos no *latch* intermediário (IC41), e ao escrever os 8 bits menos significativos, estes 2 bits anteriormente retidos no *latch* intermediário são liberados e retidos no segundo *latch* juntamente com outros 8 bits.

Este conversor, AD7520 é basicamente um conversor com saída de corrente e um amplificador operacional é utilizado para conversão corrente-tensão. Neste estágio, o conversor D/A tem a saída de tensão unipolar e um outro amplificador operacional, através do ajuste de tensão de *off set* externo, transforma a tensão unipolar em bipolar, fornecendo na sua saída uma tensão que varia de +9 Volts a -9 Volts.

A tensão de referência para este D/A é gerada a partir da tensão +12 Volts do próprio microcomputador, estabilizando-a em 9,1 Volts através do diodo zener e um amplificador operacional.

O ajuste da tensão de referência é feito por um potenciômetro (P8) e o ajuste do *off set* é feito pelo potenciômetro (P7). O potenciômetro (P9) ajusta o ganho.

A característica desse conversor está mostrada na figura [B.6].

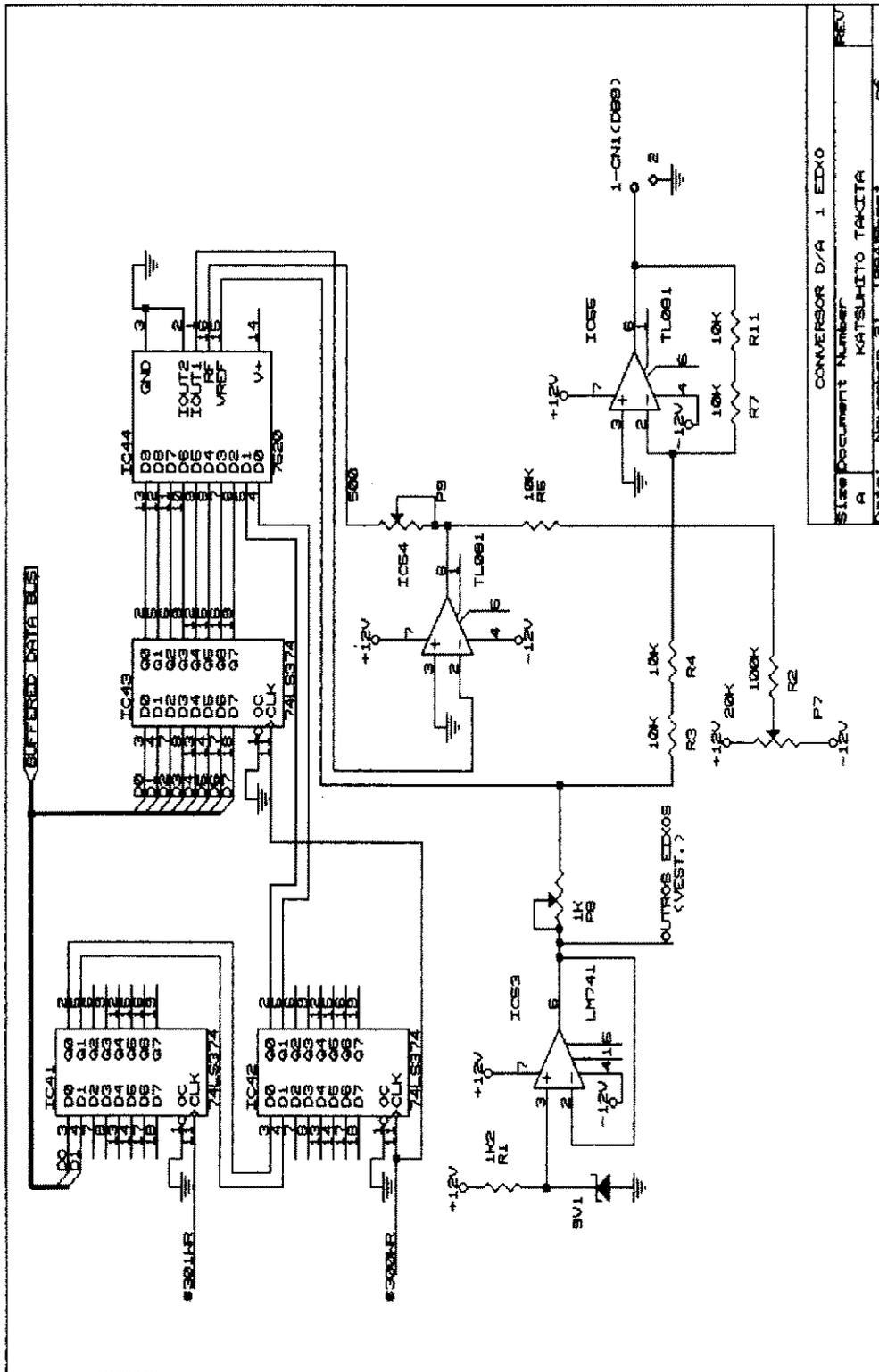


FIG. B.5 CONVERTOR D/A

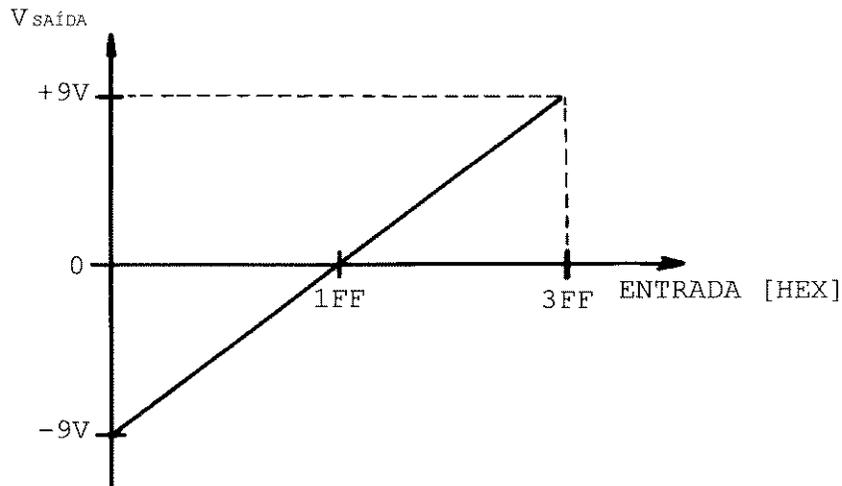


FIG. B.6 CARACTERÍSTICA DO CONVERSOR D/A

#### B.1.4 ENTRADA DIGITAL DE 16 BITS.

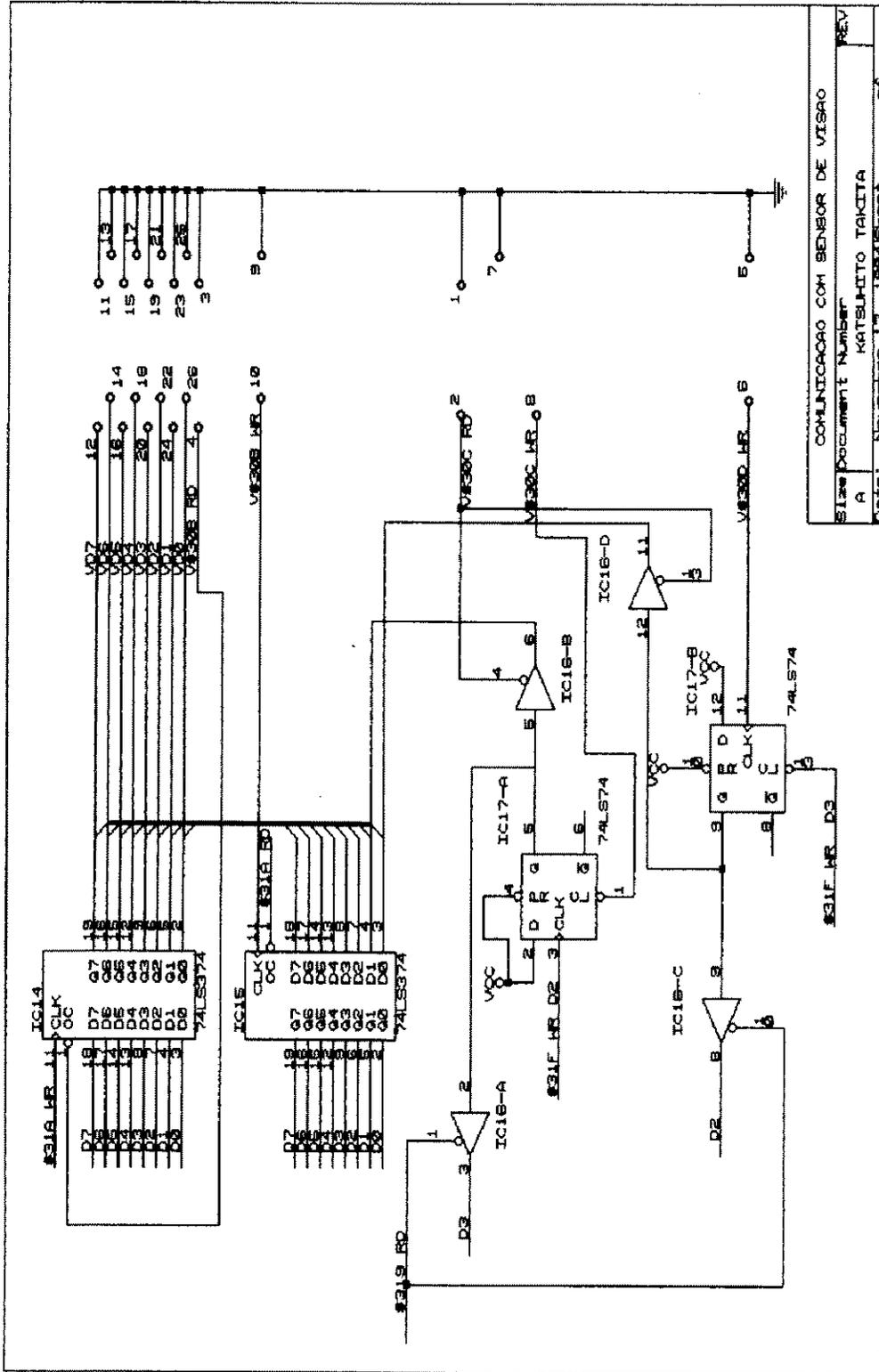
As entradas digitais de 16 bits estão implementadas em cada uma das placas através de dois dispositivos *TRI STATES*, CI 74LS244.

