

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA  
TESE DEFENDIDA POR Cássio Hervella

E APROVADA PELA  
COMISSÃO JULGADORA EM 12/05/95

João Maurício Rosário  
ORIENTADOR

**UNIVERSIDADE ESTADUAL DE CAMPINAS**  
**FACULDADE DE ENGENHARIA MECÂNICA**

# **Projeto e Desenvolvimento de um Controlador Lógico Programável Flexível para Controle de Manipuladores e Robôs Industriais**

Autor : **Cássio Hervella**

Orientador : **João Maurício Rosário**

H445p

31716/BC

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

# **Projeto e Desenvolvimento de um Controlador Lógico Programável Flexível para Controle de Manipuladores e Robôs Industriais**

**Autor : Cássio Hervella**

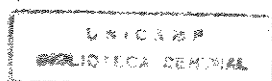
**Orientador : João Maurício Rosário** \*

**Curso: Engenharia Mecânica**

**Área de Concentração: Projeto Mecânico**

Dissertação de mestrado apresentado à comissão de Pós Graduação da Faculdade de Engenharia Mecânica da UNICAMP, como requisito para a obtenção do Título de Mestre em Engenharia Mecânica.

Campinas , 1996  
S.P. - Brasil



9311849

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

# **Projeto e Desenvolvimento de um Controlador Lógico Programável Flexível para Controle de Manipuladores e Robôs Industriais**

**Autor : Cássio Hervella**

**Orientador : João Maurício Rosário**

**Curso: Engenharia Mecânica**

**Área de Concentração: Projeto Mecânico**

Dissertação de mestrado apresentado à comissão de Pós Graduação da Faculdade de Engenharia Mecânica da UNICAMP, como requisito para a obtenção do Título de Mestre em Engenharia Mecânica.

**Campinas , 1996  
S.P. - Brasil**

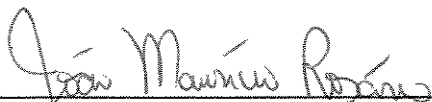
**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO**

**DISSERTAÇÃO DE MESTRADO**

# **Projeto e Desenvolvimento de um Controlador Lógico Programável Flexível para Controle de Manipuladores e Robôs Industriais**

**Autor : Cássio Hervella**

**Orientador : Prof. Dr. João Maurício Rosário**



---

**Prof. Dr. João Maurício Rosário, Presidente**  
UNIVERSIDADE ESTADUAL DE CAMPINAS-UNICAMP



---

**Prof. Dr. Hans Ingo Weber**  
UNIVERSIDADE ESTADUAL DE CAMPINAS-UNICAMP



---

**Prof. Dr. Luiz Carlos Sandoval Góes**  
INSTITUTO TECNOLÓGICO DA AERONÁUTICA-ITA

Campinas, 12 de Maio de 1995

## **Dedicatória:**

Dedico este trabalho a meus pais, como prova de eterna gratidão pelo apoio e incentivo recebidos ao longo de toda minha vida.

## **Agradecimentos**

Ao Professor e amigo João Maurício Rosário, pela orientação, pela confiança e pelo apoio no desenvolvimento deste trabalho.

A Meus Familiares, pelo encorajamento ao desenvolvimento acadêmico.

À Mônica, pelas palavras de apoio e encorajamento nos momentos mais difíceis deste trabalho.

À Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP, pelo apoio financeiro, fundamental no desenvolvimento deste projeto de pesquisa.

À UNICAMP, pela oportunidade de uma formação acadêmica de nível internacional.

Ao Departamento de Projeto Mecânico da Faculdade de Engenharia de Campinas e ao Centro de Tecnologia da UNICAMP, pela infra-estrutura e recursos materiais utilizados neste trabalho.

A Todas as pessoas que, de uma forma ou de outra, contribuíram para a realização deste trabalho.

# Sumário

<b>Introdução</b>	<b>1</b>
<b>1 - Descrição do Sistema de Controle Flexível</b>	<b>5</b>
1.1 - Descrição do Sistema	5
1.2 - Posicionamento do Problema	11
1.3 - Conclusões	12
<b>2 - Módulo de Monitoramento e Supervisão</b>	<b>13</b>
2.1 - Encoder Ótico Incremental	14
2.2 - Interface Decodificadora de Sinais de Encoder	16
2.3 - Interface de Contadores	18
2.4 - Interface de Acesso Multiplexado	21
2.5 - Interface Digital	25

2.6 - Software para Aquisição, Monitoramento, Supervisão e Inicialização	28
2.7 - Conclusões	29
<b>3 - Módulo de Controle e Interfaces de Acionamento</b>	<b>30</b>
3.1 - Interface Geradora de PWM	31
3.1.1 - Descrição da Interface	32
3.1.2 - Software para Geração de Dados da Memória do Gerador de PWM	37
3.2 - Interfaces de Acionamento de Potência Chaveada	39
3.2.1 - Descrição das Interfaces	40
3.3 - Fontes de Alimentação DC e Funções Auxiliares	43
3.4 - Conclusões	43
<b>4 - Módulo de Interface com o Usuário</b>	<b>44</b>
4.1 - Interface com o Usuário	45
4.2 - Módulo de Programação e Simulação	50
4.3 - Conclusões	51
<b>5 - Testes e Resultados Experimentais</b>	<b>54</b>



5.1 - Testes e Validações de Hardware	55
5.2 - Testes e Validações de Software	57
5.3 - Controle de Posicionamento de uma Câmera	58
5.4 - Controle de um Robô Industrial	59
5.5 - Conclusões	60
<b>6 - Conclusões e Perspectivas</b>	<b>61</b>
<b>Anexos</b>	<b>66</b>
A - Modelamento Cinemático de Robôs	67
B - Diagramas e Lay-Outs dos Hardwares Desenvolvidos	84
<b>Referências Bibliográficas</b>	<b>105</b>

## Resumo

HERVELLA, C, *Projeto e Desenvolvimento de um Controlador Programável Flexível para Manipuladores e Robôs Industriais*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1996. 73 p. Dissertação (Mestrado)

Neste trabalho foi projetado, desenvolvido e implementado um sistema de controle programável flexível, para aplicação principal no controle de manipuladores e robôs industriais.

Como método de abordagem global de um problema de controle automático, podemos dividi-lo em quatro partes principais, a primeira sendo a medição, aquisição e tratamento das grandezas a controlar, a segunda sendo o algoritmo de controle e o sistema de acionamento, a terceira sendo a automatização via programação e simulação de trajetórias e tarefas, e finalmente, a quarta parte sendo a interface homem-máquina.

Sendo assim, subdividiu-se o sistema de controle desenvolvido em quatro módulos principais, sendo eles, o módulo de Monitoramento e Supervisão, o módulo de Controle e Acionamento, o módulo de Programação e Simulação e o módulo de Interface com o Usuário; por sua vez, estes módulos são compostos por vários sub-sistemas modulares, que foram projetados, desenvolvidos e testados separadamente, para que fossem flexíveis e confiáveis,

tendo sido integrados em seguida, de várias maneiras, de modo a atender às características específicas de cada aplicação.

Dentre os sistemas desenvolvidos, temos, por exemplo, interfaces de interface de entrada e saída de dados digitais, interfaces de supervisão de posição, interfaces de acionamento de potência, interfaces de decodificação de sinais de encoders; programas de testes para as interfaces desenvolvidas, programa de interface gráfica para utilização pelo usuário, e outros.

Numa aplicação típica, no controle de um robô industrial, podemos ter diversos tipos de atuadores, sensores, mecanismos e arquiteturas do sistema mecânico a controlar, assim, com a substituição de alguns módulos, obtêm-se uma integração flexível que atende as necessidades de cada caso, sem que seja necessário um sistema de controle completamente novo, tornando a automação mais simples, eficiente e rápida.

Por exemplo, na automação de um robô, podem ser necessários dois computadores, para que as cargas de processamento de dados de controle e a carga de processamento da interface gráfica com o usuário possam ser divididas entre estes; porém, numa aplicação, como o posicionamento de uma câmera de vídeo, a simplicidade do sistema mecânico, permite a utilização de um único computador, ou mesmo, uma interface dedicada, sendo mantida toda a estrutura global do sistema de controle.

#### *Palavras Chave*

-Sistemas de Controle Flexíveis Programáveis, Monitoramento., Automação Industrial, Robótica

## Abstract

HERVELLA, C, *Projeto e Desenvolvimento de um Controlador Programável Flexível para Manipuladores e Robôs Industriais*, Campinas,: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1996. 73 p. Dissertação (Mestrado)

In this work, it has been designed, developed and implemented a programmable flexible control system, for main application in the control of manipulators and industrial robots.

As a global approximating method for an automatic control problem, we can divide it in four main parts, the first being the measurement, acquisition and treatment of the variables to control, the second being the control algorithm and the actioning system, the third being the automation through programming and simulating of jobs and trajectories, and finally, the fourth being the man-machine interface.

Thus, we divided the developed control system in four main modules, which are, the Monitoring and Supervising module, the Actioning and Control module, the Simulation and Programming module and the User's Interface; which are compound by various modular subsystems, that were designed, developed and tested separately, so that they would be dependable and flexible, being then integrated, in many ways, to full-fill the specific characteristics of each application.

Within the developed systems, we have, for example, digital data input and output interface boards, position supervising boards, power driving boards, encoder signal decoding boards; test programs for the developed boards, graphics interface program to be utilized by the user, and others.

In a typical application, in controlling an industrial robot, we can have many different types of actuators, sensors, mechanisms and architectures of the mechanical system to control, thus, by substituting a few modules we can obtain a flexible integration that answers to the needs of each case, without the need of a completely new control system, making the automation more simple, efficient and fast.

As an example, in the automation of a robot, may be necessary two computers, so that the data processing loads of control and the user's graphics interface processing load can be divided between these, but, in an application, like a video camera positioning, the simplicity of the mechanical system, enables the use of only one computer, or even, a dedicated board, keeping the global structure of the control system.

### *Key Words*

Programmable Flexible Control System, Monitoring., Industrial Automation, Robotics

## **Lista de Figuras**

Figura 1.1 - Estrutura Global de um Sistema Automático de Controle.	6
Figura 1.2 - Estrutura Simplificada do Sistema Automático de Controle.	7
Figura 1.3 - Estrutura e Implementação do Sistema Automático de Controle.	8
Figura 1.4 - Sistema de Controle Aplicado num Robô de Seis Graus de Liberdade	10
Figura 3.1 - Tela Gráfica do Programa Gerador de Formas de Onda para PWM	39
Figura 4.1 - Estrutura da Interface com o Usuário no Sistema Automático	46
Figura 4.2 - Tela Gráfica da Interface com o Usuário, com o Gerador de Aplicativos	49
Figura 5.1 - Sistema de Controle de Posicionamento de uma Câmera	58

**Lista de Tabelas**

Tabela 2.1 - Configuração de Endereços da Interface Digital.	27
Tabela 3.1 - Configuração do Tipo de PWM Gerado.	35

## Introdução

Os objetivos deste trabalho foram o estudo, o projeto, a implementação e os testes de um controlador programável flexível para aplicação principal no controle de manipuladores e robôs industriais, com uma arquitetura que possibilite a implementação dos diversos tipos de algoritmos de controle que se deseje aplicar e/ou desenvolver e testar; tais como, controladores lineares, não lineares, de torque calculado, adaptativos e outros, de acordo com as características específicas de cada aplicação.

Visou-se, ainda, obter uma estrutura de controle flexível, que possibilite a implementação e testes de malhas de controle mais complexas e eficientes para manipuladores e robôs industriais, procurando, principalmente, soluções de menor custo, de menor complexidade e de maior desempenho, através do uso de modernos microprocessadores, cujas altas capacidades de processamento, possibilitam a redução dos tempos de controle e viabilizam a implementação de algoritmos de controle mais complexos e eficientes.

Na Faculdade de Engenharia Mecânica da UNICAMP, um grupo de pesquisadores atua no domínio da Robótica e Automação Industrial, concentrando-se nas áreas de Cinemática, Dinâmica e Controle de Manipuladores e Robôs Industriais, sendo que, durante um projeto de pesquisa financiado pela FAPESP [35], no período de 1990 a 1991, foi realizado um projeto, onde foi projetado e construído um sistema para aquisição e monitoramento de sinais de posição de sistemas mecânicos via computador.

Dando continuidade a este, e aos trabalhos de pesquisa do grupo na área, seguindo a linha do estudo e da proposição de novas soluções para os problemas de controle de sistemas mecânicos, e baseados no suporte industrial existente; prosseguindo no desenvolvimento de



ferramentas e soluções para estes problemas, tendo como prioridade o problema de manipuladores e robôs industriais.

Nessa linha de pesquisa, nos deparamos com o problema dos controladores digitais existentes, que têm pouca flexibilidade, com relação aos algoritmos que podem implementar, e que têm baixo desempenho, quando em se tratando das necessidades dos sistemas robóticos, visto que, na maioria desses sistemas, os controladores convencionais evitam os problemas mais difíceis, limitando a capacidade do mecanismo, durante a sua operação, à uma fração do possível, e conseqüentemente, sub utilizando os recursos dos sistemas de acionamento dos mecanismos dos robôs e dos manipuladores, em suas diferentes formas (acionamento elétrico, hidráulico, pneumático ou outros).

Tendo em vista esta relação custo / benefício desfavorável dos sistemas robóticos existentes, em função da sub utilização do equipamento e do altíssimo custo dos controladores dedicados, destes manipuladores e robôs, percebemos a necessidade de novas soluções, como a que pesquisamos neste trabalho.

A implementação do controlador desenvolvido tem por finalidade, realizar os cálculos dos diferentes tipos de algoritmos de controle que se pretende implementar e estudar, devendo ter, para isso, a flexibilidade de programação necessária; num contexto mais amplo, o objetivo deste trabalho seria prover recursos para o estudo do comportamento dos diferentes tipos de algoritmos de controle, quando implementados em sistemas mecânicos reais, objetivando a otimização da solução do problema de controle de manipuladores e robôs industriais, através da obtenção de um controlador programável flexível que tenha um baixo custo e um alto desempenho, e que torne possível que as velocidades de operação dos robôs sejam as máximas possíveis suportadas por seus componentes mecânicos, fato que não é conseguido atualmente, pois as velocidades de operação são limitadas pelos controladores e seus algoritmos.

Neste trabalho, constituído de seis capítulos, apresentamos uma visão geral do problema de controle de robôs e manipuladores, juntamente com uma abordagem para sua solução; sendo que, no primeiro capítulo descrevemos o problema de controle de robôs e manipuladores industriais; mostrando a necessidade de um controlador mais complexo, flexível e eficiente, a partir dos problemas de modelamento Cinemático e dinâmico, e de controle de seus mecanismos, além dos problemas de custo e flexibilidade dos controladores convencionais. Descrevemos, ainda, de forma global, a implementação adotada para o controlador, mostrando os módulos que compõem o sistema de controle programável flexível.

No segundo capítulo, partimos para a descrição dos componentes do sistema de controle desenvolvido, mostrando a idéia do módulo de monitoramento, aquisição e supervisão e de seus subsistemas; os encoders óticos incrementais e as interfaces desenvolvidas ; decodificadoras de sinais de encoder, de contadores digitais, de acesso multiplexado, de interface de entradas e saídas digitais e os programas computacionais desenvolvidos para monitoramento e supervisão.

No terceiro capítulo, descrevemos o módulo de acionamento e controle, juntamente com a arquitetura de um algoritmo para controle de robôs; e descrevemos seus subsistemas, as interfaces geradoras de PWM (Pulse-Width-Modulation), os programas para programação das formas de onda desejadas destas interfaces, as interfaces de acionamento de potência chaveadas e as fontes de alimentação CC e funções auxiliares.

No quarto capítulo, é apresentada a estrutura do módulo de interface com o usuário justificando, a necessidade de uma interface com o usuário de alta produtividade e qualidade, uma maior integração com o ambiente de controle e com o ambiente de dados da produção; e descrevemos o ambiente multitarefa e a interface gráfica desenvolvidos. Apresentamos, ainda, a estrutura proposta para o módulo de programação e simulação e de seus subsistemas, os programas computacionais para programação em linguagem de alto nível, para simulação de

trajetórias e tarefas, para desenho plano e espacial de trajetórias, tarefas, objetos, para desenho dos ambientes de trabalho e do próprio mecanismo do robô.

No quinto capítulo, descrevemos os testes de validação e operação, os resultados obtidos e algumas aplicações possíveis do sistema de controle; e, finalmente, mostramos as conclusões e perspectivas para o sistema de controle proposto; fornecendo, ainda, nos anexos, um resumo da teoria de modelamento e controle de robôs utilizada, os principais diagramas elétricos das interfaces construídas e os seus respectivos lay-outs.

## **Capítulo 1**

### **Descrição do Sistema de Controle Flexível**

#### **1.1 - Descrição do Sistema**

O sistema de controle programável flexível deve ter uma estrutura global genérica, que possibilite a sua adequação aos vários tipos de mecanismos robóticos que se deseja automatizar, tanto do ponto de vista de programação e de controle, quanto do ponto de vista dos sensores e acionadores com os quais este irá se ligar.

Assim, uma abordagem global da estrutura modular do sistema automático de controle pode ser vista na figura I.1, sendo que qualquer sistema pode ser subdividido como mostra a estrutura, mesmo que alguns dos módulos sejam de pouca importância ou mesmo inexistentes em algumas aplicações.

Os módulos de programação e simulação, não fazem parte do escopo de desenvolvimento deste trabalho, sendo abordados apenas de forma estrutural e funcional, visando seu futuro desenvolvimento e integração com o restante do sistema; sendo assim, podemos simplificar a estrutura do sistema desenvolvido, aprofundando-nos na sua implementação física, como mostra a figura 1.2.

Devemos, então, selecionar o método de implementação de cada módulo, se em hardware ou em software; sabemos, que um sistema de controle totalmente implementado em software seria inicialmente desejável, uma vez que o projeto, a construção e a manutenção de software são mais rápidos e acarretam menores custos.

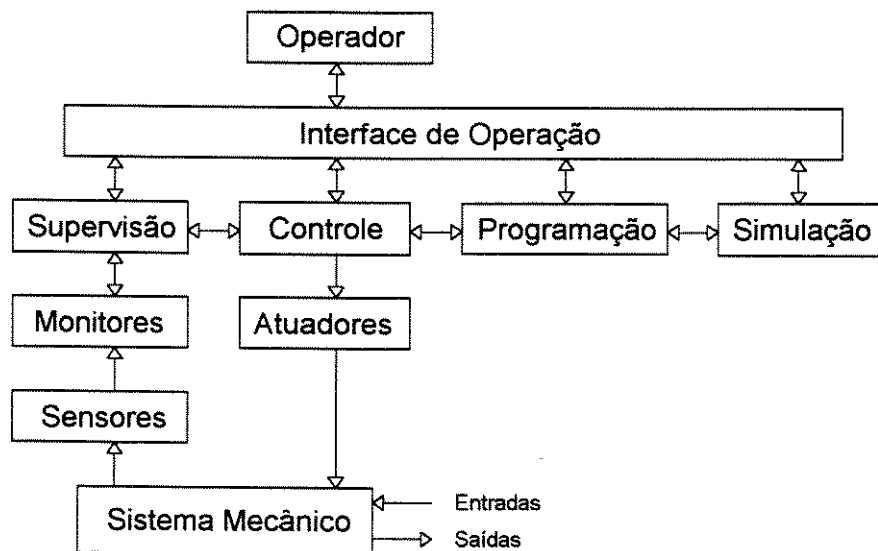


Figura 1.1 - Estrutura Global de um Sistema Automático de Controle.