#### B.1.5 CIRCUITO DE COMUNICAÇÃO PARALELA.

O circuito de comunicação paralela com *hand shake* simples é ilustrado na figura [B.7]. Como pode ser observado neste circuito, foram utilizados dois *latches* com *tri-states* para emissão e recepção de dados e dois *flip-flop* tipo D e quatro *tri-states* para estabelecer o *hand shake*, sendo que o *hand shake* é realizado através de *software*.

Para realizar a emissão de uma mensagem é necessário seguir os seguintes passos:

1. Inicialmente deve ser verificado se o *buffer* do receptor está vazio, verificando-se o bit  $D_3$  do endereço \$319RD.



COMUNICACAO COM SENSOR DE VISAO  
 Size Document Number: 74LS74  
 A KATSUHIKO TAKITA  
 Date: November 17, 1994 Sheet 1 of 1

FIG. B.7 CIRCUITO DE COMUNICACAO PARALELA

2. Após constatado que esse bit é zero, escreve-se a mensagem no endereço \$31AWR.
3. Após a escrita dessa mensagem, escreve-se a seqüência 0-1-0 no bit D<sub>2</sub> do endereço \$31FWR para avisar o receptor a existência da mensagem e, ao mesmo tempo, levantar o bit cheio do receptor.

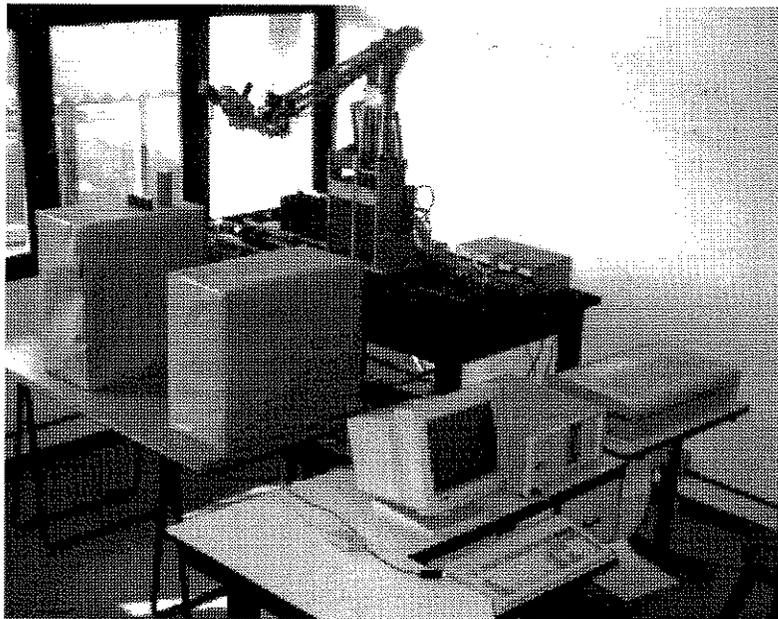
O procedimento para leitura da mensagem é o seguinte:

1. Inicialmente é verificado se existe mensagem no *buffer* de recepção, lendo o bit D<sub>2</sub> do endereço \$319RD.
2. Se o estado desse bit for igual a 1 então realiza-se a leitura do endereço \$31ARD onde se encontra a mensagem.
3. Após a leitura da mensagem, escreve-se 0 no bit D<sub>3</sub> do endereço \$31FWR, para avisar que o *buffer* de recepção está vazio e, ao mesmo tempo, desativa-se o *flag* de EXISTE MENSAGEM.

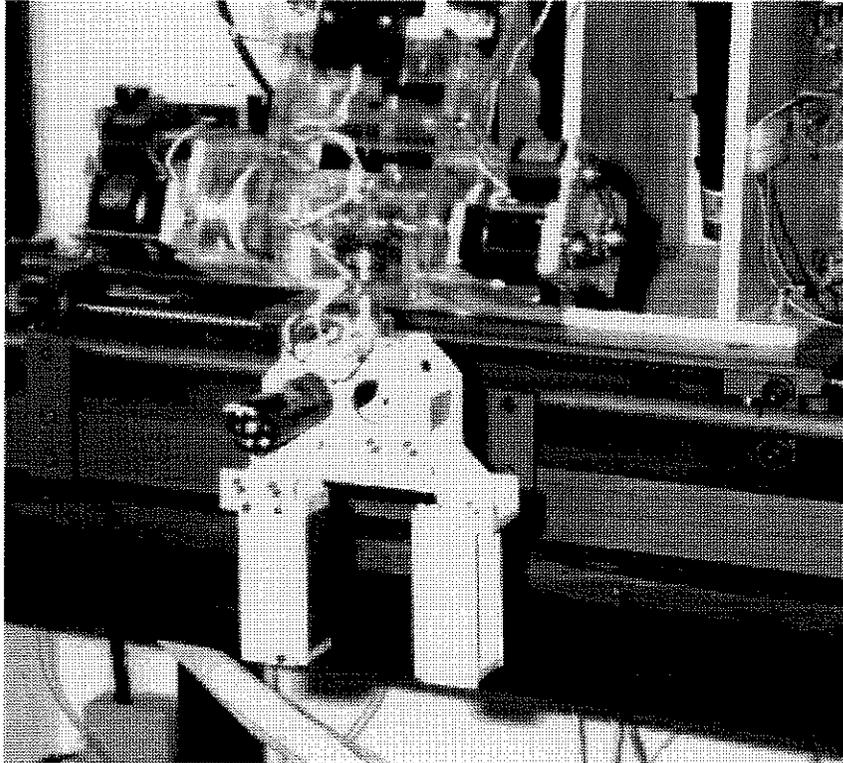
## APÊNDICE C

### FOTOGRAFIAS DO ROBÔ

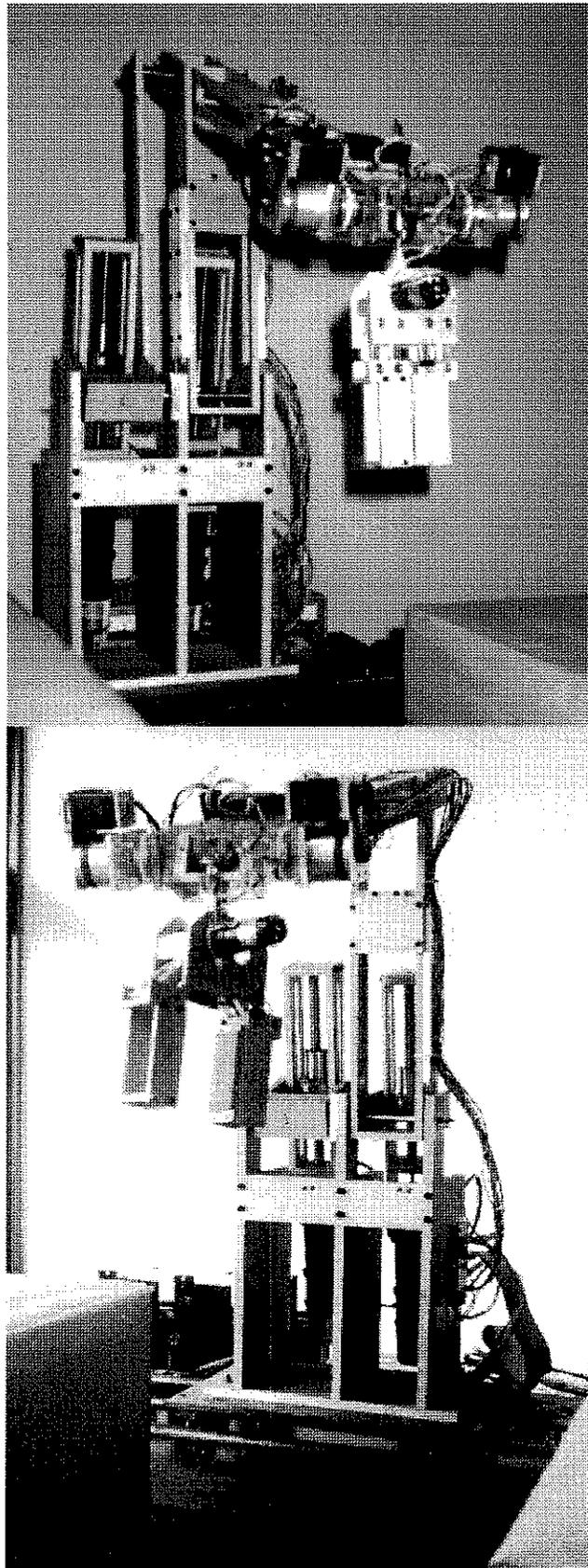
#### C.1 VISÃO GERAL DO SISTEMA DO ROBÔ



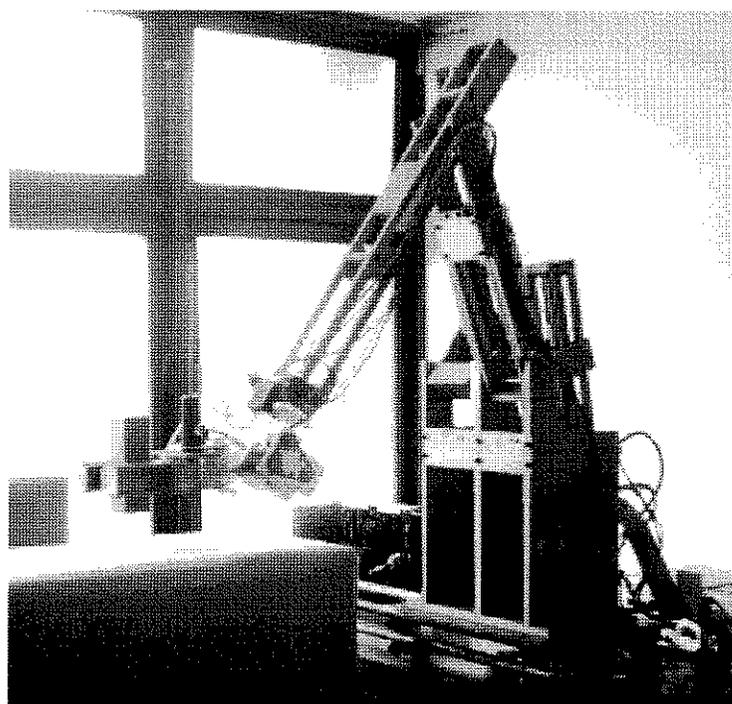
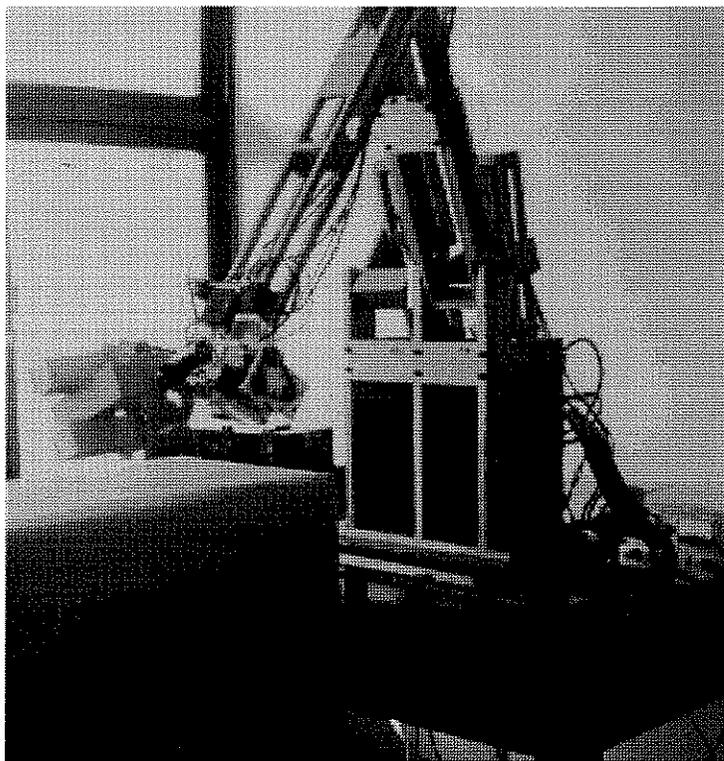
C.2 DETALHES DO PUNHO E DA GARRA.



C.3 VISTA FRONTAL DO ROBÔ



C.4 TRANSPORTE DE OBJETO.



## BIBLIOGRAFIAS

- [01] ARIMOTO, S., MIYAZAKI F.. *Stability and Robustness of PID Feedback Control Law for Manipulators of Sensory Capability*. Robotics Research , MIT Press, Cambridge, MA, 1984.
- [02] ASADA, H., SLOTINE, J.-J.E.. *Robot Analysis and Control*. John Wiley, 1986.
- [03] BRADY, Michael, HOLLERBACH, John M., JOHNSON Timothy L. et al. *Robot Motion: Planning and Control*. MIT Press, Cambridge, MA, 1982.
- [04] BRIGHAM, E.O.. *The Fast Fourier Transform*. Prentice-Hall, 1990.
- [05] CASTRUCCI, Plínio, SALES, Roberto Moura. *Controle Digital*. Edgar Blücher, São Paulo, 1990.
- [06] Engineering Staff of Magnetic Technology. *Motion Control Engineering Handbook*.
- [07] FU, K.S., GONZALEZ, R.C., LEE, C.S.G.. *Robotics: Control, Sensing, Vision and Intelligence*. McGraw-Hill, Singapore, 1987.

- [8] FURUTA, K., KOSUGE, K., MUKAI N.. *Control of Articulated Robot Arm with Sensory Feedback: Laser Beam Tracking System*. IEEE Transactions on Industrial Electronics, vol. 35, No. 1, 1988.
- [09] FURUTA, K., MITSUJI, S.. *Path Control of a Three-Dimensional Linear Motional Mechanical System Using Laser*. Transactions on Industrial Electronics, vol. 35, No.1, 1988.
- [10] GROB, B.. *Televisão e Sistemas de Vídeo*. Guanabara, Rio de Janeiro, 1989.
- [11] HASEGAWA, K., MIZUTANI T.. *On the Autonomous Trajectory Generating Servomechanism for Manipulator Control*. SYSTEMS SCIENCE, vol. 9, No. 3, Wroclaw, Poland, 1983.
- [12] HASEGAWA, K.. *Technological Advance in Robotics and their Automated Manufacturing*. 1º CONAI, São Paulo, 1983.
- [13] HIBBLER, R.C.. *Dinâmica*. Campus, São Paulo, 1986.
- [14] KABUKA, Mansur, DESOTO J., MIRANDA, J.. *Robot Vision Tracking System*. IEEE Transactions on Industrial Electronics, vol. 35, No.1 ,1988.
- [15] KUO, Benjamin C.. *Digital Control Systems*. Second Edition, Saunders College, Orlando, Florida, 1992.