Porém, a natureza física dos módulos de sensores e acionamento inviabilizam a sua implementação em software, tendo sido implementados, por este motivo, em hardwares específicos; além disso, optou-se pela implementação do módulo de monitoramento em hardware, devido à grande velocidade de processamento que este requer ao tratar os dados dos sensores; sendo que, os módulos restantes foram implementados em software, devido à sua natureza essencialmente lógica e não necessariamente física.

Visando a máxima flexibilidade do sistema global, a implementação, não flexível, dos módulos de sensores, monitoramento e acionamento, em hardwares específicos, deve ser compensada por uma capacidade de seus subsistemas de serem substituídos e/ou configurados

de acordo com as características específicas de cada mecanismo a controlar, mantendo-se um padrão de ligação destes com o computador, através de uma interface comum de comunicação para todos os hardwares específicos.

Esta interface digital, como mostra a figura 1.3, é responsável pela padronização e pela flexibilidade do sistema na parte que é implementada em hardwares específicos, fazendo a conexão entre estes e os módulos implementados em software.

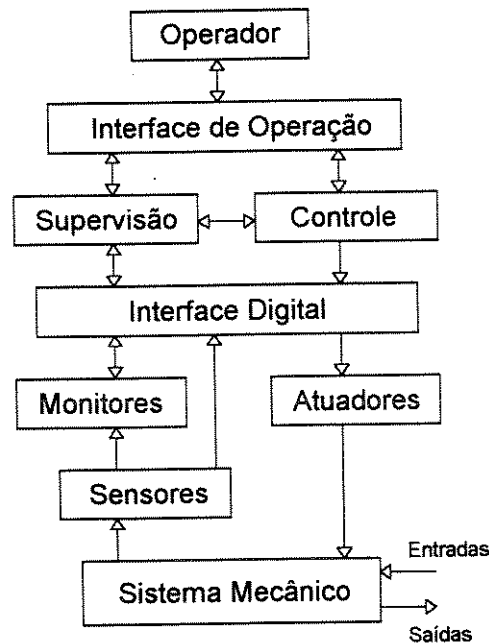


Figura 1.2 - Estrutura Simplificada do Sistema Automático de Controle.

Nestas bibliotecas, devem estar implementados os procedimentos específicos para acesso aos hardwares conectados externamente, sendo que, seus endereços, ou seja, suas posições físicas de ligação, devem poder ser fornecidas pelo usuário que configurou o sistema.

Em forma de bibliotecas, devem estar ainda, os procedimentos, modelos matemáticos e algoritmos específicos para supervisão e controle de um mecanismo em particular, bem como,

os modelos geométricos espaciais de peças e do meio ambiente, para que se possa programar e simular o funcionamento do mecanismo, com realismo suficiente, para uma programação e operação segura e eficiente do mecanismo.

Como exemplo, mostramos a aplicação do sistema de controle num robô de seis graus de liberdade, como mostra a figura 1.4, nesta aplicação, consideramos as seis juntas movidas por motores DC acionados por interfaces de potência chaveadas por PWM's, além disso, em cada junta, temos um encoder ótico incremental, cujos sinais vão para decodificadores, onde através de contadores, é realizado o monitoramento das posições das juntas, em sendo as mesmas, acessadas pelo software de supervisão, através das interfaces de interface e multiplexadoras, existentes nesta configuração do sistema de controle.

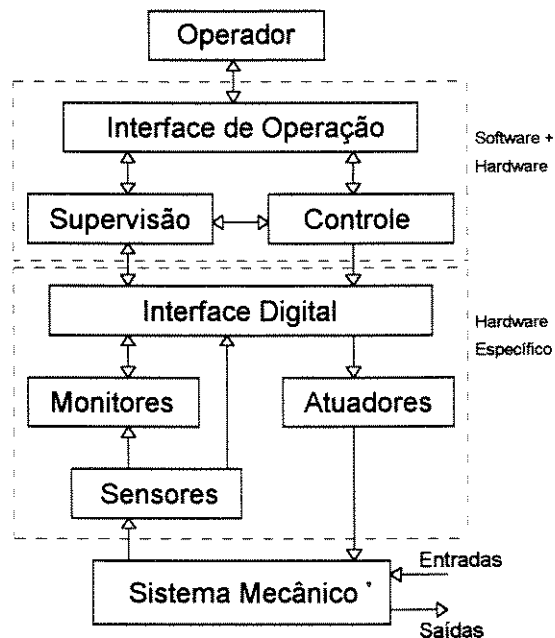


Figura 1.3 - Estrutura e Implementação do Sistema Automático de Controle.

Em software, então, o módulo de supervisão converte estas posições das juntas em posições espaciais do elemento terminal do robô, visto que o modelo geométrico do mecanismo é totalmente conhecido, fornecendo estes dados à interface de operação para

monitoramento pelo usuário e subsídios para o módulo de controle, para que este aplique o algoritmo de controle escolhido durante a execução de tarefas, comandando, assim, os atuadores das juntas.

Através da interface de operação, são fornecidas as informações a respeito de trajetórias e tarefas de posicionamento programadas pelo operador do sistema, obtidas a partir de programação on-line, ou, de bibliotecas de tarefas programadas off-line; além disso, a interface de operação fornece ao usuário os dados do sistema, vindos do módulo de supervisão, para sua análise.

A interface de operação é o centro de gerenciamento do sistema de controle, coordenando o funcionamento de todos os módulos implementados em software, podendo ser configurado de acordo com as características do mecanismo controlado.

Neste exemplo, temos os hardwares específicos de acionamento, sensores e monitoramento propostos para o mecanismo do robô, e conectados ao computador via interfaces digitais padrão.

Neste caso, devido ao número de graus liberdade que se deseja controlar, o número de portas de entrada e saída digitais deve ser fornecido por três interfaces digitais, o que não altera a programação das bibliotecas de software, bastando fornecer os endereços de acesso destas interfaces aos respectivos módulos.

Além disso, neste exemplo, devido a carga de processamento que um robô de seis graus de liberdade requer, devemos dividir os módulos implementados em software em dois computadores, assim, concluí-se pela implementação dos módulos de supervisão e controle em um microcomputador, conectado e comunicando-se com outro microcomputador,



responsável pelos módulos de interface com o usuário, e outras tarefas do sistema, externas ao controlador, ambos providos de interfaces de comunicação de rede de alta velocidade.

Devido ao método de implementação escolhido, esta divisão fica a cargo dos módulos de interface de operação, presentes em ambos os computadores, configurados convenientemente, comunicando-se através das interfaces de rede e de bibliotecas padrões de procedimentos de comunicação.

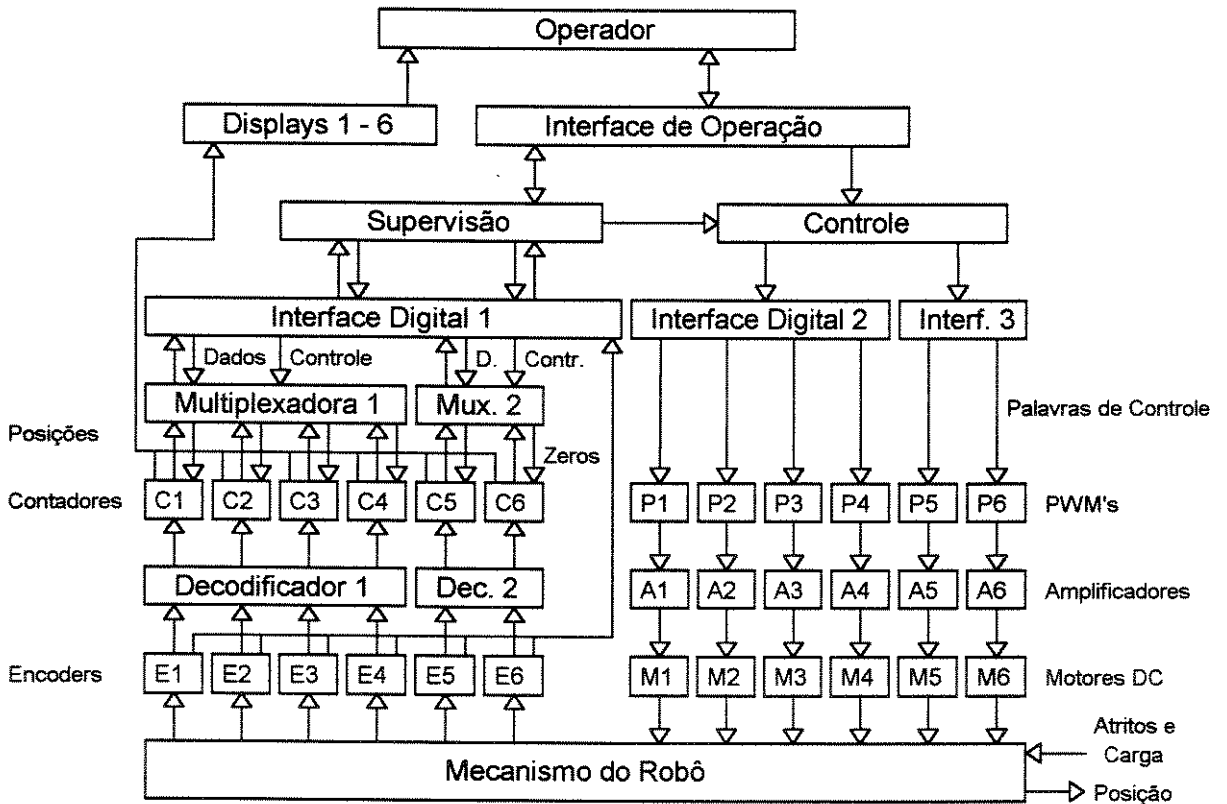


Figura 1.4 - Sistema de Controle Aplicado num Robô de Seis Graus de Liberdade

Esta metodologia torna a divisão de tarefas por vários computadores numa rede fácil para o usuário, uma vez que a comunicação é feita automática e transparentemente pelo

módulo de interface de operação; outra vantagem deste método é a possibilidade de operação e programação do robô a distância, numa central de gerenciamento fora do chão de fábrica; além disso, é possível, ainda, que vários robôs sejam ligados em rede, com uma supervisão centralizada de uma linha de produção, por exemplo.

## 1.2 - Posicionamento do Problema

Analisando-se este exemplo típico de um robô industrial, concluímos, através de um modelamento Cinemático e dinâmico do mecanismo [Anexo A], que o processamento matemático de um algoritmo de controle simples (PID, por exemplo) para este robô, tomaria muito tempo, provocando um aumento no período de amostragem da malha de controle; sendo que, os controladores convencionais superam este problema simplificando o algoritmo de controle (linearizando-o, truncando-o ou até mesmo substituindo-o por um simples PID), ou aumentando a complexidade do hardware computacional (utilizando normalmente processamento paralelo, implementado através de arquiteturas proprietárias específicas para aplicação em robôs).

No primeiro caso, a simplificação do algoritmo de controle implica numa pior resposta dinâmica do mecanismo, e deve ser acompanhada de uma redução nas velocidades e acelerações dos atuadores e/ou de uma redução na carga máxima que este pode movimentar, reduzindo a possibilidade de saturação nos atuadores; isto para que o sistema tenha uma margem de segurança aceitável no que se refere a sua estabilidade.

No segundo caso, aumentar a complexidade do hardware soluciona o problema de tempos de processamento, porém, implica num custo muito elevado, visto que estes controladores são computadores desenvolvidos somente e especificamente para o controle de robôs, não sendo aplicáveis em outros equipamentos de uso comum, implicando ainda, num custo elevado, também, devido a sua escala de produção ser reduzida.

Deve-se salientar, ainda que, ambas as soluções, para a complexidade dos algoritmos de controle de robôs, não possibilitam boas relações custo/benefício, e além disso, são soluções fechadas que não permitem alterações ou desenvolvimento posterior, sendo que na maioria dos casos, estes controladores não permitiriam a conexão de interfaces homem-máquina e outros sistemas acessórios ou de gerenciamento.

No sistema desenvolvido, estas restrições não existem, pois, a implementação do controlador utiliza computadores padrões (com microprocessadores da família 80x86 na arquitetura dos computadores pessoais), de aplicação comercial e custo reduzido; cujas capacidades de processamento são tão elevadas quantas as capacidades dos controladores específicos convencionais, com a vantagem de ser possível a divisão de tarefas e a conexão com outros sistemas via interligação destes computadores em rede; ou seja, um sistema aberto e de baixo custo relativo.

### **1.3 - Conclusões**

Neste capítulo, foi apresentada uma visão global do sistema de controle proposto e sua estrutura, a abordagem teórica utilizada na sua obtenção, a metodologia utilizada na sua implementação, e as possibilidades de sua aplicação.

Além disso, foram analisadas as motivações teóricas e práticas que levaram ao desenvolvimento da estrutura flexível obtida.

Em seguida, descrevemos os módulos que compõem a estrutura de controle flexível desenvolvida.

## **Capítulo 2**

### **Módulo de Monitoramento e Supervisão**

O módulo de sensores, monitoramento e supervisão é responsável, do ponto de vista global de controle, pela realimentação das variáveis observadas dos sistemas mecânicos, devendo fornecer os valores destas e de outras variáveis, dependentes destas, para o módulo de controle e para o usuário, para processamento, e eventuais análise e/ou acompanhamento.

Sendo assim, as informações colhidas sobre o mecanismo devem seguir um caminho similar, qualquer que seja a grandeza física medida, caminho este, iniciado nos sensores, onde normalmente são convertidos de suas formas físicas originais para uma forma elétrica mais comum (como, por exemplo, sinais elétricos digitais ou analógicos com faixas de valores padronizadas), facilitando seu posterior tratamento.

Em seguida, estes sinais podem ou não ser tratados num módulo de monitoramento, dependendo da necessidade e do tipo de sinal fornecido pelos sensores, sendo que neste módulo, quando presente, os sinais sofrem uma análise e/ou um tratamento prévios, visando a sua adequação a interface padrão do computador, e/ou visando um diagnóstico do sensor e/ou dos dados adquiridos, e/ou, ainda, visando um tratamento em tempo real dos dados (necessidade comum em sensores cujos sinais têm frequências altas quando comparadas com a frequência da malha de controle implementada em software, sendo um exemplo típico, os encoders óticos).

Finalmente, no módulo de supervisão, é feito o processamento dos dados adquiridos, visando a sua análise em termos de faixas de segurança, de cálculos de variáveis de estado, modelos geométricos de robôs, e etc...

Neste capítulo, descrevemos os subsistemas deste módulo, desenvolvidos para a implementação do sistema de controle flexível.

## **2.1 - Encoder Ótico Incremental**

Um encoder ótico incremental é um sensor que fornece dados relativos a posição e deslocamento angular de um eixo, através de pulsos elétricos, a medida que este eixo, ao qual o encoder se encontra acoplado, gira.

Foi projetado e testado um encoder ótico incremental, baseado nos tipos convencionais, porém, através de uma implementação de baixo peso final do encoder, visando-se a aplicação em robôs e manipuladores, onde a redução do peso dos sensores é um fator fundamental no aumento da capacidade de carga do mecanismo, com conseqüente melhoria na relação custo/benefício do sistema.

Através desta implementação buscou-se, ainda, uma precisão relativamente alta, a um baixo custo, tendo esta implementação, sido motivada pelas características desfavoráveis de peso, custo e precisão dos vários sensores disponíveis no mercado.

Um encoder ótico incremental, fornece, num formato codificado, a informação de variação de posição angular discretizada deste eixo, este diferencial de posição deve ser integrado para a obtenção do deslocamento total, e/ou deve ser relacionado à uma base de tempo para a obtenção da velocidade angular.

Devido a esta natureza incremental, a posição angular do encoder só pode ser conhecida através da contagem (integração discreta) do número de estados fornecidos pelas pistas defasadas (diferenciais de posição), somados num sentido de rotação e subtraídos no outro, sempre, com relação a algum ponto de referência (um valor de inicialização).

O encoder projetado, fornece uma precisão de 1024 partes por revolução de seu eixo, através de duas pistas de 256 pulsos por revolução, defasadas de  $90^\circ$  elétricos ( $1/4$  de pulso), resultando em uma precisão efetiva de  $360^\circ/1024$ , que corresponde a 0.3516 graus de precisão.

Este encoder possui, ainda, uma terceira pista, com uma fenda de  $360^\circ/1024$ , que serve para referência de inicialização da posição de seu eixo, sendo que num sistema de controle de posição, esta referência serve como o zero do encoder e consequentemente, pode servir como zero para inicialização da posição do mecanismo robótico.

Os sinais elétricos das três pistas são produzidos por fotosensores, que detectam a passagem ou não de luz num disco impresso em impressora laser em plástico transparente, sendo que, no disco está impresso o desenho da pista de contagem, com a resolução desejada, e o desenho da pista de referência.

Uma mascara com as fendas para a luz dos sensores, com uma precisão idêntica à do disco, é usada para facilitar e garantir a defasagem de  $90^\circ$  entre os dois sensores das pistas de contagem, quando da sua montagem e ajuste mecânicos.

Os desenhos dos encoders e das mascaras impressos constam no Anexo A.4, tendo sido impressos vários discos e mascaras por folha para um maior aproveitamento de material. Os desenhos foram obtidos através de programação em linguagem PostScript e impressos em impressora de 600d.p.i. (pontos por polegada de resolução), responsável pela alta qualidade do resultado final.

## 2.2 - Interface Decodificadora de Sinais de Encoder

A interface decodificadora de sinais de encoder é uma interface de monitoramento de baixo nível de complexidade, onde os sinais de até quatro encoders óticos incrementais são tratados em tempo real, com a finalidade de realizar a sua conversão para um formato que as interfaces de contadores numéricos (interfaces de monitoramento de nível mais alto de complexidade) possam entender.

Ou seja, os encoders fornecem dois bits que variam conforme a variação de posição do eixo do encoder, além disto, estes dois bits assumem valores lógicos "0" e "1", defasados, no tempo, de meia onda, a medida que seu eixo gira.

Assim, estes sinais têm que ser convertidos em, um bit que forneça o sentido de rotação do encoder (horário ou anti-horário, conforme a defasagem entre os dois bits vindos do encoder) e um bit que forneça o instante em que a posição do eixo variou (neste instante este bit varia de "0" para "1", fazendo com que o contador incremente ou decamente seu valor, atendendo à especificação dos componentes eletrônicos utilizados nas interfaces de contadores).

Quanto ao seu funcionamento, quando o eixo do encoder completa uma revolução, cada um dos bits fornecidos, sofreu N transições lógicas entre "0" e "1", onde N é o número de pulsos por pista do encoder (valor que caracteriza a sua precisão de medida angular, estando tipicamente entre 256 e 1024), a interface decodificadora trata os sinais defasados dos dois fotossensores dos encoders, obtendo uma precisão 4 vezes maior que N, fornecendo a unidade de medida angular mecânica como sendo :

$$1 \text{ Pulso Contado} = \frac{360 \text{ Graus}}{4 \times \text{Número de Pulsos por Pista}} [ \text{Graus} ] , \quad (1)$$

A interface codificadora de sinais de encoders incrementais tem as seguintes características de aplicação :

Realiza a conversão do formato incremental (dois bits) para o formato sentido e variação (dois bits), tem capacidade de converter os sinais de quatro encoders, enviando estes sinais convertidos às respectivas interfaces contadoras numéricas (X, Y, Z e W).