- [16] LATHI, B.P.. *Modern Digital and Analog Communication Systems*. Second Edition, Saunders College, Orlando, Florida, 1989.
- [17] *Linear Data Book*. National Semiconductor Corporation, 1985.
- [18] MOCHIZUKI, J., TAKAHASHI, M., HATA, S.. *Unpositioned Workpieces Handling Robot with Visual and Force Sensors*. IEEE Transactions on Industrial Electronics, vol. IE-34, No.1, 1987.
- [19] MURAKAMI, S.. *Método Simples de Estruturação do Servomecanismo*. Revista Projeto Mecanico. maio, 1985. (em língua japonesa).
- [20] NUSSBAUMER, H. J.. *Fast Fourier Transform and Convolution Algorithms*. Second Edition, Springer-Verlag, Lausanne, Switzerland, 1982. p.80-82.
- [21] OOBA, R.. *Intelligent Sensor Technology*. Ohm, Tokyo, 1990. (em língua japonesa).
- [22] PADUA, Regina Celia T., ALLEN, Robert. *Sensors in Robotics: Review and Applications*. Anais do 6º Congresso Brasileiro de Automática, Belo Horizonte, MG, 1986.
- [23] PAUL, R.P.. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, MA, 1981.

- [24] SATO, S.. *Construção de um Cartão de Interfaceamento para Sinal de Vídeo Composto*. Revista TÉCNICA DE TRANSISTORES, Editora CQ, Tokyo, março, 1989. (em língua japonesa).
- [25] SHIGLEY, Joseph Edward. *Cinemática dos Mecanismos*. Editora da Universidade de São Paulo, São Paulo, 1969.
- [26] SIMÕES, M.A., TAKITA, K., ROHRER, H.G.. *Utilização de Robôs Industriais*. 2º Congresso Nacional de Automação Industrial, São Paulo, 1985.
- [27] SNYDER, Wesley E.. *Industrial Robots: Computer Interfacing and Control*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.
- [28] SPONG, Mark. W., VIDYASAGAR, M.. *Robot Dynamics and Control*. John Wiley, 1989.
- [29] TAKITA, K.. *Desenvolvimento de uma Linguagem Simples de Programação para Robôs Industriais*. XVIII Congresso Nacional de Informática, Rio de Janeiro, 1984.
- [30] TAKITA, K., ALVES, J.B.M., MIZUTANI, T.. *Estrutura Hierárquica de Controle do Robô Industrial*. 2º Congresso Nacional de Automação Industrial, São Paulo, 1985.
- [31] TAKITA, K.. *A Study of Programming Method for a Robot*. 11º IMACS World Congress, Oslo, Noruega, 1985.

- [32] TAKITA, K., VIEIRA, A.L., PALHARES A.G.B.. *Applications of the Autonomous Trajectory Generating Technique on a TRR Robot Mechanism*. World Congress on Telecommunication and Control, Rio de Janeiro, 1985.
- [33] TAKITA, K., VIEIRA, A. L., PALHARES, A.G.B.. *Geração Autônoma de Trajetória Contínua Espacial Aplicada a um Robô TRR*. 2º Congresso Latinoamericano de Control Automatico, Buenos Aires, 1986.
- [34] TAKITA, K.. *Um método de Representação e Controle de Sistema Discreto - Mark Flow Graph*. I Encontro Regional Norte-Nordeste de Automática, Belém, Pará, 1987.
- [35] TAKITA, K.. *Uma Proposta de Terminologia no Setor de Robótica no Brasil*. Simpósio sobre Robótica, SOBRACON, São Paulo, 1987.
- [36] TAKITA, K., VIEIRA, A., PALHARES, A.G.B.. *Desenvolvimento de uma Garra com Sensores para Manuseio de Peças Frágeis*. 9º Congresso Brasileiro de Automática, Vitória, ES, 1992.
- [37] TIPPLER, Paul A.. *Física*. Segunda Edição, Guanabara Dois, Rio de Janeiro, 1984.
- [38] TODD, D. J.. *Fundamentals of Robot Technology*. Halsted Press, 1986.
- [39] *TTL Applications Manual*. Texas Instruments Electronics Series, 1981.

- [40] *TTL Handbook*. Editora CQ, Tokyo, 1990. (em língua japonesa).
- [41] VIEIRA, A.L., TAKITA, K., PALHARES A.G.B.. *O Efeito de Controle de Velocidade das Juntas no Desempenho da Geração de Trajetória de um Braço Mecânico*. 6º Congresso Brasileiro de Automática. Belo Horizonte, MG, 1986.
- [42] VIEIRA, A.L.. TAKITA, K., PALHARES, A.G.B.. *On the Autonomous Tridimensional Trajectory Generating Thecnique for Articulated Robot Arm*. 3rd International Workshop on Advanced Motion Control, University of California at Berkeley, Berkley, 1994.
- [43] VIEIRA, A.L.. *Relatório Parcial para Tese de Doutorado*. DSCE - FEE - UNICAMP , Campinas , SP, 1995.