Recebe alimentação para seus circuitos e um sinal de relógio auxiliar de interfaces de funções auxiliares e alimentação, sendo que, a frequência deste relógio, dá máxima frequência dos sinais gerados pelos encoders que podem ser codificados, sem que ocorram perdas de dados vindos dos encoders, uma vez que este relógio é responsável pela frequência de amostragem dos circuitos do decodificador.

Os sinais codificados por esta interface têm as seguintes características de temporização:

O sinal de relógio deve ter uma frequência mínima de, por exemplo, 4 MHz para um funcionamento seguro, quando, se está utilizando uma rotação de 14600 R.P.M. com um encoder que possua 4096 pulsos por pista (uma precisão de 0,0220 graus de rotação do eixo do encoder).

Pode-se calcular a frequência mínima necessária para uma operação segura pela fórmula :

$$F_{\text{máx.}} = \frac{4 \cdot N \cdot n_{\text{máx.}}}{60} \text{ [ Hz ]}, \quad (2)$$

onde :

N é o número de pulsos por pista do encoder.



nmáx. é a velocidade angular máxima alcançada pelo encoder durante a operação, medida em R.P.M..

Os sinais codificados (de sentido ( $\overline{U/D}$ ) e de variação (ou relógio, CLK)) não são válidos, ou seja não têm significado correto, quando o sinal de relógio codificado (CLK) está num nível lógico "1", pois, quando o relógio está em "1" o sinal de sentido de rotação pode variar, isto se deve ao fato de que durante este período a interface codificadora está definindo qual o sentido de rotação do encoder, consequentemente, o valor do bit de sentido pode conter uma informação incorreta, sendo assim, é necessário que os sinais codificados não sejam processados nestes intervalos, para tanto, as interfaces de contadores consideram a transição de "1" para "0" do bit de variação de posição, como sendo o instante em que se deve incrementar ou decrementar os seus valores atuais de posição.

## 2.3 - Interfaces de Contadores

A interface contadora numérica digital é uma interface de monitoramento de alto nível de complexidade, onde sinais de contagem, em formato adequado, são processados em tempo real e podem ser monitorados (não necessariamente em tempo real) pelo operador, e/ou pelo computador, com a finalidade de dispensar o computador da grande carga de processamento que o monitoramento, em tempo real, de encoders incrementais requer, não devido à complexidade de seu processamento, mas devido à alta taxa de variação que podem atingir estes sinais.

Esta interface converte, ainda, os dados de posição processados em um formato adequado para leitura e processamento pelo computador, além disso, seus valores podem ser alterados, permitindo a inicialização da posição do encoder com relação à posição de "zero" do mecanismo do robô.

Opcionalmente, a interface permite, ainda, que se conecte, a ela, uma interface com displays digitais, para que o operador possa monitorar visualmente, independentemente do computador, os valores de contagem dos encoders, possibilitando um diagnóstico seguro do funcionamento dos subsistemas de sensores e o dos subsistemas de aquisição.

A opção de visualização da posição do mecanismo, através de displays, pode ser muito importante e útil, quando o mecanismo automático sob monitoramento é do tipo de eixos coordenados, como, por exemplo, tornos, fresas e outras máquinas operatrizes; nestes casos, a unidade de contagem pode ser calculada de modo a coincidir com a precisão dos eixos da máquina, possibilitando um monitoramento direto, nos displays, da posição ou orientação, da peça, ou da ferramenta (por exemplo, medida em milésimos de milímetros, centésimos de milímetros, centésimos de grau, etc...).

Em aplicações simples, em mecanismos de posicionamento manual, pode-se operar apenas com os subsistemas de sensores e monitoramento de posição de encoders, sendo que a interface de contadores pode operar independentemente da presença de um computador.

Lembrando que a unidade de contagem desta interface é um pulso de variação de posição do encoder, sendo o valor mostrado nos displays medido nesta unidade, cabendo ao software de aquisição e monitoramento, quando necessário, a sua conversão para uma unidade mais útil ao operador (por exemplo, radianos ou graus, com as aproximações necessárias).

A interface de contadores digitais tem as seguintes características de aplicação :

Esta interface possui seis dígitos significativos (podendo contar de 0 à 999999 posições de encoder) que são exibidos, opcionalmente, em seis mostradores digitais, a arquitetura desta interface possibilita a sua expansão para até oito dígitos significativos sem a necessidade de alteração das outras interfaces do sistema.

Recebe os sinais codificados dos encoders (sentido e variação) realizando o incremento ou decremento da posição atual conforme indicar o sinal de sentido de rotação, possibilitando, opcionalmente, que o valor atual seja exibido nos mostradores digitais em tempo real.

Além disto, ela deve ser capaz de armazenar os valores presentes nos contadores, em um certo instante, em registradores, para que os dados não se alterem durante o processo de leitura pelo computador, ou seja, os dados adquiridos pelo módulo de supervisão, no computador, são os dados previamente "congelados" (armazenados em registradores), isto se dá, através de um sinal de controle ( $\overline{\text{LATCH}}$ ) que o computador envia a estas interfaces.

Ela recebe ainda, um sinal que habilita a contagem ( $\overline{\text{CTEN}}$ ), que possibilita uma inicialização, e/ou calibração, das posições dos encoders, tarefa esta que pode ser realizada pela função carrega dados (sinal  $\overline{\text{LOAD}}$ ), desta interface, da seguinte maneira; Coloca-se o valor que se deseja carregar no registrador de saída de dados do computador e então envia-se a palavra de controle que seleciona o par de dígitos que se deseja alterar, e em seguida repete-se a palavra com o sinal de carregar ativo (bit do sinal  $\overline{\text{LOAD}} = 0$ , na palavra de controle).

A arquitetura da interface consiste em seis circuitos contadores década binários ligados em cascata, compartilhando o mesmo barramento de carga de dados provenientes de uma interface de controle e inicialização, sendo que esta interface pode ser a própria interface digital, ou pode ser uma interface controladora de acesso multiplexado; além disso, suas saídas, são ainda, enviadas aos decodificadores / drivers dos displays (onde o formato decimal binário é convertido para sete segmentos, opcionalmente, caso a interface de displays estiver presente, acionando os sete diodos emissores de luz de cada dígito do mostrador).

Além disto, essas saídas, decimais em formato binário, são enviadas aos três registradores de oito bits (duas décadas por registrador), onde estão disponíveis para serem armazenados e/ou adquiridos, conforme comando do computador, sendo que as saídas destes registradores compartilham um barramento local de leitura de dados para o computador, onde

são acionadas, exclusivamente, uma de cada vez (um registrador de cada vez), pela palavra de controle, durante as operações de leitura.

A interface possui, ainda, um barramento local de escrita, utilizado para inicialização dos valores de contagem (posição) dos eixos dos encoders, onde são carregados os valores desejados e comandados os respectivos carregamentos dos três pares de contadores.

## **2.4 - Interface de Acesso Multiplexado**

A interface de acesso multiplexado é uma interface de controle de comunicação e multiplexação de dados, ela deve realizar o controle das interfaces contadoras, sendo que os sinais de controle vindos da interface digital são distribuídos para estas interfaces, selecionando quais funções estas devem realizar em determinado momento, isto é feito a partir de uma palavra de controle que a interface de acesso recebe do módulo de monitoramento e supervisão.

Dentre as funções desta interface temos, a seleção da interface de contadores e do par de dígitos para escrita (inicialização), do par de leitura (aquisição da posição atual), além de controlar a habilitação dos contadores (que devem ser desabilitados durante a inicialização).

Além disto, ela deve realizar uma isolação e amplificação dos sinais que recebe e envia aos subsistemas aos quais está ligada, visto que a quantidade de circuitos com os quais ela se comunica através de seu barramento interno de dados e controle é grande, este fato caracterizaria uma carga excessiva numa ligação direta das interfaces contadoras com a interface paralela, tornando indispensável a isolação e amplificação dos sinais que se deseja enviar e receber.

Caso isto não fosse feito, este excesso de carga poderia resultar em mal funcionamento durante operação e até danos nos circuitos da interface paralela com consequências que poderiam atingir a interface principal do computador.

A interface controladora de acesso multiplexado tem seguintes características de aplicação :

É constituída de circuitos buffers (amplificadores isoladores de corrente) que atuam sobre os sinais recebidos, enviando-os às interfaces contadoras e atuam sobre os sinais dos dados de contagem, isolando-os e enviando-os para a interface de interface paralela.

Possui os decodificadores de parte da palavra de comando que selecionam qual par de dígitos (Pares : 1 e 2, 3 e 4, 5 e 6), de qual das interfaces contadoras numéricas (Interfaces contadoras dos eixos : X, Y ou Z) que serão acessados pela interface paralela para leitura ou escrita.

Na operação de escrita nas interfaces contadoras, o decodificador de escrita seleciona e escreve no par de dígitos desejado ao receber o sinal que controla o modo de operação de escrita nas interfaces (Sinal :  $\overline{\text{LOAD}}$ ), sinal que ao mudar de 1 para 0 faz com que seja carregado o par de contadores selecionado, com o conteúdo previamente carregado no registrador de dados de saída.

Recebe os sinais de contagem  $\overline{\text{CLK}}$  (relógio) e U/D (contar acima ou abaixo) da interface de codificação dos sinais dos encoders e os transfere para as respectivas interfaces contadoras numéricas (Interfaces : X, Y e Z).

Recebe alimentação para seus circuitos e o sinal externo de controle de contagem ( $\overline{\text{CTEN}}$  (função de habilitar ou desabilitar) da interface de funções auxiliares e alimentação, distribuindo este sinal e a alimentação para as interfaces contadoras numéricas.

Possibilita a conexão de três interfaces contadoras numéricas de seis dígitos, transmitindo à essas os sinais de dados e controle e recebendo dessas os sinais de dados, o projeto desta interface possibilita que a partir de pequenas alterações a conexão de até quatro interfaces de oito dígitos, sem alterar os componentes eletrônicos utilizados, apenas aproveitando-os completamente, pois estão sub-utilizados na atual implementação (isto não implica que o atual projeto pode ter seu custo reduzido pela remoção das partes não utilizadas, isto não é possível, pois estas partes se encontram em circuitos integrados não divisíveis).

A arquitetura da interface é a seguinte :

Uma palavra de controle de seis bits deve estar presente em sua entrada de controle (neste caso esta palavra é fornecida pela interface paralela), com seus bits com as seguintes funções :

Bits 0 e 1 : selecionam a interface contadora numérica que se deseja acessar :

Bit 0	Bit 1	Interface selecionada
0	0	Eixo X
0	1	Eixo Y
1	0	Eixo Z
1	1	Eixo W (possível expansão)

Bits 2 e 3 : seleccionam o par de dígitos que se deseja acessar da interface contadora selecionada pelos bits 0 e 1 :

B it 2	B it 3	Par de dígitos seleccionados
0	0	1 e 2 (menos significativos)
0	1	3 e 4
1	0	5 e 6 (mais significativos)
1	1	7 e 8 (possível expansão)

Bit 5 : ( $\overline{\text{LATCH}}$ ) Quando este sinal vai de 1 para 0, os valores de todos os dígitos de todas as interfaces são armazenados para que o computador possa lê-los sem que estes variem durante as operações de leitura, os dados ficam armazenados até que este sinal varie de 0 para 1.

Bit 6 : ( $\overline{\text{LOAD}}$ ) Quando este sinal vai de 1 para 0, o valor presente nos bits de dados são carregados no par de dígitos seleccionados previamente pelos bits 1, 2, 3 e 4.

Quando se deseja ler os dados de um par de par de dígitos, basta seleccioná-los conjuntamente com o sinal  $\overline{\text{LATCH}}$  em 0, lembrando-se de não levar este sinal para 1 até que todas as leituras tenham sido realmente efetivadas.

Os sinais de dados não são alterados por esta interface, eles são apenas amplificados e direcionados para o par seleccionado pela palavra de controle.

## **2.5 - Interface Digital**

A interface digital é uma interface de comunicação de dados de uso geral, de 32 bits de entrada e 32 bits de saída, que é utilizada no sistema para entrada e saída de dados dos contadores e para saída de dados de controle de atuadores.

Listamos o programa utilizado para testes, o esquema dos circuitos e o lay-out da interface de circuito impresso da interface digital, no anexo B.

A seguir, descrevemos a Interface de Entrada e Saída Digital Paralela, com 8 Portas de 8 Bits cada (quatro de entrada e quatro de saída), otimizada para se obter uma melhor relação custo / benefício, conseguindo-se uma expansão de sua capacidade com relação a projetos anteriores, além disso, no sistema global, controlador - supervisor - sistemas de aquisição - sistemas de atuação, seriam necessárias, utilizando-se interfaces convencionais, mais interfaces de entrada e saída digitais, fato que justifica, também, a construção desta nova interface, reduzindo-se o número total de interfaces necessárias para uma determinada aplicação.

### **2.5.1 - Descrição da Interface**

A função desta interface é realizar a transferência, entrada e saída de sinais de dados e comandos, que se deseja enviar e receber dos sistemas de aquisição de dados e sensores e que se deseja enviar aos atuadores, ela compatibiliza as características eletrônicas e lógicas do barramento do computador com as características dos sistemas externos ao computador, como por exemplo a interface de acesso multiplexado do sistema de aquisição e monitoramento de dados de encoders.

Esta interface somente transfere dados e comandos, não realizando nenhum processamento sobre os mesmos, os seus circuitos devem processar apenas os sinais de



comunicação provenientes do barramento do computador, atendendo às instruções de transferência de dados, quando seu endereço se faz presente.

Esta interface tem as seguintes características de aplicação:

É compatível com o barramento padronizado I/O CHANNEL da IBM, utilizado nos microcomputadores IBM PC,XT e AT, tendo a alimentação dos seus circuitos eletrônicos feita a partir do próprio barramento do computador ao qual está conectada, não necessitando de alimentação externa.

A sua arquitetura consiste de 4 registradores, de 8 bits cada, de saída, e 4 registradores, de 8 bits cada, de entrada, esta interface pode ser utilizada com qualquer outro fim de comunicação, sendo que, na montagem com o sistema de aquisição de dados e monitoramento, são utilizados um registrador de entrada e um de saída para enviar e receber os dados de contagem dos encoders óticos, e seis bits de um outro registrador de saída para enviar as palavras de comando para o sistema.

Os registradores estão mapeados em I/O (Entrada / Saída), ou seja, as leituras dos dados disponíveis nos registradores de entrada devem ser feitas através de instruções IN (em linguagem de máquina dos processadores utilizados nos computadores compatíveis com a família IBM PC-AT) e as escritas dos dados nos registradores de saída, devem ser feitas através de instruções OUT, ambas realizadas nos endereços digitais dados na tabela abaixo :

Operação	Endereço (Bits)										Função
	A	A	A	A	A	A	A	A	A	A	
	9	8	7	6	5	4	3	2	1	0	
Escrita	1	s	s	s	s	s	s	s	s	0	0 Escreve no registrador OUT
	8	7	6	5	4	3	2				0
Escrita	1	s	s	s	s	s	s	s	s	0	1 Escreve no registrador OUT
	8	7	6	5	4	3	2				1
Escrita	1	s	s	s	s	s	s	s	s	1	0 Escreve no registrador OUT
	8	7	6	5	4	3	2				2
Escrita	1	s	s	s	s	s	s	s	s	1	1 Escreve no registrador OUT
	8	7	6	5	4	3	2				3
Leitura	1	s	s	s	s	s	s	s	s	0	0 Lê o registrador IN 0
	8	7	6	5	4	3	2				
Leitura	1	s	s	s	s	s	s	s	s	0	1 Lê o registrador IN 1
	8	7	6	5	4	3	2				
Leitura	1	s	s	s	s	s	s	s	s	1	0 Lê o registrador IN 2
	8	7	6	5	4	3	2				
Leitura	1	s	s	s	s	s	s	s	s	1	1 Lê o registrador IN 3
	8	7	6	5	4	3	2				

Obs.: sn são os bits do endereço base que devem ser selecionados através de "jumpers" na interface. Nesta seleção a chave n fechada significa sn = 0 e a chave n aberta significa sn = 1.

Tabela 2.1 - Configuração de Endereços da Interface Digital.

No processo de seleção dos "jumpers", estão sendo alterados os bits com os quais o endereço presente no barramento do computador está sendo comparado, deste modo escolhe-se à qual endereço a interface deverá responder numa operação de I/O.

Deste modo, o endereço base selecionado, deve ser escolhido de maneira que não coincida com os endereços já utilizados por outras interfaces no barramento ou com endereços de circuitos de uso interno do computador, visto que, isso pode provocar erros de funcionamento do computador e dos sistemas nele conectados. A bibliografia com os endereços recomendados para interfaces desenvolvidas pelos usuários do computador [7], recomenda o endereço 300 hexadecimal, para interfaces protótipos.

Os sinais dos registradores escrita e leitura são dispostos externamente ao computador, através de dois conectores padrão DIN para flat cable (cabo chato) de 64 pinos.

## **2.6 - Software para Aquisição, Monitoramento, Supervisão e Inicialização**

O software para aquisição, monitoramento, inicialização e supervisão de sistemas mecânicos consiste de várias possíveis bibliotecas de procedimentos, sendo responsável pelo controle dos hardwares de aquisição e monitoramento, fornecendo para o usuário e/ou para o módulo de controle os dados adquiridos dos sensores, sendo assim, do ponto de vista do usuário, os hardwares de aquisição e monitoramento são invisíveis.

Para isso, as bibliotecas de procedimentos, especificamente desenvolvidas para cada tipo de hardware, devem inicializar estes subsistemas, caso estes necessitem, no momento em que se está ligando o sistema mecânico, ou no momento em que se deseja recalibrar o sistema, conforme o seu procedimento de inicialização e calibração globais.

Por exemplo, as posições dos eixos, adquiridas através de encoders e de interfaces de contadores, devem ser inicializadas, ou seja, o mecanismo deve ser levado a uma posição mecânica conhecida, em termos de valores de contagem dos encoders, sendo que, em seguida,

estes valores devem ser carregados nas interfaces contadoras para sua inicialização. Este procedimento pode ser repetido sempre que se desejar recalibrar o sistema.

Durante a operação, os procedimentos de supervisão adquirem os valores dos sensores, controlando o processo, por exemplo, de multiplexagem e endereçamento das interfaces e dos pares de dígitos a serem lidos de posições de encoders em interfaces contadoras, transformando os valores lidos em valores mais úteis para o usuário, como por exemplo, os valores de posição da garra de um robô em coordenadas cartesianas (realizado o cálculo do modelo Cinemático direto do mecanismo).

No módulo de supervisão, pode ser feito, ainda, o processamento dos dados adquiridos, visando a sua análise em termos de faixas de segurança, de cálculos de variáveis de estado, modelos geométricos de robôs, e etc...

## **2.7 - Conclusões**

Neste capítulo, foi descrito o módulo de monitoramento e supervisão e seus subsistemas, a metodologia utilizada na sua implementação, suas características técnicas e de utilização, e as possibilidades de sua aplicação. Mostramos, ainda, a flexibilidade dos subsistemas desenvolvidos, quanto a sua estrutura e utilização.

A seguir, descrevemos o módulo de Acionamento e Controle, que se interliga com o módulo de monitoramento e supervisão em várias configurações possíveis.

## **Capítulo 3**

### **- Módulo de Controle e Interfaces de Acionamento**

O módulo de acionamento e controle é responsável, do ponto de vista global de controle, pelo processamento dos valores das variáveis observadas dos sistemas mecânicos (provenientes do módulo de aquisição, monitoramento e supervisão) e dos valores desejados para estas variáveis (provenientes do módulo de programação de trajetórias e tarefas), devendo fornecer em tempo real, para os atuadores do mecanismo, os valores das variáveis de controle.

A conversão de sinais de controle lógicos em sinais de potência elétrica e posteriormente, em potência mecânica, se dá através dos subsistemas de acionamento e dos atuadores, conforme o tipo de atuador utilizado, é necessário um tipo de subsistema de acionamento.

Para se obter um valor de saída mais preciso no atuador, pode-se utilizar, nos procedimentos de cálculo dos algoritmos de controle do mecanismo, uma compensação destes valores mediante a utilização de funções de transferência inversas, ou curvas de características, sempre em software.

A adoção do módulo de controle em software se faz necessária devido ao fato de que se deseja utilizar algoritmos de controle diferentes para os mecanismos, além disso, a natureza não linear dos modelos dinâmicos diretos e inversos de robôs, utilizados nos cálculos da malha de controle, tornam inviáveis um controle desacoplado de cada junta do mecanismo robótico,

devido aos problemas de esforços consequência das acelerações da gravidade, centrífugas, coriolis e giroscópicas, que fazem parte do processo de movimentação de um mecanismo robótico.

Com exceção da aceleração da gravidade, os outros efeitos se tornam consideráveis com o aumento das velocidades de deslocamento do mecanismo, assim, caso se deseje utilizar mais eficientemente o sistema robótico, deve-se levar em conta estes efeitos não lineares; esta é a razão principal para o desenvolvimento deste sistema, cujo módulo de controle precisa ser obrigatoriamente em software.

A seguir, descrevemos os subsistemas deste módulo, desenvolvidos para a implementação do sistema de controle flexível :

### **3.1 - Interface Geradora de PWM**

A interface conversora Digital - Analógica, implementada através de um gerador PWM (pulse width modulation - modulador por largura de pulsos), é uma interface de conversão de uso geral, para utilização pelo controlador, no acionamento de motores, válvulas, resistências, etc.

A seguir, descrevemos a interface geradora de PWM, projetada para se obter uma melhor relação custo / benefício, conseguindo-se uma grande flexibilidade em sua capacidade de aplicação, devido ao fato de que uma mesma interface de baixo custo pode ser utilizada para se gerar vários padrões de PWM, visto que isto é feito em ROM (circuito integrado de memória somente de leitura), que pode ser programada conforme se deseje; sendo que, para esta interface foi desenvolvido um programa gráfico onde se pode editar as formas de onda que esta interface gerará mediante a programação de sua memória ROM.

As interfaces de conversão D/A implementadas através de PWM, possuem ainda, a vantagem de uma eficiência extremamente elevada, quando comparadas a outras interfaces, pois trabalha como uma chave, ligando e desligando, enquanto outras se comportam como resistências variáveis, dissipando a energia que não é enviada a carga.

A única desvantagem do PWM é que se faz necessária uma filtragem do sinal gerado, pois este é constantemente alternado entre zero e o máximo de potência, fato amenizado por filtros capacitivos, indutivos e por filtros intrínsecos as cargas, como por exemplo motores, que são altamente indutivos. Além disso, pode-se construir um PWM com alta frequência de chaveamento para se facilitar a filtragem, porém com o cuidado na observância das restrições de eficiência dos semi-condutores de potência utilizados.

O esquema dos circuitos e o lay-out da interface de circuito impresso da interface geradora de PWM são apresentados no Anexo B.

### **3.1.1 - Descrição da Interface**

A função desta interface é realizar a conversão, dos sinais de controle provenientes do controlador, através da interface de entrada e saída de dados paralela, na forma de uma palavra de controle (um byte, com valor de 0 a 255), para enviá-los à uma interface de acionamento de potência.

Esta interface compatibiliza as características eletrônicas e lógicas do sinal de controle proveniente do controlador, digital de oito bits, com as características das interfaces de acionamento, que podem ser do tipo chaveadas, ou seja, cuja potência elétrica é ligada e desligada a uma alta frequência, fornecendo à carga (através do atuador) uma potência média proporcional ao valor do sinal de controle desejado.

Esta interface pode operar de quatro modos, quanto a saída pode gerar dois tipos de sinal de largura modulada, no primeiro tipo, a largura do pulso periódico gerado é proporcional ao valor do sinal de controle (PWM real), no segundo modo, são gerados o número de pulsos dado pelo sinal de controle, espalhados no tempo, durante um período composto pelo máximo de pulsos possíveis (modulação por posição de pulso); portanto, o primeiro tipo de sinal gerado é recomendado para interfaces de potência cujos circuitos chaveadores não podem suportar frequências muito altas (por exemplo, transistores integrados de baixo custo), enquanto o segundo é recomendado para interfaces de potência que possuem circuitos de chaveamento de alta frequência (por exemplo, transistores integrados de potência de alta frequência tipo MOSFET).

Quanto ao tipo de acionador, esta interface, também pode gerar dois tipos de saídas, no primeiro, pode-se controlar acionadores unidirecionais, ou seja, cuja potência elétrica pode ser variada de 0 a 100%, no outro tipo, pode-se controlar acionadores bidirecionais, cuja potência elétrica pode ser variada de -100% a +100%; para isso, o processamento do sinal de entrada é realizado de outra forma, o sinal de controle (um byte) passa assumir valores na faixa de -127 a 127.

Esta interface tem as seguintes características de aplicação:

.Seu sinal de controle é gerado através de uma palavra composta por oito bits organizados num byte, compatível com as saídas da interface de entradas e saídas digitais paralela. Suas oito saídas também seguem o mesmo padrão, sendo implementadas com circuitos integrados da família TTL-74LS, e finalmente sua alimentação se dá através de um conector para uma fonte externa.

.A sua arquitetura consiste de um registrador de oito bits (opcional, necessário apenas no caso de sinais de entrada ruidosos ou sem "bufferização"), que guarda o valor do byte de



controle, passando-o à uma memória somente de leitura (ROM) de 64 Kbytes por oito bits, que é o centro do gerador de PWM, nesta memória, o valor do byte de controle é utilizado para se decodificar e acessar um dos 256 blocos, de 256 bytes da memória ( $256 \times 256 \text{ bytes} = 64 \text{ Kbytes} = 65536 \text{ bytes}$ ), nestes blocos, em que dividimos a memória, estão desenhadas as formas de onda que desejamos como saídas do gerador, portanto, a memória funciona como uma função, pois para cada valor de entrada temos acesso a um bloco de valores programável.

Tendo a possibilidade de programar na memória os valores de cada bit, podemos acessar sequencialmente os bytes num bloco, se conectarmos um contador de 256 estados (oito bits) aos oito bits menos significativos da memória, e sendo assim, os 256 valores num bloco são acessados e repetidos indefinidamente em função do valor de controle, fornecendo em cada bit de saída uma forma de onda periódica composta por 256 pulsos programáveis.

Nota-se, que esta implementação de um gerador de 256 funções programáveis para cada um dos oito bits da saída, é infinitamente mais ampla do que a aplicação dada a ela nesta interface, através da mudança dos 64 Kbytes gravados na memória, podem-se obter vários tipos de funções, para muitas aplicações diferentes, por exemplo, pode-se gravar na memória a sequência de bits necessária para se obter os sinais de controle dos transistores de um inversor trifásico.

Devido à abstração que esta interface proporciona, através da implementação das funções numa memória programável, foi desenvolvido um programa de computador que é capaz de editar as formas de onda, dos oito bits de um bloco de cada vez, sendo simples a sua utilização, pois graficamente na tela são apresentadas as formas de onda, e com o auxílio de um ponteiro do tipo mouse, pode-se editar os bits nestas formas de onda. Ao final da edição, o usuário têm a opção de salvar as formas de onda num arquivo binário que é a imagem, que deve ser gravada na memória por um gravador de memórias programáveis.

Na gravação da memória foi utilizado o gravador de memórias e de arrays lógicos programáveis da Data I/O [15], o 2900 Programing System, que leu o arquivo gerado pelo programa editor de formas de onda, e o gravou na memória, cabe salientar que esta implementação de um gerador PWM só foi possível com a utilização de circuitos programáveis, no caso a memória, que cumpre o papel de simplificar a implementação do gerador, aumentar as possibilidades de sua utilização e baratear o produto final, através da redução do número de componentes e do tamanho físico da interface.

A frequência dos sinais de saída, do gerador PWM, é dada por um oscilador, e pode ser alterada pela mudança nos valores dos capacitores e resistores que o compõem, esta frequência têm um limite máximo, que é dado pela frequência máxima de operação da memória, não consistindo um problema prático, pois com a memória lenta (de baixo custo) implementada, poderia se controlar uma interface de potência com frequência até 500KHz, que é uma frequência elevada para os casos atuais.

A saída da memória é passada por um registrador, que serve para amplifica-la, uma vez que a memória selecionada têm capacidade de potência de saída baixa em relação a desejada, devido à sua tecnologia de baixo consumo de energia, memória da família de circuitos integrados CMOS.

Para a seleção e operação do gerador PWM em seus quatro modos, devemos configurar um "jumper" na interface, além disso, temos outros dados operacionais do gerador PWM especificados na tabela 3.1:

Modo de Operação	Unidirecional, Alta Frequência	Unidirecional, Baixa Frequência	Bidirecional, Alta Frequência	Bidirecional, Baixa Frequência
	Bit 0	Bit 2	Bits 4 e 6	Bits 4 e 6
Saída(s) Válida(s)				

Saída(s) Válida(s) Negada(s)	Bit 1	Bit 3	Bits 5 e 7	Bits 5 e 7
Sinal de Entrada	0 a 255	0 a 255	-127 a +127	-127 a +127
Sinal de Saída	0 a 100%	0 a 100%	-100% a +100%	-100% a +100%
Posição do "jumper"	5-6	5-6	3-4	1-2

Tabela 3.1 - Configuração do Tipo de PWM Gerado.

Na Tabela 3.1, podemos observar :

As saídas negadas estão presentes, para facilitar a utilização, pois são comuns as interfaces de potência que utilizam sinais de controle negado, como por exemplo a interface de potência projetada.

Quando em operação bidirecional, são geradas duas saídas para conjuntos de dois transistores que operam complementarmente, enquanto um está acionado (recebendo o sinal PWM), o outro deve estar desligado, visto que a operação bidirecional nas interfaces de potência normalmente implica em mutua exclusão, devido ao seu funcionamento interno.

Quando em operação bidirecional, ainda, deve-se notar que o sinal de entrada (um byte) passa a ser interpretado como um valor de sete bits (os sete bits menos significativos), mais um bit (o oitavo bit) com o sinal, bit 8 = 0 significa saída positiva (bits 4 e 5 ativos e bits 6 e 7 desligados), bit 8 = 1 significa saída negativa (bits 6 e 7 ativos e bits 4 e 5 desligados). Ao utilizarmos os pares de saídas bidirecionais, podemos inverter estes, para obter um sentido de rotação de um motor horário ou anti-horário.

No processo de seleção do modo de operação da interface através dos "jumpers", esta sendo alterado o tamanho do bloco que é acessado pelo contador, de 256 estados (para o modo unidirecional), para 128 estados (no modo bidirecional), isto porque, nos modos bidirecionais o bit 8 é utilizado como indicador de sinal, reduzindo a precisão possível para metade, ou seja, a precisão no modo unidirecional é  $1/256 = 0.39\%$  (numa escala de 0 a 100%), enquanto a precisão no modo bidirecional é  $1/128 = 0.78\%$  (num e no outro sentido).

Os sinais de entrada e saída são dispostos externamente à interface através de dois conectores padrão DIN para flat cable (cabo chato) de 16 pinos cada, porém somente os bits de saída válidos para cada modo de operação têm uma forma de onda utilizável.

### **3.1.2 - Programa para Geração de Formas de Onda para PWM**

O pacote computacional desenvolvido para geração e edição das formas de onda que a interface geradora de PWM produz, é um programa gráfico desenvolvido para a geração do arquivo binário que será gravado no circuito integrado de memória (ROM) que será utilizado na interface geradora de PWM.

Este programa possibilita a edição bit a bit, através do uso de um mouse, das seqüências de 256 bits do sinal PWM geradas, para cada valor digital de oito bits, colocado na entrada da placa.

Como podemos ver na figura 3.1, o programa foi implementado utilizando-se os protótipos dos objetos desenvolvidos para o módulo de interface com o usuário, objetos como o mouse, o teclado, o sistema gráfico, sistema de janelas móveis, botões para avanço e retrocesso, botões para salvar e ler arquivos de PWM, sistema de gravação e leitura de arquivos, campos de exibição de dados e objetos gráficos específicos, para desenho e edição das formas de onda.

Podemos ver, ainda, na figura, dois blocos com as seqüências de 128 bits por 8 bits geradas para o valor de entrada igual a 11 (uma vez que a seqüência gerada têm 256 bits e não cabe na tela, esta foi dividida em dois blocos, como mostram os valores 22 e 23 nos respectivos quadros de controle), vemos na primeira linha, por exemplo, uma forma de onda PWM de alta freqüência, com onze bits ligados (6 bits ligados no primeiro bloco e continuando no segundo bloco, 5 bits ligados) igualmente espaçados, sendo que na segunda linha temos esta mesma seqüência de bits negada.

Na terceira e quarta linhas, temos uma forma de onda PWM de baixa freqüência, com onze bits ligados (onze bits consecutivos ligados apenas no primeiro bloco), sendo que na quarta linha temos esta mesma seqüência de bits negada.

Nestes exemplos, tanto as formas de onda de alta freqüência, como as de baixa, fornecem um valor médio de saída de  $11/256$ , de onde pode se concluir que, programando-se convenientemente as 256 possíveis seqüências de 256 bits (65.536 bits do circuito de memória utilizada) consegue-se um circuito cujas saídas são PWM com valor médio igual a  $n/256$  onde  $n$  é o valor de entrada de oito bits (0 a 255). Podendo-se, ainda, programar um tipo de PWM diferente para cada uma das oito saídas do circuito de memória utilizado.

Podemos ver, ainda, no gerador de formas de onda, nos quadros de controle os números do bit que está sendo editado (neste caso : 116 na seqüência de 0 a 127, por 0 entre os bits de saída de 0 a 7), além disso, temos os botões com as setas horizontais brancas, que servem para se avançar e retroceder um intervalo de 128 bits na memória total de 65.536 bits, além do cursor do mouse (neste caso apenas um pixel acima do sexto bit ligado da primeira linha).

Vemos, ainda, uma janela móvel, utilizada para teste e debug de cores e do cursor do mouse, que mostra as coordenadas do mouse na tela (581,75), a cor do cursor do mouse (1 - azul) e a cor do fundo sob o cursor do mouse (0 - preto).

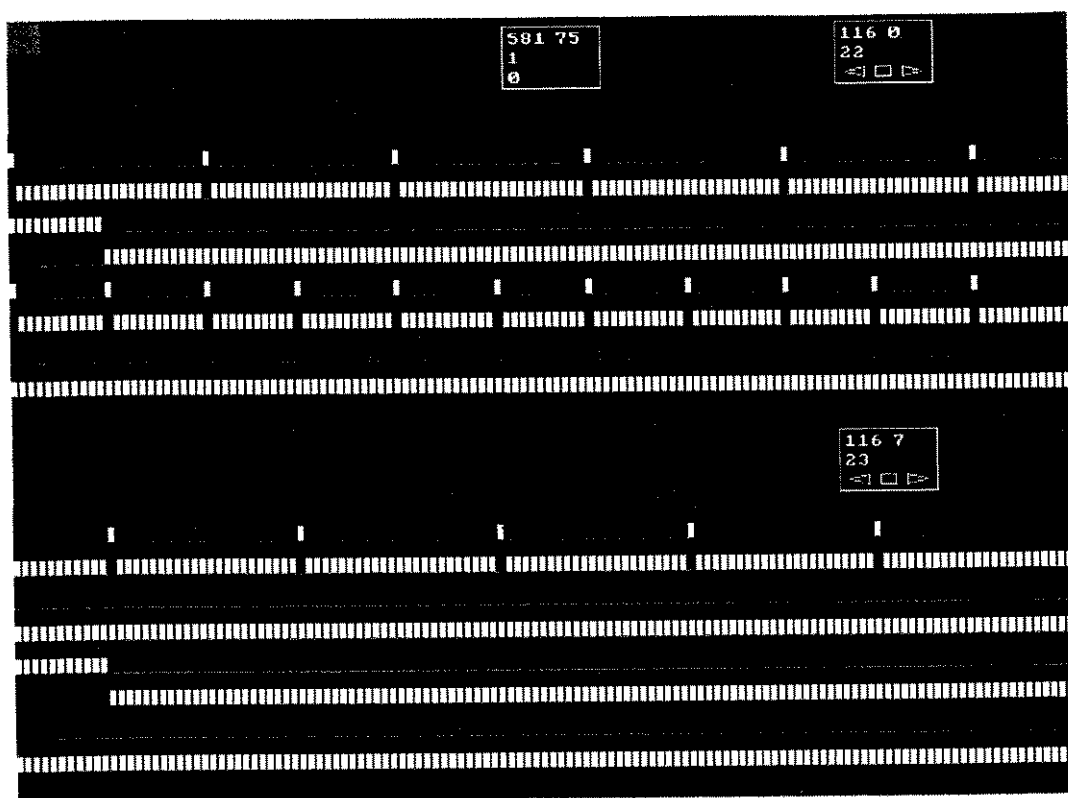


Figura 3.1 - Tela Gráfica do Programa Gerador de Formas de Onda para PWM

### 3.2 - Interfaces de Acionamento de Potência Chaveada

As duas interfaces de acionamento de potência elétrica desenvolvidas, são interfaces de potência de uso geral do tipo chaveadas, para utilização no acionamento dos atuadores dos mecanismos robóticos, a primeira aciona cargas DC bidirecionais, e a segunda aciona cargas DC unidirecionais, ambas tendo sido projetadas para serem controladas por interfaces geradoras de PWM de baixa frequência, devido ao baixo custo relativo dos componentes

empregados em sua confecção, o que limita as suas velocidades de resposta aos sinais de controle.

Essas interfaces podem ser utilizadas no acionamento de motores, válvulas, etc; o controlador as comanda, enviando o valor de potência relativa desejada, através de uma interface de interface de saída paralela, em seguida, este valor de controle é convertido num sinal PWM, numa interface geradora de PWM, sendo que o sinal resultante é utilizado para chavear a potência elétrica fornecida às cargas, produzindo um valor de potência elétrica proporcional a palavra de controle enviada pelo controlador.

O esquema dos circuitos das interfaces de acionamento de potência elétrica estão no Anexo B. A seguir, descrevemos as interfaces de acionamento de potência desenvolvidas.

### **3.2.1 - Descrição das Interfaces**

A função destas interfaces é realizar a conversão, dos sinais de controle de chaveamento provenientes do controlador, através da interface de entrada e saída de dados paralela e da interface geradora de PWM, na forma de um ou dois bits de controle, na potência elétrica que se deseja enviar aos atuadores.

Estas interfaces compatibilizam as características eletro-eletrônicas e lógicas do sinal de controle proveniente do gerador de PWM, digital de um ou dois bits, com as características elétricas dos atuadores, que podem ser do tipo DC bidirecionais ou unidirecionais, fornecendo à estas cargas um sinal de potência chaveado no tempo a uma alta frequência (relativamente aos tempos de resposta dos atuadores), resultando num valor médio conforme desejado pelo controlador.

A interface de acionamento de potência bidirecional é alimentada por uma fonte de tensões simétrica, cuja tensão deve ser menor ou igual a tensão nominal de operação do atuador ou carga, para que, à plena potência, não haja dano ao atuador, enquanto a interface unidirecional é alimentada por uma fonte de tensão simples.

A interface de acionamento de potência bidirecional pode enviar a carga (um motor DC, por exemplo) de 0 a 100% de potência elétrica num sentido de rotação ou no outro, com a precisão do sinal que a controla (estando limitada em sua frequência de chaveamento máxima por seus transistores de chaveamento e por seus fotoacopladores de entrada), necessitando para isso, de dois sinais para acionamento de seus transistores, enquanto a interface unidirecional pode enviar a carga (uma servo-válvula, por exemplo) uma potência elétrica de 0 a 100% apenas num sentido, e necessitando para isso, de apenas um sinal de acionamento.

Estas interfaces tem as seguintes características de aplicação:

Seus sinais de controle, sob a forma de um ou dois bits, são compatíveis com as saídas da interface geradora de PWM, e suas saídas (comum e positiva/negativa) têm a mesma especificação da fonte de alimentação, quanto a tensão máxima fornecida a carga, porém, as mesmas podem ser controlada.

A alimentação necessária se dá através de um conector para uma fonte externa, que é a única alimentação necessária as interfaces, visto que os sinais de controle do gerador PWM são isolados na entrada através do uso fotoacopladores integrados, que são alimentados pela mesma fonte externa.

A sua arquitetura consiste de um ou dois transistores de potência chaveados pelos sinais de controle do gerador PWM, isolados eletricamente por fotoacopladores (o sinal de controle liga ou desliga um diodo emissor de luz, luz esta que incide na junção semicondutora de um fototransistor controlando a sua condutividade, caracterizando o comportamento de uma



chave), resistores são utilizados para regular as correntes de acionamento dos transistores de potência, em função das tensões de alimentação em uso.

As interfaces foram projetadas para possibilitarem uma ampla gama de potências da carga acionada e tensões de acionamento, bastando-se, para isso selecionar um dos transistores da família TIP com configuração interna em par Darlington, em embalagem padronizada TO-220; que suporte estas especificações de chaveamento na frequência de operação desejada.

Os sinais de chaveamento na interface bidirecional devem ser mutuamente exclusivos, visto que uma situação de condução de ambos os transistores de potência significa um curto circuito na fonte de alimentação, podendo causar danos em um ou em ambos, para isso, pode-se colocar um dispositivo de segurança, como um fusível, para interromper o funcionamento quando ocorrer uma sobrecarga, porém este deve ser especificado com muito cuidado, para que não abra em situações normais de transientes.

Quando utilizando uma interface de acionamento unidirecional, devemos tomar cuidado com a polaridade de algumas cargas, pois dependendo de seu tipo, estas não aceitam tensões negativas, enquanto que nas cargas bidirecionais, basta inverter a polaridade das saídas para inverter os sentidos de rotação horário e anti-horário, com relação ao sinal do controlador (a correspondência do controle positivo e negativo com a tensão de acionamento positiva e negativa).

As interfaces podem ainda ter capacitores de estabilização da tensão de acionamento, para cargas onde se deseje uma menor influência da frequência de chaveamento na tensão de saída.

### 3.3 - Fontes de Alimentação DC e Funções Auxiliares

As interfaces de Funções Auxiliares e Alimentação, são interfaces que devem fornecer a potência elétrica consumida pelos circuitos na forma de uma tensão de + 5 VDC estabilizada, para o padrão TTL dos circuitos integrados utilizados nas outras interfaces do sistema.

Além disto, foi projetada para fornecer um relógio auxiliar, sinal responsável pela taxa de amostragem e processamento da interface codificadora de sinais de encoders, controlada pela frequência deste relógio.

Esta interface pode ser alterada para fornecer mais potência com melhor rendimento na parte de alimentação e para fornecer um relógio de maior frequência do que o utilizado inicialmente com frequência  $FCLK = 500 \text{ kHz}$ , que é suficiente para o sistema mecânico nos quais está sendo utilizada.

### 3.4 - Conclusões

Neste capítulo, foi descrito o módulo de controle e interfaces de acionamento e seus subsistemas e acessórios, a metodologia utilizada na sua implementação, suas características técnicas e de utilização, e as possibilidades de sua aplicação. Mostramos, ainda, a flexibilidade dos subsistemas desenvolvidos, quanto a sua funcionalidade e aplicação.

A seguir, descrevemos o módulo de interface com o usuário, que se interliga com os módulos de monitoramento e supervisão e com o módulo de controle e interfaces de acionamento, mostrando, ainda, a estrutura dos módulos de programação e simulação.

## **Capítulo 4**

### **Módulo de Interface com o Usuário**

O módulo de interface com o usuário e núcleo do sistema computacional é responsável, do ponto de vista global de controle, pelo gerenciamento da entrada, saída e manipulação dos valores das variáveis desejadas para os sistemas mecânicos (provenientes do módulo de programação de trajetórias e tarefas, e simulação) e dos valores observados para estas variáveis (provenientes do módulo de aquisição, monitoramento e supervisão), devendo fornecer a estes módulos, os respectivos valores para processamento, e/ou análise, e/ou eventuais relatórios e/ou arquivamento.

Do ponto de vista do usuário, este módulo é responsável pela interface gráfica amigável com o operador na supervisão e comando do mecanismo robótico, podendo ser configurada de maneira a possibilitar o acesso a variáveis do sistema controlado, a alterações de configurações de hardware, a testes de verificação de funcionamento e manutenção, a relatórios, a arquivos, a situação de periféricos, a entradas de dados, a programação de tarefas, e muitas outras possibilidades.

A interface gráfica possibilita, ainda, a partir de sua estrutura e objetos, a implementação do módulo de programação e simulação de trajetórias e tarefas para o mecanismo robótico em questão, aumentando a eficiência na velocidade de utilização do sistema.

A seguir, descrevemos este módulo, desenvolvido para a implementação do sistema de controle flexível :

#### **4.1 - Interface Gráfica com o Usuário**

Foi desenvolvida uma interface gráfica do tipo de múltiplas janelas, com processamento pseudo concorrente de tarefas, composta por objetos e baseada em eventos.

Esta interface, foi projetada e implementada em linguagem Pascal, com muitas partes em Assembler, para que fosse mais eficiente em termos de tempos de execução, e para permitir o acesso ao sistema operacional e ao hardware, de maneira a possibilitar uma utilização eficiente das interfaces de hardware desenvolvidas.

Esta interface gráfica têm o objetivo de padronizar e facilitar o desenvolvimento de vários tipos de aplicativos de supervisão dos mecanismos a controlar, incluindo o programa de interface com o usuário do robô, que será utilizado pelo computador.

O objetivo principal desta interface com o usuário é aumentar a produtividade e a qualidade na operação do mecanismo à nível de gerenciamento de produção, quando sob cuidados do operador da célula de manufatura ao qual o mecanismo robótico pertence.

A interface desenvolvida assemelha-se a um ambiente Windows provido de um gerador de aplicativos do tipo do Microsoft Visual Basic para Windows, guardando-se as devidas proporções em termos de complexidade e recursos.

Podemos observar os objetos (subsistemas) principais da interface gráfica com o usuário na Figura 4.1, onde temos os objetos e procedimentos em desenvolvimento :

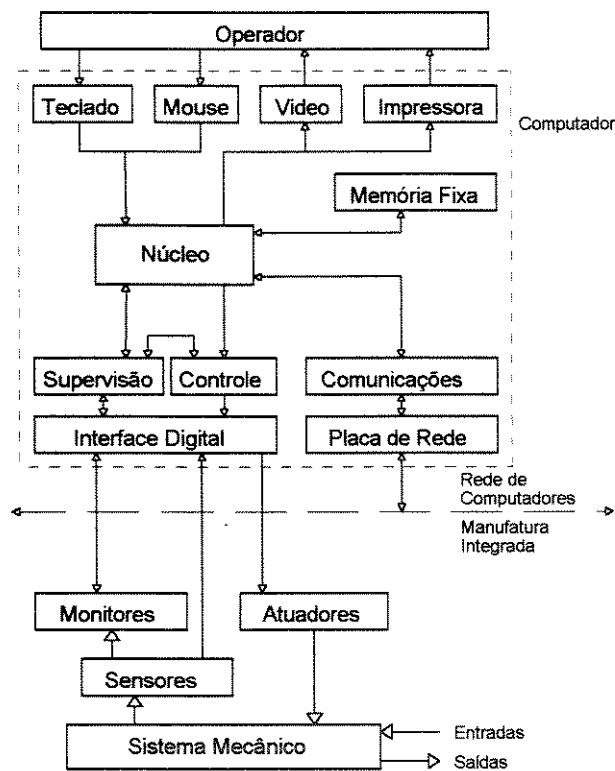


Figura 4.1 - Estrutura da Interface com o Usuário no Sistema Automático

Foram desenvolvidos e encontram-se em funcionamento os seguintes procedimentos e/ou objetos da interface gráfica :

Interface com o teclado :

Desenvolvida em linguagem de máquina, possibilita a utilização plena dos recursos do teclado, não disponíveis nas linguagens de programação comuns.

Interface com o mouse :

Também em linguagem de máquina, possibilita uma melhor utilização dos recursos do mesmo.

Interfaces de comunicações :

Se encontram em desenvolvimento, em linguagem de máquina, pois não são disponíveis nas linguagens comuns.

Interface gráfica para vídeo. Ela é composta de :

- .Janelas, com estrutura funcional e gráfica;
- .Botões de Comando;
- .Campos de texto;
- .Campos para entradas de dados;
- .Campos para exibição de dados;
- .Barras de rolagem;
- .Caixas de listagem.

Controladores de comunicação.

Controlador de impressão.

Controlador de arquivos.

Controlador de bancos de dados.

Procedimentos para controle de objetos.

Procedimentos para acesso a interfaces de hardware.

Procedimentos para implementação de controle.

Outros objetos, em desenvolvimento.

Outros procedimentos, em desenvolvimento.

Os itens em funcionamento, já têm o seu núcleo básico implementado, faltando a implementação das partes opcionais destes itens para que a interface fique completa.

Os itens em desenvolvimento, são aqueles que já possuem definidos o algoritmos básicos para seu funcionamento (alguns tendo sido testados separadamente, facilitando a depuração dos mesmos), porém, não tendo sido implementados na interface.

Na Figura 4.2, podemos ver uma tela gráfica da interface com o usuário juntamente com o gerador de aplicativos "Designer - Main Window", onde as três janelas claras (Debug, VarTypes e Vars) pertencem ao gerador de aplicativos e a janela escura (Main Window) é o aplicativo.

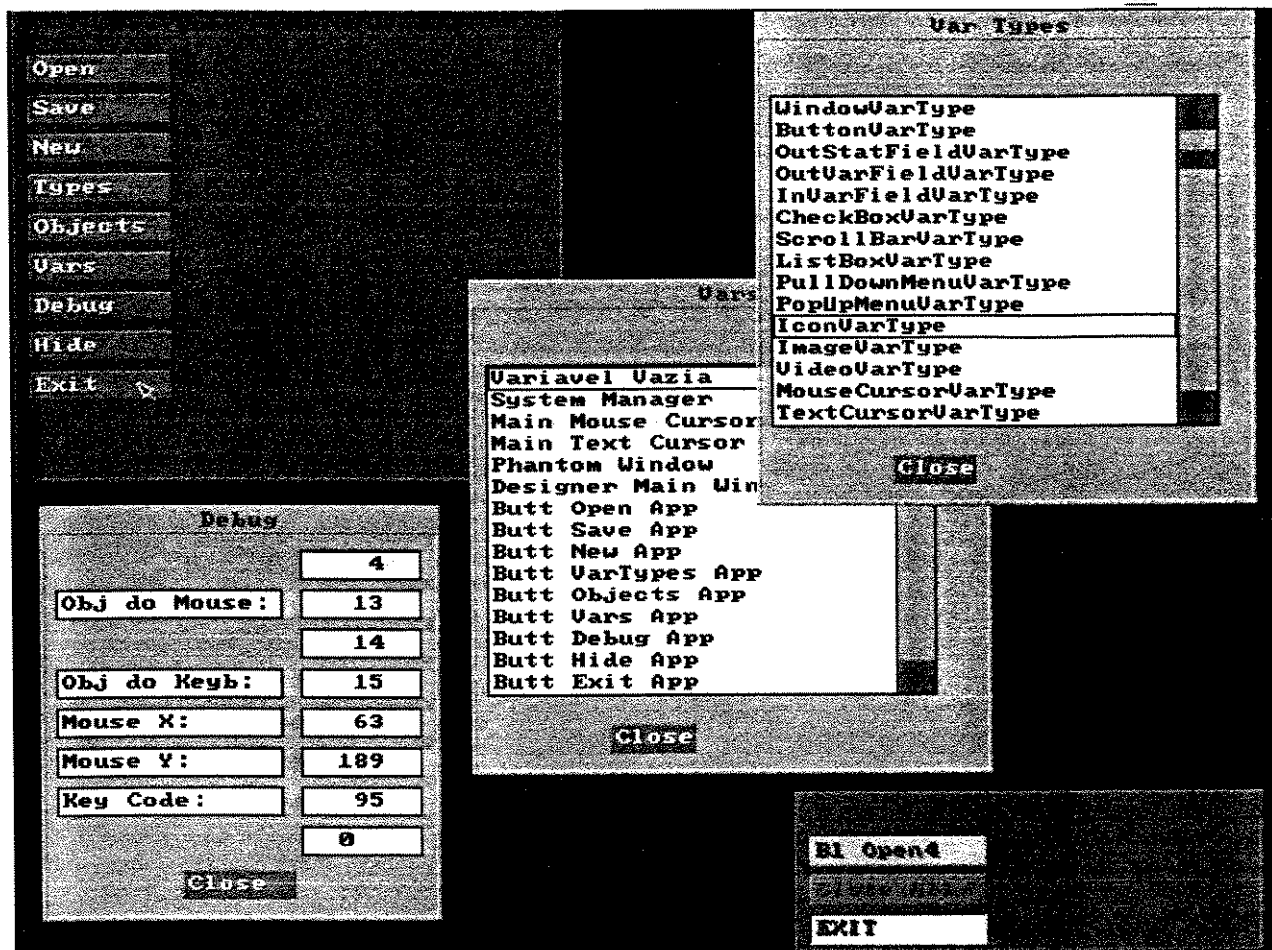


Figura 4.2 - Tela Gráfica da Interface com o Usuário, com o Gerador de Aplicativos

Uma diferença dos geradores de aplicativos convencionais é que os objetos usados pelo aplicativo são idênticos aos utilizados pelo gerador de aplicativos, com a diferença que a programação em pascal é necessária apenas ao programador que desenvolveu a interface, não sendo necessária ao usuário programador que irá gerar um aplicativo para controle de um robô, e sendo totalmente desconhecida do usuário operador do aplicativo de controle do robô.



Podemos ver, ainda, muitos dos objetos desenvolvidos, como por exemplo : o cursor do mouse, o cursor do teclado, janelas, botões, campos de saídas de dados, campos de entrada de dados, listas de dados e barras de controle de posição.

Foram desenvolvidas em linguagem de alto nível e em linguagem de máquina, rotinas e programas de teste das várias funções do controlador, tais como rotinas de inicialização, comunicação com o sistema de aquisição e monitoramento, e comunicação com o outro computador via rede padrão ethernet, e está sendo desenvolvido um programa de testes, implementando um algoritmo de controle e outras rotinas específicas deste algoritmo, no controle de um sistema completo.

#### **4.2 - Módulo de Programação e Simulação**

O módulo de programação e simulação é responsável, do ponto de vista global do sistema de controle, pela geração das seqüências de tarefas e trajetórias desejadas para o mecanismo robótico a ser controlado. Neste módulo, o usuário pode programar as posições geométricas onde o mecanismo robótico deve parar e as trajetórias que este deve percorrer, durante a realização das tarefas.

Isto deve ser feito, no caso de um robô, no espaço cartesiano do ambiente de trabalho do mesmo, para isso, o módulo de programação deve permitir a definição das peças, dos objetos, dos obstáculos, dos "palletes", das ferramentas, e do próprio mecanismo do robô, para que seja possível a simulação gráfica em três dimensões na tela de um computador, via a simulação da operação do mecanismo; garantindo-se que não haverá colisão do robô com os objetos do ambiente de trabalho.

Esse módulo deve ser capaz, ainda, de calcular o modelo Cinemático inverso do mecanismo robótico e de calcular trajetórias interpoladas no espaço cartesiano, para que seja possível a geração das informações de entrada do módulo de controle e acionamento.

Ele, deve ainda, poder ser programado em linguagem de alto nível, seguindo algum padrão (como por exemplo AML, ou outra), ou possibilitar a conversão de programas entre estas linguagens e uma linguagem própria.

Além disso, este módulo, não se encontra incluso no escopo deste trabalho, não tendo sido desenvolvido devido seu conteúdo extenso e complexo, porém, a interface gráfica desenvolvida tem toda a infra-estrutura necessária à sua implementação, tendo sido projetada de acordo com a necessidade do futuro acoplamento com o módulo de programação e simulação.

Como exemplo disso, temos, a necessidade de arquivamento de tarefas já programadas e que devem ser repetidas na execução de outros lotes da mesma peça, isto poderá ser feito através dos procedimentos de leitura e escrita de arquivos de dados, utilizados para se gravar, criar, alterar e recuperar tarefas e trajetórias para o mecanismo, bem como as descrições do ambiente de trabalho em questão.

### **4.3 - Conclusões**

Neste capítulo, apresentamos o módulo de interface com o usuário e seus subsistemas e acessórios, suas características técnicas e de utilização, e as possibilidades de sua aplicação. Mostramos, ainda, a flexibilidade destes subsistemas, quanto a sua utilização e aplicação.

Apresentamos, também, a estrutura, a funcionalidade e as características que os módulos de programação e simulação deverão ter para se interligar aos outros módulos desenvolvidos, mantendo a modularidade e flexibilidade proposta para o sistema.

No capítulo V, descrevemos os testes realizados e resultados experimentais obtidos para as interfaces e programas desenvolvidos, nos quais se validou o funcionamento dos mesmos, de acordo com as especificações descritas.

## **Capítulo 5**

### **Testes e Resultados Experimentais**

Na obtenção dos subsistemas em hardware, todas as interfaces obtidas foram projetadas, construídas e testadas, inicialmente, através de protótipos, e posteriormente, em sua forma final de utilização; sendo que, a partir dos protótipos, puderam ser obtidas interfaces mais bem projetadas em termos de utilização dos recursos dos circuitos integrados, nestas empregados; e em termos de custos destas, devido a redução de seu tamanho físico, e puderam ser, ainda, reparadas e/ou otimizadas várias das funções destas interfaces, visando seu funcionamento correto e aprimorado.

Foram realizados vários testes eletrônicos para se verificar o correto funcionamento do sistema de controle, na ligação dos vários subsistemas que serão utilizados, além disso, efetuou-se a validação dos softwares desenvolvidos durante este trabalho.

Além disso, foram desenvolvidas em linguagem de alto nível e de máquina, rotinas e programas para teste das várias funções do controlador, tais como rotinas de inicialização, comunicação com o sistema de aquisição e monitoramento, e comunicação com o outro computador via rede padrão ethernet, e está sendo desenvolvido um programa de testes, implementando um algoritmo de controle e outras rotinas específicas deste algoritmo, no controle de um sistema mecânico completo.

Atualmente os protótipos desenvolvidos estão sendo implementados em sistemas mecânicos disponíveis no laboratório de automação e robótica da Faculdade de Engenharia

Mecânica, possibilitando alguns testes de controle em sistemas de uso industrial, como por exemplo a bancada de simulação de um grau de liberdade de um robô e um sistema de posicionamento de câmera de vídeo.

### **5.1 - Testes e Validações de Hardware**

Para as interfaces de hardware projetadas, desenvolvidas e construídas, foram realizados testes de validação do seu funcionamento correto, através de uma aplicação prática simplificada, ou, de algum tipo de circuito que simula-se a aplicação desejada, para que fosse possível a verificação do completo atendimento das especificações iniciais, tendo sido desenvolvidos softwares específicos para testes dos mesmos, quando possível, para se tornar os testes mais realísticos do ponto de vista prático de sua utilização.

A interface decodificadora de sinais de encoder foi submetida aos sinais de um circuito gerador de pulsos, que, simulando os sinais defasados de encoders incrementais, possibilitou o seu teste, através da análise dos sinais decodificados num osciloscópio, desde frequências de rotação baixíssimas, até o limite máximo permissível constatado de operação desta interface; como este circuito não possibilita o teste da interface decodificadora em frequências de operação próximas de zero, utilizou-se um encoder incremental real para este fim, constatando-se o correto funcionamento e precisão esperados, concluindo-se que esta interface não tem restrição quanto a frequência mínima de operação e nem quanto a inversão no sentido de rotação do encoder.

A interface de contadores foi testada utilizando-se o esquema de testes da interface decodificadora, sendo que esta forneceu os sinais para a interface contadora em testes, variou-se novamente a velocidade de rotação do encoder (ou similar), verificando-se a correta operação das interfaces contadoras, concluindo-se que estas têm um limite de operação segura de 3 MHz no sinal gerado pelo encoder (muito superior a qualquer aplicação normal).

A interface de acesso multiplexado foi testada, inicialmente, utilizando-se as interfaces de contadores testadas, e simulando-se uma interface de controle, simulando-se as operações de escrita, controle, acesso e leitura, verificando-se o seu correto funcionamento, em seguida, testou-se a interface de acesso, ligada ao computador através da interface digital, controlada por um programa de testes, verificando-se seu correto funcionamento e obtendo-se o limite de operação segura de 1 MHz para a interface de acesso (também, muito superior ao necessário).

A interface digital foi testada conectada a um computador, através da observação de seu comportamento via osciloscópio, em seguida, um software de testes para esta interface testou a integridade da saída e da entrada de dados, via escrita de todos os valores possíveis nas portas de saída e posterior leitura e comparação pelas portas de entrada, confirmando-se a operação correta para todas as frequências de operação obtidas a partir do programa e dos computadores utilizados.

A interface geradora de PWM foi testada analisando-se todos os sinais PWM gerados, no osciloscópio, para todos os valores de entrada e nas várias configurações possíveis, inicialmente, com sinais de entrada estáticos e em seguida com sinais de entrada gerados pelo computador, através da interface digital, verificando-se o seu correto funcionamento.

As interfaces de acionamento de potência, foram testadas utilizando-se a interface geradora de PWM como fonte de controle de seus sinais, e acionando um motor DC como carga, variando-se os valores de tensão desejados e observando-se seu correto funcionamento na tensão média gerada na carga, com uma pequena e esperada diferença no valor médio, devido aos opto-isoladores utilizados na proteção dos circuitos lógicos dos circuitos de potência.

As fontes de alimentação desenvolvidas são fontes comuns com saídas reguladas para alimentação de circuitos lógicos, ou fontes não reguladas de potência para acionamento de atuadores.

O encoder ótico incremental desenvolvido não pode ser considerado definitivo, devido a ajustes nos tempos de chaveamento de seus transistores, isto limita a frequência de operação e consequentemente a rotação máxima de operação. O corpo do encoder desenvolvido não foi definido ainda, devido a ampla gama de possíveis invólucros, devendo-se definir um tipo de fixação e acoplamentos padrões.

## **5.2 - Testes e Validações de Software**

Os softwares desenvolvidos, subsistemas do módulo de interface com usuário, foram testados durante seu desenvolvimento, mediante a colocação destes em funcionamento a cada passo de sua produção, e a cada novo recurso implementado, conforme a especificação desejada; isto garantiu, através da sua operação real, um correto desenvolvimento e funcionamento destes subsistemas, além disso, a abordagem de projeto de software modular e reutilizável, e com características voltadas a objetos, facilitou muito a sua implementação.

Esta abordagem foi testada com sucesso no desenvolvimento do programa para geração de dados de programação da memória PROM da interface geradora de PWM, neste programa, foram desenvolvidos vários protótipos dos objetos e procedimentos que seriam utilizados no módulo de interface com o usuário, como por exemplo a interface com o mouse, a interface com o teclado, a interface com o sistema de arquivos, as janelas gráficas simplificadas; tendo-se concluído que estes objetos foram quase que integralmente reutilizados na interface com o usuário, mediante alguns aprimoramentos inicialmente previstos.

Um dos sistemas mecânicos testados, foi o sistema de posicionamento de uma câmera de vídeo, como mostra a figura 5.1, onde temos, uma plataforma rotativa (sobre a qual está fixa a câmera de vídeo) com relação a um pedestal fixo, sendo que, na parte inferior da plataforma se encontra um encoder ótico incremental que fornece a posição da câmera (após a inicialização dos contadores, quando se obtêm a referência de posicionamento), a partir dos sinais do encoder, decodificados, contados e adquiridos pelo sistema de controle, o programa de supervisão, controle e interface com o usuário, gera os sinais de acionamento para a interface de geração de PWM, que controla a interface de acionamento de potência, fornecendo potência elétrica ao motor DC acoplado

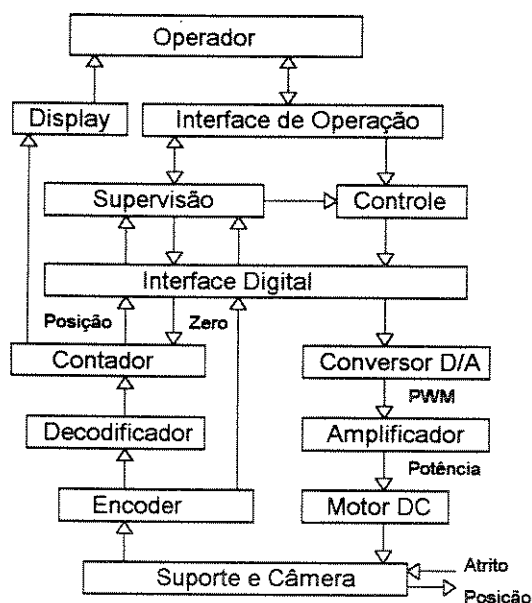


Figura 5.1 - Sistema de Controle de Posicionamento de uma Câmera

a base da plataforma, girando-a a partir de uma ordem do operador, via teclado, para posições pré-programadas pelo usuário, via aprendizado, ou via programação do número de pulsos, ou, ainda, via programação da posição medida em graus, com relação ao zero de referência.



## 5.4 - Controle de um Robô Industrial

Como fase final dos testes, teremos a ligação do sistema de controle a um robô industrial, como visto na figura I.4, porém, devido ao custo do mecanismo, devem-se tomar as precauções necessárias para sua realização, devendo-se para tanto, realizar todos os testes possíveis em sistemas mecânicos mais simples, antes da aplicação no robô.

Para tanto, estamos realizando, inicialmente, a ligação do sistema de controle, instalado em um computador, limitado aos módulos de entrada e de interface com o usuário, com um mecanismo robótico; este mecanismo será o robô Manutec r3, o qual deseja-se, monitorar e supervisionar, durante sua operação normal, antes de se tentar controlá-lo.

Também com esse objetivo, estamos utilizando uma bancada dinâmica para os testes de hardware e de controle; nesta bancada, composta por um Motor DC - Embreagem - Freio - Inércia Variável, se pretende simular uma Junta do Robô Manutec r3, já que neste mecanismo robótico, os motores de acionamento das juntas se encontram sempre em rotação à velocidade nominal, e quando se deseja movimentar uma junta o seu freio é relaxado e a embreagem é acionada transferindo-se energia mecânica do motor para junta, ao passo que o freio é acionado e a embreagem solta quando se deseja uma posição estática para a junta.

Para esta bancada, tem-se como objetivo, ainda, a implementação de um sistema completo de controle, utilizando-se o sistema desenvolvido, possibilitando a comparação de seu desempenho real com seu desempenho previsto via simulação, resultados obtidos com auxílio de softwares de simulação de sistemas não lineares e discretos.

Neste teste, um computador extra será utilizado para monitorar a atuação do controlador, isto, para que seja possível uma verificação independente do funcionamento do

sistema construído; além disso, esta bancada tem sido utilizada nos testes práticos dos <sup>60</sup> subsistemas desenvolvidos.

## **5.5 - Conclusões**

Neste capítulo, apresentamos os testes realizados e resultados experimentais obtidos para as interfaces e programas desenvolvidos.

No próximo capítulo, apresentamos as conclusões dos trabalhos realizados e as perspectivas para a sequência dos mesmos, no aprimoramento e na complementação do sistema desenvolvido, com relação ao desenvolvimento de outras interfaces, subsistemas da interface com o usuário e os módulos de programação e simulação, além do desenvolvimento das estratégias e dos algoritmos de controle.

## **Capítulo 6**

### **Conclusões e Perspectivas**

O objetivo fundamental desse trabalho foi obter um controlador digital programável e flexível de alto desempenho e baixo custo, capaz de atender aos requisitos necessários à implementação dos diversos algoritmos de controle propostos para manipuladores e robôs industriais, tais como controladores PID, de torque calculado, não lineares, adaptativos e outros.

O sistema de controle desenvolvido atende às características e aos requisitos inicialmente propostos de performance, de flexibilidade e de custos, tendo sido, assim, atingidos os objetivos propostos para o trabalho.

O controlador obtido servirá essencialmente como ferramenta para o estudo e resolução dos problemas de controle dos sistemas mecânicos do tipo não lineares, encontrados nos manipuladores e robôs industriais, onde o controlador tem se mostrado um item fundamental no desempenho do sistema robótico [20], onde desejamos atingir um melhor desempenho (com o aumento de produtividade destes sistemas) à um menor custo (substituindo-se um controlador dedicado de alto custo, por um controlador flexível de baixo custo).

Além disso, o sistema de controle já pode ser utilizado em sistemas mais simples, como os que já foram testados.

Inicialmente o método de implementação do controlador, seria a construção de um controlador baseado em um microprocessador RISC (Reduced Instruction Set Computer) [20,

21, 23, 24, 25, 27, 28], que ficaria residente no interior de um computador padrão IBM/AT<sup>62</sup> (comunicando-se com este através do barramento padrão ISA), este método foi alterado para um controlador implementado em um computador padrão IBM/AT (quando necessário, comunicando-se com outro computador que pode servir como supervisor de tarefas e programação, através de uma interface de comunicações do tipo serial, ou do tipo de redes locais Ethernet).

Esta alteração, teve como motivo uma grande dificuldade encontrada no fornecimento e suporte técnico dos componentes que seriam utilizados no controlador, bem como no suporte a nível de software para o microprocessador, assim, optou-se pela conclusão do trabalho através de outro método de implementação do controlador, um método com vantagens e desvantagens.

Uma desvantagem do controlador implementado em outro computador separado do computador de interface com o usuário e programação, é que, a velocidade de comunicação entre estes passa a ser um fator relevante no problema, visto que numa implementação do controlador diretamente no barramento do computador supervisor, permitiria uma velocidade de comunicação muitas vezes superior a necessária aos dados e comandos a serem trocados entre estes.

Passamos a ter como possíveis interfaces de comunicação, as interfaces padrão RS-232C seriais, cujo custo é muito baixo, cuja taxa de transferência não passa da centena de Kbytes por segundo em distâncias da ordem de metros; ou as interfaces Ethernet padrão IEEE 802.3, usadas em redes locais de computadores, cujo custo é relativamente baixo e cuja taxa de transferência é da ordem de 800 Kbytes por segundo em distâncias da ordem de uma centena de metros. Foram testadas estas duas possibilidades com sucesso, sendo preferível a utilização da interface de rede para a maioria das aplicações.

Uma vantagem do novo método de implementação do sistema de controle, é que, devido ao fato de que controlador e supervisor podem estar separados e ligados por uma conexão de comunicação, o supervisor pode estar numa sala de operações mais apropriada às tarefas de programação e supervisão, que lhe cabem neste sistema robótico, fora do ambiente de chão de fábrica, além disso, sendo utilizada uma implementação de interface de comunicações através de interfaces de rede Ethernet, o supervisor, que é fundamentalmente um programa (relativamente independente do computador onde estão instalados os hardwares específicos), pode ser utilizado à priori em qualquer computador da rede local, além de possibilitar a integração da célula de manufatura do robô com o restante da rede de automação da fábrica.

Outra desvantagem, é que o custo do controlador dedicado que seria construído, seria pouco menor quando de uma produção seriada, do que o custo de um controlador implementado num computador independente, que necessita de muitos subsistemas auxiliares para seu funcionamento.

Porém, uma implementação do controlador independente, em um computador padrão, torna possível a escolha de um computador para este controlador com recursos e desempenho na medida do necessário, para os programas de controle que se estiver utilizando, podendo-se escolher desde um computador baseado no microprocessador 80286 (tipo o IBM AT) com clock de 12MHz até um microprocessador 80486 ou Pentium (da mesma família, mais moderno e veloz que o 80486) com clock de 100MHz ou mais, apenas função do desenvolvimento da indústria de computadores pessoais, com melhoria no desempenho da ordem de até dezenas de vezes (porém, acompanhado de aumentos no custo do computador).

A solução utilizando-se uma implementação do controlador em um computador padrão, continua visando a utilização de um computador altamente difundido e com muitos recursos de programação, comunicação e desempenho, além disso, acessível à um custo relativamente

baixo, deste modo, nesta implementação serão utilizados no supervisor e também no controlador, sendo que o supervisor ficará responsável pela geração de trajetórias, pela interface gráfica com o usuário, e por outras tarefas de gerenciamento do robô; ficando o controlador responsável pela interface com o ambiente de trabalho do robô (neste estarão ligados o sistema de aquisição e monitoramento de dados de encoders, os atuadores eletromecânicos e os sensores da célula de manufatura do robô), e ficando responsável também pelo controle do robô, ou seja, pelos cálculos dos algoritmos de controle do robô; desta forma, são aproveitados no desenvolvimento do trabalho, os recursos de hardware dos computadores existentes no laboratório.

Com relação aos algoritmos à implementar no controlador, as disciplinas e a revisão bibliográfica, já forneceram várias possibilidades de interesse, obtidas na análise de vários tipos de estratégias de controle, e resultados experimentais de controladores utilizando alguns destes algoritmos, que quando implementados em controladores suficientemente velozes conseguem aumentar o desempenho dos sistemas robóticos e/ou aumentar a robustez do controle, porém, à custos não tão baixos, todavia sendo ótimos indicadores do sucesso destes algoritmos e controladores.

Como plataforma para implementação do controlador, teremos a oportunidade de testar os vários algoritmos de controle, em computadores de várias capacidades de processamento, sendo o mais veloz um IBM AT compatível com processador Intel 80486DX com clock de 50MHz, 8 Mbytes de memória RAM e cache de memória de 256Kbytes.

Consideramos que os estudos aprofundados realizados na área de microprocessadores, barramentos, componentes auxiliares, funcionamento e construção de computadores CISC e principalmente RISC, foram muito importantes no que se refere a utilização, projeto e construção de interfaces de interface, interfaces para aquisição de dados, interfaces de acionamento, sistemas auxiliares, programas para controle automático, programas para testes

de hardware e programas para interface com o usuário; tendo sido muito importantes para este projeto de pesquisa e constituindo base sólida para muitos outros projetos na área.

Como perspectivas de pesquisa e desenvolvimento do sistema de controle, temos, a possibilidade de produção de outros tipos de hardware para uso específico com sensores e atuadores ainda não explorados, como por exemplo, conversores de sinais analógicos para sinais digitais (para conversão de grandezas contínuas como, por exemplo, : temperatura, pressão, torque, força, posição (potenciômetros e resistores variáveis), velocidades, acelerações e etc.), tacômetros digitais (aplicação na medição de velocidade e rotação de máquinas e objetos do ambiente de trabalho), conversores de sinais digitais industriais para aquisição como entradas digitais (conversão de 110VAC, 220VAC, 4-20mA e outros, para digitais TTL de 5 VDC).

Como exemplos de hardwares para aplicação com atuadores, temos: conversores D/A para controle chaveado de motores de corrente alternada trifásicos (possibilitando controle de velocidade e torque em motores de baixo custo e fácil manutenção), controladores para motores de passo (para aplicação em posicionamento de sistemas auxiliares como: máquinas de usinagem, esteiras de movimentação, e outros), conversores digitais para relês e chaves industriais (com isolamento opto-eletrônica, para utilização com painéis e equipamentos industriais), e outros atuadores.

Com relação ao desenvolvimento de software, temos amplas perspectivas de aprimoramento, complementação e conclusão dos objetos e procedimentos iniciados e planejados, tendo ainda, todo o módulo de programação e simulação por ser desenvolvido, para que seja possível a sua utilização, padronizada, pelo usuário final do sistema, visto que até o momento os softwares de interface com o usuário foram desenvolvidos especificamente para os sistemas mecânicos testados, não sendo possível, ainda, a programação através de linguagens padrões da robótica.

## Anexos

Nos Anexos, apresentamos dados utilizados, resultados de estudos teóricos e resultados práticos, com o lay-out das placas de hardware desenvolvidos; um resumo da teoria de modelamento dinâmico e controle de robôs, um resumo da teoria utilizada no desenvolvimento dos estudos teóricos realizados, e base para os modelos obtidos.



## Anexo A - Modelamento Dinâmico de Robôs

Neste anexo, fazemos um resumo da teoria utilizada no modelamento de robôs e manipuladores, para que se tornem mais claros os algoritmos pesquisados, os programas desenvolvidos e a abordagem dada ao problema de controle de robôs. Este resumo foi baseado na referência bibliográfica : Craig, J.J., " Introduction to Robotics: Mechanics & Control ", além de conter vários tópicos adicionais.

### A.I - Introdução

Devido a sua complexidade e à sua natureza espacial e multivariável, o problema de controle de robôs e manipuladores industriais requer uma abordagem mais profunda e detalhada, desde o modelamento até o controle, passando pela necessidade de um modelo cinemático direto, um modelo cinemático inverso, um modelo dinâmico, um modelo para a estrutura do controle, e uma interface amigável com o usuário, isto para que sua incorporação numa linha produtiva, se de com a maior eficiência que seu sistema eletro-mecânico possa oferecer.

### A.II - Descrição Espacial

Em robótica, assim como na dinâmica dos sólidos espaciais, é necessária uma representação do espaço tridimensional e dos objetos neste localizados ( robô, ferramentas, peças, obstáculos, etc. ), quanto à sua posição e orientação. a representação destas grandezas, aqui utilizada, é baseada em Sistemas de Coordenadas Ortonormais ( Cartesianas ).

Todas as grandezas são definidas em relação à algum sistema de coordenadas cartesianas, sendo adotado um Sistema de Coordenadas Universal, em relação ao qual tudo pode ser referenciado direta ou indiretamente, através da definição da posição e da orientação de um sistema de coordenadas em relação a outro.

#### Descrição de Posição :

Uma posição no espaço, medida sempre em relação a um sistema de coordenadas, é definida por um vetor de posição  $3 \times 1$ , com início na origem deste sistema de coordenadas :

$$\mathbf{r}_{P/A} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$$

Onde temos a descrição da posição  $P$  em relação ao sistema de coordenadas  $A$ , através de suas componentes escalares  $p_x$ ,  $p_y$  e  $p_z$  no vetor de posição do ponto  $P$ , resultantes da decomposição deste sobre os versores  $i$ ,  $j$  e  $k$  do sistema de coordenadas, em relação ao qual esta posição está sendo medida, ou seja, em relação ao sistema de coordenadas  $A$ .

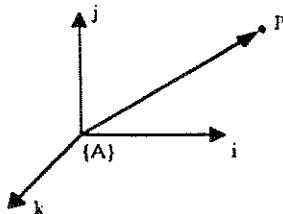


Figura A-II.1.

Na figura II.1 vemos a representação da posição  $P$  ( equivalentemente pode-se pensar em  $P$  como um ponto no espaço ), medida em relação ao sistema de coordenadas  $A$ ,  $r_{P/A}$ .

### Descrição de Orientação :

Um objeto no espaço é constituído por primitivas, cuja definição se dá através de pontos que as descrevem, por exemplo : um cilindro pode ser definido por dois pontos de centro ( vetores de início e fim da linha de centro ) e um raio ( vetor perpendicular a linha de centros ).

Assim, os objetos aqui descritos são definidos por pontos fixos em relação a um sistema de coordenadas ( aqui chamados, genericamente, de sistema de coordenadas do objeto ), que pode ser fixo ou móvel em relação ao sistema de coordenadas universal.

Portanto, a orientação de um objeto no espaço, se dá através da definição da orientação do sistema de coordenadas fixo a este objeto.

Uma maneira conveniente de se descrever a orientação de um sistema de coordenadas é através da descrição de seus três versores  $i$ ,  $j$  e  $k$ , agrupados numa matriz  $3 \times 3$  ( comunmente chamada de matriz de mudança de bases ou matriz de rotação, ou ainda, matriz de orientação ):

$$\begin{bmatrix} i & j & k \end{bmatrix}_{B/A} = \begin{bmatrix} r_{i/B} & r_{j/B} & r_{k/B} \end{bmatrix} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}$$

Onde temos a descrição da orientação do sistema de coordenadas  $B$  em relação ao sistema de coordenadas  $A$ , através de seus versores  $i$ ,  $j$  e  $k$  na matriz de rotação, resultantes da decomposição destes sobre os versores  $i$ ,  $j$  e  $k$  do sistema de coordenadas, em relação ao qual esta orientação está sendo medida, ou seja, em relação ao sistema de coordenadas  $A$ .

Onde  $i_x$ ,  $i_y$  e  $i_z$ ;  $j_x$ ,  $j_y$  e  $j_z$ ; e  $k_x$ ,  $k_y$  e  $k_z$  são as componentes ( escalares ) dos versores  $i$ ,  $j$  e  $k$  do sistema de coordenadas  $B$ , decompostos sobre os versores  $i$ ,  $j$  e  $k$  do sistema de coordenadas  $A$ .

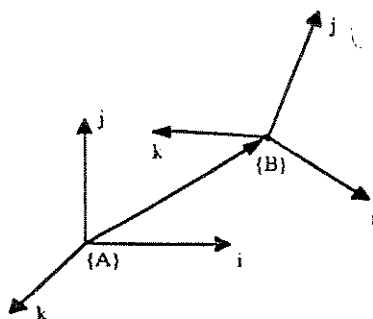


Figura A-II.2.

Na figura II.2 vemos a representação do sistema de coordenadas B, medido em relação ao sistema de coordenadas A,  $[i \ j \ k]_{B/A}$ .

### Descrição do Objeto :

Assim, a descrição completa da localização de um objeto no espaço se dá através da descrição da posição e da orientação do sistema de coordenadas fixo a este, através de um vetor de posição e de uma matriz de orientação :

$$\mathbf{r}_{B/A}, [i \ j \ k]_{B/A}.$$

Onde temos a descrição da localização do sistema de coordenadas B, medido em relação a A, através do vetor de posição da origem do sistema de coordenadas B e de sua matriz de rotação, como mostra a figura II.2.

### Sistemas de Coordenadas e Representação Estendida :

Quando utilizamos vários sistemas de coordenadas na descrição dos objetos e grandezas físicas dos sistemas robóticos, e desejamos realizar operações aritméticas utilizando estas descrições, é necessário que estas grandezas estejam sendo expressas num mesmo sistema de coordenadas.

Neste momento, observamos claramente a necessidade de uma representação mais completa das grandezas físicas, assim, é adotada a seguinte representação estendida, mostrando em qual sistema de coordenadas esta grandeza está sendo expressa :

$$\mathbf{r}_{P/A} \Rightarrow \mathbf{r}_{P/A/A} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_{/A}.$$

Onde temos a descrição do vetor de posição P, medida em relação ao sistema de coordenadas A, e expressa no sistema de coordenadas A, cujas componentes escalares assumem os valores  $p_x$ ,  $p_y$  e  $p_z$ .

Para que isto fique mais claro, observemos a figura II.3 :

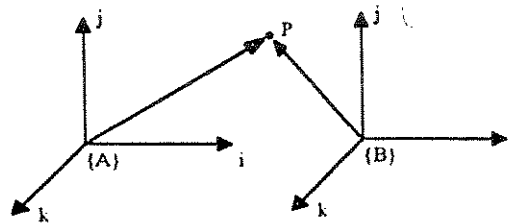


Figura A-II.3.

Na figura II.3 vemos a representação da posição P, medida em relação ao sistema de coordenadas A, o vetor de posição  $\mathbf{r}_{P/A}$ , e medida em relação ao sistema de coordenadas B, o vetor de posição  $\mathbf{r}_{P/B}$ , vetores diferentes, apesar de localizarem o mesmo ponto no espaço, além disso, podemos expressar estes dois vetores, tanto no sistema de coordenadas A, como no sistema B :

$$\mathbf{r}_{P/A} \Rightarrow \mathbf{r}_{P/A/A}, \mathbf{r}_{P/A/B}, \text{ e }, \mathbf{r}_{P/B} \Rightarrow \mathbf{r}_{P/B/A}, \mathbf{r}_{P/B/B}$$

Neste caso, como os dois sistemas de coordenadas têm a mesma orientação, diferindo apenas por uma translação, a representação dos vetores  $\mathbf{r}_{P/A}$  e  $\mathbf{r}_{P/B}$  num ou no outro sistema de coordenadas é numericamente igual, ou seja,  $\mathbf{r}_{P/A/A} = \mathbf{r}_{P/A/B}$ , e,  $\mathbf{r}_{P/B/A} = \mathbf{r}_{P/B/B}$ , isto fica claro se pensarmos que :

$$\mathbf{r}_{P/A/B} = \mathbf{r}_{P/B/B} - \mathbf{r}_{A/B/B}$$

A posição de P medida em relação a A é um vetor com início em A e fim em P.

Porém, nos casos onde os sistemas de coordenadas se encontram desalinhados, é necessária uma mudança de sistemas de coordenadas onde as grandezas estão expressas, como estudamos a seguir.

### Mudanças de Sistemas de Coordenadas :

Quando utilizamos vários sistemas de coordenadas na descrição dos objetos e grandezas físicas dos sistemas robóticos, e desejamos realizar operações aritméticas utilizando estas descrições, é necessário que estas grandezas estejam sendo expressas num mesmo sistema de coordenadas. Assim, devemos saber como mudar o sistema de coordenadas no qual esta grandeza está sendo expressa.

Por exemplo, na figura II.4, desejamos descrever a posição P, medida em relação ao sistema de coordenadas A, vetor  $\mathbf{r}_{P/A}$ , sabendo-se sua posição medida em relação a B, vetor  $\mathbf{r}_{P/B}$ , e sabendo-se a posição de B medida em relação a A, vetor  $\mathbf{r}_{B/A}$  e matriz de orientação

$$[i \ j \ k]_{B/A}.$$

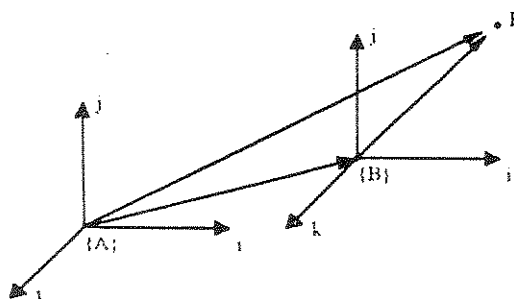


Figura A-II.4.

Podemos obter  $\mathbf{r}_{P/A}$ , simplesmente por uma soma de dois vetores :

$$\mathbf{r}_{P/A} = \mathbf{r}_{P/B} + \mathbf{r}_{B/A}.$$

Porém, esta equação, não diz em que sistemas de coordenadas os vetores estão expressos, assim, para a sua realização aritmética, devemos utilizar a representação estendida :

$$\mathbf{r}_{P/A/A} = \mathbf{r}_{P/B/A} + \mathbf{r}_{B/A/A}.$$

Com todos os vetores expressos num mesmo sistema de coordenadas, mas as grandezas conhecidas, utilizando-se a representação estendida, são :  $\mathbf{r}_{P/B/B}$ ,  $\mathbf{r}_{B/A/A}$  e  $[i \ j \ k]_{B/A/A}$ .

Novamente, no caso, onde A e B têm a mesma orientação, há apenas uma translação entre os dois sistemas de coordenadas, e  $\mathbf{r}_{P/B/A} = \mathbf{r}_{P/B/B}$ , daí :

$$\mathbf{r}_{P/A/A} = \mathbf{r}_{P/B/A} + \mathbf{r}_{B/A/A} = \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A}.$$

Porém, no caso onde A e B não têm a mesma orientação, como na figura II.5, temos que  $\mathbf{r}_{P/B/A} \neq \mathbf{r}_{P/B/B}$ , e conseqüentemente, devemos utilizar a matriz de orientação para obter  $\mathbf{r}_{P/B/A}$  a partir de  $\mathbf{r}_{P/B/B}$ .

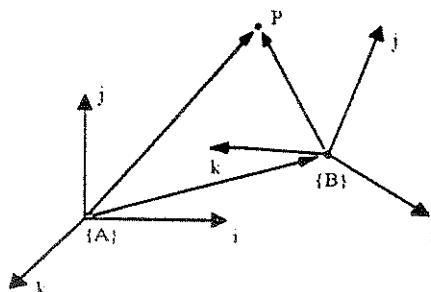


Figura A-II.5.

Podemos obter  $\mathbf{r}_{P/B/A}$ , a partir de  $\mathbf{r}_{P/B/B}$ , se escrevermos ( da algebra linear ).  
alternativamente à representação aqui utilizada, que :

$$\mathbf{r}_{P/B/B} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_{/B} = p_x \cdot \mathbf{r}_{i/B/B} + p_y \cdot \mathbf{r}_{j/B/B} + p_z \cdot \mathbf{r}_{k/B/B}.$$

Lembrando que, o vetor  $\mathbf{r}_{P/B/B}$  é formado pela soma de suas componentes nas direções dos versores  $i, j$  e  $k$ , do sistema de coordenadas B, onde estes valores estão sendo expressos. Mas os versores do sistema de coordenadas B,  $\mathbf{r}_{i/B/B}$ ,  $\mathbf{r}_{j/B/B}$  e  $\mathbf{r}_{k/B/B}$ , são conhecidos da definição de matriz de orientação de B em relação ao sistema A :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \mathbf{r}_{i/B/A} & \mathbf{r}_{j/B/A} & \mathbf{r}_{k/B/A} \end{bmatrix} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}_{/A},$$

Assim, podemos escrever :

$$\mathbf{r}_{P/B/B} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}_{/B} \Rightarrow \mathbf{r}_{P/B/A} = p_x \cdot \mathbf{r}_{i/B/A} + p_y \cdot \mathbf{r}_{j/B/A} + p_z \cdot \mathbf{r}_{k/B/A} =$$

$$\mathbf{r}_{P/B/A} = p_x \cdot \begin{bmatrix} i_x \\ i_y \\ i_z \end{bmatrix}_{/A} + p_y \cdot \begin{bmatrix} j_x \\ j_y \\ j_z \end{bmatrix}_{/A} + p_z \cdot \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}_{/A} = \begin{bmatrix} p_x \cdot i_x + p_y \cdot j_x + p_z \cdot k_x \\ p_x \cdot i_y + p_y \cdot j_y + p_z \cdot k_y \\ p_x \cdot i_z + p_y \cdot j_z + p_z \cdot k_z \end{bmatrix}_{/A} =$$

$$\mathbf{r}_{P/B/A} = [i \ j \ k]_{B/A/A} \cdot \mathbf{r}_{P/B/B}$$

Chegamos à operação necessária para mudar o sistema de coordenadas onde está representada uma grandeza, através da utilização da matriz de orientação, agora podemos escrever a equação :

$$\mathbf{r}_{P/A} = \mathbf{r}_{P/B} + \mathbf{r}_{B/A}$$

Em sua forma completa :

$$\mathbf{r}_{P/A/A} = \mathbf{r}_{P/B/A} + \mathbf{r}_{B/A/A} = [i \ j \ k]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A}$$

Levando em consideração a translação do sistema de coordenadas B com relação a A e levando em consideração a rotação de B em relação a A, cabe notar que, se A e B têm a mesma orientação a matriz de rotação se torna a matriz identidade, pois os versores do sistema de coordenadas B estão alinhados com os do sistema de coordenadas A, e, consequentemente,  $\mathbf{r}_{P/B/A} = \mathbf{r}_{P/B/B}$ , simplificando a equação para apenas uma translação, como mostrado anteriormente.

Deve-se notar a diferença que existe entre as mudanças de sistemas de coordenadas, que são de dois tipos bastante distintos, numa mudança de sistemas de coordenadas, deseja-se saber o valor de uma grandeza medida em relação a outro sistema de coordenadas; na outra, deseja-se

saber o valor de uma grandeza expressa em um outro sistema de coordenadas, podemos deixar isto claro, com o exemplo a seguir, onde conhecemos  $\mathbf{r}_{P/B/B}$ , e podemos obter :

$$\mathbf{r}_{P/B/A} = [\mathbf{i} \ \mathbf{j} \ \mathbf{k}]_{B/A/A} \cdot \mathbf{r}_{P/B/B}, \text{ e}$$

$$\mathbf{r}_{P/A/A} = \mathbf{r}_{P/B/A} + \mathbf{r}_{B/A/A} = [\mathbf{i} \ \mathbf{j} \ \mathbf{k}]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A}.$$

Grandezas completamente diferentes,  $\mathbf{r}_{P/B/A}$  é a posição P medida em relação ao sistema de coordenadas B ( vetor com início na origem de B e fim no ponto P ), expressa no sistema de coordenadas A; enquanto,  $\mathbf{r}_{P/A/A}$  é a posição P medida em relação ao sistema de coordenadas A, e expressa em A, obtidas por métodos distintos.

Como exemplo, vejamos o caso planar da figura II-6, onde temos uma translação e uma rotação entre os sistemas de coordenadas B e A :

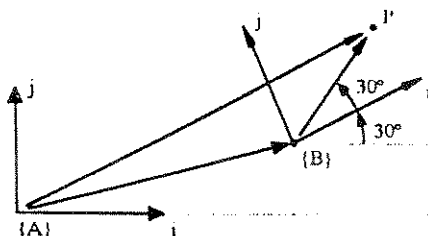


Figura A-II.6.

Sendo conhecidos :

$$\mathbf{r}_{P/B/B} = \begin{bmatrix} 0.866 \\ 0.500 \\ 0 \end{bmatrix}_{/B}, \quad \mathbf{r}_{B/A/A} = \begin{bmatrix} 2.0 \\ 0.5 \\ 0 \end{bmatrix}_{/A}, \quad [\mathbf{i} \ \mathbf{j} \ \mathbf{k}]_{B/A/A} = \begin{bmatrix} 0.866 & -0.500 & 0 \\ 0.500 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/A}.$$

Podemos obter a posição de P medida em relação a A expressa em A,  $\mathbf{r}_{P/A/A}$  :

$$\begin{aligned} \mathbf{r}_{P/A/A} &= \mathbf{r}_{P/B/A} + \mathbf{r}_{B/A/A} = [\mathbf{i} \ \mathbf{j} \ \mathbf{k}]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A} = \\ &= \begin{bmatrix} 0.866 & -0.500 & 0 \\ 0.500 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/A} \cdot \begin{bmatrix} 0.866 \\ 0.500 \\ 0 \end{bmatrix}_{/B} + \begin{bmatrix} 2.0 \\ 0.5 \\ 0 \end{bmatrix}_{/A} = \begin{bmatrix} 0.750 - 0.250 - 2.0 \\ 0.433 + 0.433 + 0.5 \\ 0 + 0 \end{bmatrix}_{/A} = \begin{bmatrix} 2.500 \\ 1.366 \\ 0 \end{bmatrix}_{/A} \end{aligned}$$

Este resultado pode ser conferido facilmente utilizando-se a trigonometria.

### Aritmética das Mudanças de Sistemas de Coordenadas :

Na prática, em robótica, são utilizados vários sistemas de coordenadas para definir a localização das muitas peças componentes do robô e de seu ambiente de trabalho, neste cenário, muitos sistemas de coordenadas são definidos em relação a outros numa sequência lógica que facilite a sua utilização, por exemplo, no caso do robô, cada elemento de sua cadeia cinemática tem sua localização definida com relação ao elemento anterior, e o primeiro com relação a base deste.

Sendo assim, para se obter uma grandeza expressa num sistema de coordenadas, muitas vezes são necessárias mudanças sucessivas de sistemas de coordenadas, por exemplo, quando desejamos saber a localização da garra do robô medida em relação ao sistema de coordenadas da base deste, devemos realizar sempre a mesma sequência de mudanças de sistemas de coordenadas do último elemento até a base; portanto, é muito útil obter uma transformação composta, que forneça a mudança de sistemas de coordenadas diretamente.

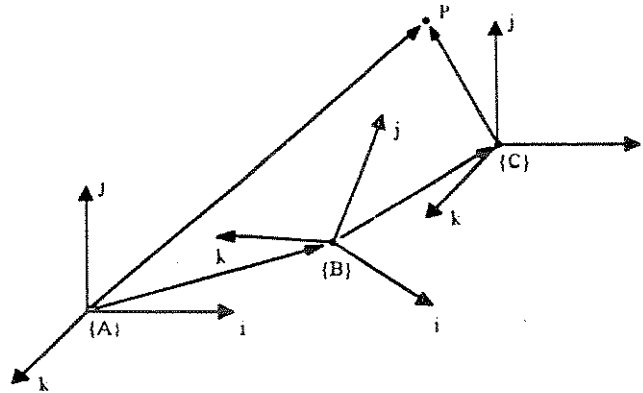


Figura A-II.7.

Podemos obter uma mudança de sistemas de coordenadas composta de C para A, como mostra a figura II-7, pois sabemos que :

$$\mathbf{r}_{P/B/B} = [i \ j \ k]_{C/B/B} \cdot \mathbf{r}_{P/C/C} + \mathbf{r}_{C/B/B}, \text{ e, } \mathbf{r}_{P/A/A} = [i \ j \ k]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A}.$$

E, substituindo-se  $\mathbf{r}_{P/B/B}$  na equação de  $\mathbf{r}_{P/A/A}$ , obtemos :

$$\mathbf{r}_{P/A/A} = [i \ j \ k]_{B/A/A} \cdot [i \ j \ k]_{C/B/B} \cdot \mathbf{r}_{P/C/C} + [i \ j \ k]_{B/A/A} \cdot \mathbf{r}_{C/B/B} + \mathbf{r}_{B/A/A}.$$

Assim, concluímos que, como esperado, a orientação composta é resultado da multiplicação das rotações sucessivas, e que, a translação composta é resultado da soma das translações sucessivas :

$$[i \ j \ k]_{C/A/A} = [i \ j \ k]_{B/A/A} \cdot [i \ j \ k]_{C/B/B}, \quad \mathbf{r}_{C/A/A} = [i \ j \ k]_{B/A/A} \cdot \mathbf{r}_{C/B/B} + \mathbf{r}_{B/A/A}.$$



Alternativamente, podemos obter este mesmo resultado, escrevendo :

$$\mathbf{r}_{P/A} = \mathbf{r}_{P/C} + \mathbf{r}_{C/B} + \mathbf{r}_{B/A} ,$$

E, mudando o sistema de coordenadas onde cada elemento está expresso, para o sistema de coordenadas A, através das rotações necessárias.

Outra situação comum, é a necessidade de se mudar de sistemas de coordenadas na ordem inversa da definição destes, ou seja, na figura II-5 podemos desejar saber  $\mathbf{r}_{P/B/B}$ , conhecendo-se  $\mathbf{r}_{P/A/A}$ ,  $\mathbf{r}_{B/A/A}$  e  $[i j k]_{B/A/A}$ , mas nesta situação, a equação anteriormente utilizada, continua sendo válida, e podemos manipulá-la :

$$\mathbf{r}_{P/A/A} = [i j k]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + \mathbf{r}_{B/A/A} \Rightarrow$$

$$[i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{P/A/A} = [i j k]_{B/A/A}^{-1} \cdot [i j k]_{B/A/A} \cdot \mathbf{r}_{P/B/B} + [i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{B/A/A} \Rightarrow$$

$$[i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{P/A/A} = \mathbf{r}_{P/B/B} + [i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{B/A/A} \Rightarrow$$

$$\mathbf{r}_{P/B/B} = [i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{P/A/A} - [i j k]_{B/A/A}^{-1} \cdot \mathbf{r}_{B/A/A}$$

Mas, podemos comparar esta equação com a equação a seguir, que deve fornecer o mesmo resultado :

$$\mathbf{r}_{P/B/B} = [i j k]_{A/B/B} \cdot \mathbf{r}_{P/A/A} + \mathbf{r}_{A/B/B} ,$$

Obtemos, então, as relações de mudança de coordenadas na ordem inversa :

$$[i j k]_{A/B/B} = [i j k]_{B/A/A}^{-1} , \text{ e } \mathbf{r}_{A/B/B} = -[i j k]_{A/B/B} \cdot \mathbf{r}_{B/A/A}$$

Mas, da algebra linear, a inversa de uma matriz de versores ortonormais é igual à sua transposta, e então, obtemos :

$$[i j k]_{A/B/B} = [i j k]_{B/A/A}^T , \text{ e } \mathbf{r}_{A/B/B} = -[i j k]_{A/B/B} \cdot \mathbf{r}_{B/A/A}$$

Onde, podemos interpretar que, a matriz de orientação do sistema de coordenadas A medida em relação a B é a inversa ( igual à transposta ) da matriz de orientação do sistema de coordenadas B medida em relação a A, além disso, a translação de A em relação a B é, como podemos ver na figura II-5, o vetor de translação de B em relação a A, negado e expresso em B, através da rotação de A em relação a B, pela multiplicação da matriz de orientação invertida.

### Considerações Sobre Orientações de Sistemas de Coordenadas :

Ficou claro que, a representação de orientação de sistemas de coordenadas adotada neste trabalho, teve o objetivo de simplificar muito a aritmética anteriormente desenvolvida, através da descrição dos três versores  $i$ ,  $j$  e  $k$  de um sistema de coordenadas, agrupados numa matriz  $3 \times 3$ ; porém, esta abordagem não é interessante do ponto de vista do usuário.

Podemos fazer algumas simplificações, baseados nas propriedades dos três versores da matriz de orientação, pois, sendo ortonormais, os produtos internos destes versores dois a dois devem ser nulos, já que estes são perpendiculares dois a dois, e o módulo de cada um deve ser igual a 1 :

$$i \cdot j = 0, \quad j \cdot k = 0, \quad k \cdot i = 0, \quad |i| = 1, \quad |j| = 1, \quad |k| = 1$$

Vemos que, a matriz de orientação, composta por nove termos, apresenta seis equações de dependências entre estes termos, assim conclui-se que a orientação de um corpo, ou sistema de coordenadas, apresenta apenas três graus de liberdade, e não nove.

Sendo assim, apresentamos algumas formas alternativas de representar uma orientação especificando apenas três parâmetros, o que simplifica bastante a tarefa de se programar, por exemplo, a orientação da garra do robô em suas tarefas, ao invés de se ter que montar uma matriz com nove elementos, sendo as colunas ortonormais.

Lembrando que devemos tomar muito cuidado com a visualização espacial de uma orientação, pois, as rotações geralmente não são comutativas, ou seja, a ordem nas mudanças de orientação não pode ser alterada; já que utilizamos matrizes para implementá-las, e estas não são comutativas.

### Z - Y - X Angulos de Euler :

Uma representação possível pode ser obtida pelos chamados ângulos de Euler, onde a orientação de um sistema de coordenadas é obtida através de três rotações simples sucessivas, em torno dos eixos Z, Y e X, nesta ordem.

Podemos obter a matriz de orientação do sistema de coordenadas B em relação a A, se introduzirmos as três rotações dos ângulos de Euler, através de dois sistemas de coordenadas fictícios auxiliares, como mostra a figura II-8 :

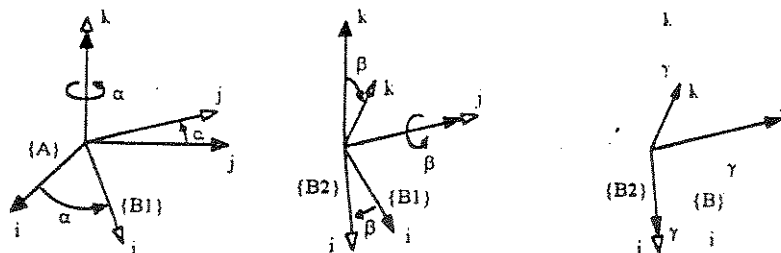


Figura A-II.8.

Onde, a matriz de orientação final de B para A é dada por :

$$[i \ j \ k]_{B/A} = [i \ j \ k]_{B1/A} \cdot [i \ j \ k]_{B2/B1} \cdot [i \ j \ k]_{B/B2}$$

E, podemos escrever as matrizes das rotações intermediárias em função dos respectivos ângulos de rotação  $\alpha$ ,  $\beta$  e  $\gamma$  :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/A} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}_{/B1} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}_{/B2} =$$

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos\alpha \cdot \cos\beta & \cos\alpha \cdot \sin\beta \cdot \sin\gamma - \sin\alpha \cdot \cos\gamma & \cos\alpha \cdot \sin\beta \cdot \cos\gamma + \sin\alpha \cdot \sin\gamma \\ \sin\alpha \cdot \cos\beta & \sin\alpha \cdot \sin\beta \cdot \sin\gamma + \cos\alpha \cdot \cos\gamma & \sin\alpha \cdot \sin\beta \cdot \cos\gamma - \cos\alpha \cdot \sin\gamma \\ -\sin\beta & \cos\beta \cdot \sin\gamma & \cos\beta \cdot \cos\gamma \end{bmatrix}_{/A}$$

Onde,

Tendo obtido a matriz de orientação em função dos ângulos de Euler, podemos obter os ângulos de rotação  $\alpha$ ,  $\beta$  e  $\gamma$  para qualquer matriz de orientação genérica :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}_{/A}$$

Comparando-se os elementos desta, com os elementos da matriz anterior, obtemos :

$$\alpha = \text{Atan2}(i_y, i_x),$$

$$\beta = \text{Atan2}(-i_z, \sqrt{i_x^2 + i_y^2}),$$

$$\gamma = \text{Atan2}(j_z, k_z),$$

Onde,  $\text{Atan2}(a, b)$ , é o arco tangente de  $a$  dividido por  $b$ , porém, resultando um ângulo entre  $0^\circ$  e  $360^\circ$ , levando em consideração os sinais dos argumentos para determinar o quadrante em que se encontra o resultado. além disso,  $\text{Atan2}(a, 0)$  não deve ser considerado um erro, e sim retornar  $90^\circ$ , ou  $270^\circ$ , conforme o sinal de  $a$ .

Não entraremos na discussão das singularidades desta solução, que deve ser bastante analisada, quando de sua utilização prática.

Analogamente aos ângulos de Euler, podemos definir outras representações através de rotações sucessivas, como por exemplo, um sistema bastante comum, chamado Roll, Pitch e Yaw, obtidos através de rotações em torno dos eixos X, Y e Z, nesta ordem.

#### Ângulos Z - X - Y :

Porém, preferimos uma outra representação, que facilite a utilização pelo operador e que facilite a sua definição em desenhos de tarefas no ambiente de trabalho do robô: esta representação pode ser obtida através de rotações em torno dos eixos Z, X e Y, nesta ordem, e poderia ser chamada de Ângulos de Planta, Elevação e Torsão (sendo que, esta nomenclatura será evitada, já que, pode ser confundida com outras grandezas), como mostra a figura II.9 :

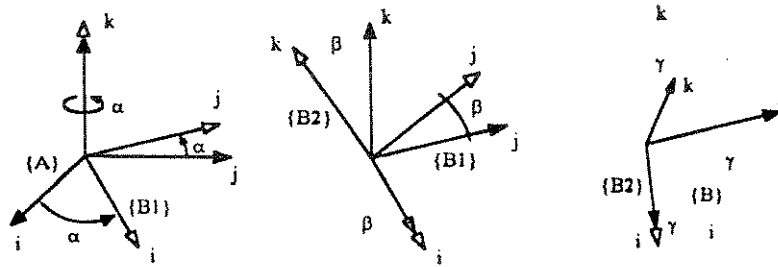


Figura A-II.9.

Na figura II.10, podemos ver a utilização prática da representação pelos ângulos Z, X e Y, na qual,  $\alpha$  pode ser observado diretamente, numa vista da planta do ambiente de trabalho, como sendo o ângulo entre a projeção do vetor  $j$  do sistema B no plano horizontal e o vetor  $j$  do sistema A;  $\beta$  pode ser observado numa vista de perfil, como sendo o ângulo entre o vetor  $j$  do sistema B e o plano horizontal; e  $\gamma$  pode ser observado numa vista perpendicular ao eixo Y, como sendo o ângulo de rotação do sistema B sobre seu próprio vetor  $j$ :

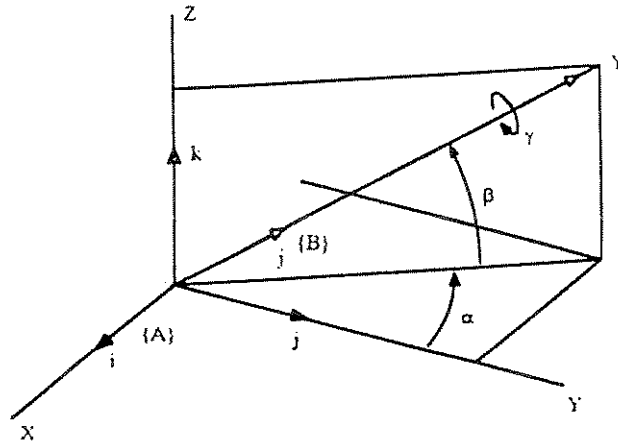


Figura A-II.10.

Nota-se, que esta representação é basicamente apoiada no vetor  $j$  do sistema de coordenadas B, que em se tratando da garra do robô, é comumente posicionado na direção longitudinal desta, facilitando a utilização pelo programador.

A matriz de orientação final do sistema de coordenadas B em relação ao sistema de coordenadas A, pode ser obtida em função dos respectivos ângulos de rotação  $\alpha$ ,  $\beta$  e  $\gamma$ :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/A} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}_{/B1} \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix}_{/B2} =$$

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos\alpha \cdot \cos\gamma - \sin\alpha \cdot \sin\beta \cdot \sin\gamma & -\sin\alpha \cdot \cos\beta & \cos\alpha \cdot \sin\gamma + \sin\alpha \cdot \sin\beta \cdot \cos\gamma \\ \sin\alpha \cdot \cos\gamma + \cos\alpha \cdot \sin\beta \cdot \sin\gamma & \cos\alpha \cdot \cos\beta & \sin\alpha \cdot \sin\gamma - \cos\alpha \cdot \sin\beta \cdot \cos\gamma \\ -\cos\beta \cdot \sin\gamma & \sin\beta & \cos\beta \cdot \cos\gamma \end{bmatrix}_{/A}$$

Tendo obtido a matriz de orientação em função dos ângulos da representação Z, X e Y, podemos obter os ângulos de rotação  $\alpha$ ,  $\beta$  e  $\gamma$  para qualquer matriz de orientação genérica:

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix}_{/A}, \text{ obtemos :}$$

$$\alpha = \text{Atan2}(-j_x, j_y),$$

$$\beta = \text{Atan2}(j_z, j_y/\cos\alpha),$$

$$\gamma = \text{Atan2}(-i_z, k_z),$$

Porém, nossa solução não é válida para  $\beta = 90^\circ$  e nem para  $\beta = 270^\circ$ , pois nestes casos a rotação em torno de X faz com que o versor j se alinhe com o versor k, portanto a rotação em torno de Y se torna redundante a rotação em torno de Z, mostrando que apenas uma destas rotações é necessária para se descrever esta orientação.

Isto pode ser observado na matriz de rotação, que se torna reduzida, e pode ser simplificada :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos\alpha \cdot \cos\gamma \mp \sin\alpha \cdot \sin\gamma & 0 & \cos\alpha \cdot \sin\gamma \pm \sin\alpha \cdot \cos\gamma \\ \sin\alpha \cdot \cos\gamma \pm \cos\alpha \cdot \sin\gamma & 0 & \sin\alpha \cdot \sin\gamma \mp \cos\alpha \cdot \cos\gamma \\ 0 & \pm 1 & 0 \end{bmatrix}_{/A} =$$

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} \cos(\alpha \pm \gamma) & 0 & \pm \sin(\alpha \pm \gamma) \\ \sin(\alpha \pm \gamma) & 0 & \mp \cos(\alpha \pm \gamma) \\ 0 & \pm 1 & 0 \end{bmatrix}_{/A}, \text{ para } \beta = 90^\circ \text{ e } \beta = 270^\circ.$$

Neste caso, devido a redundância entre  $\alpha$  e  $\gamma$ , arbitramos  $\gamma = 0^\circ$ , e podemos obter os ângulos de rotação  $\alpha$  e  $\beta$  para esta matriz de orientação :

$$[i \ j \ k]_{B/A/A} = \begin{bmatrix} i_x & 0 & k_x \\ i_y & 0 & k_y \\ 0 & j_z & 0 \end{bmatrix}_{/A}, \text{ para } \beta = 90^\circ \text{ e } \beta = 270^\circ, \text{ e obtemos :}$$

$$\alpha = \text{Atan2}(i_y, i_x),$$

$$\beta = \text{Arcsin}(j_z),$$

$$\gamma = 0^\circ.$$

Outra representação possível, mas de pouca utilidade prática, pode ser obtida através de uma única rotação em torno de uma determinada direção no espaço, de uma quantidade angular conhecida, comumente chamada de Equivalente Ângulo - Eixo, onde se obtém o equivalente de três rotações em torno dos versores em uma única rotação em torno de uma direção equivalente. Não discutiremos esta representação, pois não a utilizamos neste trabalho.

Numa implementação de uma interface com o operador, o sistema de controle do robô pode permitir que o mesmo, possa escolher entre diversos tipos de forma de representação de orientação, visto que todas podem ser convertidas na matriz de rotação equivalente.

### A.III - Cinemática de Manipuladores

Para lidarmos com o mecanismo de um robô ( chamado aqui de braço do robô, ou de manipulador, para que se evidencie a ausência dos sistemas de controle, programação, e etc. ), o consideraremos como um cadeia cinemática composta por elos teóricamente rígidos ligados por juntas, sendo que cada elo é sustentado pelo anterior, e têm algum tipo de movimento em relação a este na junta; o primeiro estando fixo a base do manipulador, e o último podendo ser sua garra, podendo lidar com objetos e realizar tarefas.

Afixamos a cada elo da cadeia cinemática do robô, um sistema de coordenadas, para que esta esteja totalmente localizada no espaço, e descrevemos cada um destes sistemas de coordenadas em relação ao anterior.

No estudo cinemático do robô, o objetivo principal é saber a localização de sua da garra, medida em relação ao sistema de coordenadas da base, em função das posições relativas dos sistemas de coordenadas afixados, que são conhecidas em função das variáveis das juntas do manipulador; isto, a medida em que este se movimenta, através da articulação de suas juntas. Este é chamado de Modelo Cinemático Direto do robô, onde se obtém a posição do manipulador no espaço cartesiano, em função das variáveis das juntas.

Normalmente, cada elo do manipulador tem um grau de liberdade em relação ao anterior, podendo este ser implementado através de uma junta revoluta ou prismática, nos casos de juntas com mais de um grau de liberdade, podemos modelá-la como várias juntas de um grau de liberdade cada, assim, desenvolvemos aqui a teoria para modelar juntas com um grau de liberdade cada.

Os elos da cadeia cinemática do robô devem receber nomes, que podem ser números de 0 a n. por exemplo, começando pela base fixa do mesmo, e indo até o enésimo elo, a extremidade livre que desejamos posicionar, então, o elo número 1 é o primeiro elemento móvel do manipulador, e assim por diante. Para um manipulador conseguir posicionar sua garra genericamente no espaço, são necessárias, no mínimo, seis juntas, pois sua localização possui seis graus de liberdade, podendo ser obtida a partir de três parâmetros de posição e três parâmetros de orientação, como foi visto anteriormente.

Manipuladores típicos têm 5 ou 6 juntas, porém qualquer número é teoricamente possível. além disso, podem surgir outros tipos de estruturas cinemáticas nos elos do manipulador, como paralelogramos e outros, porém, estes podem ser modelados matematicamente como vínculos de algum tipo entre as variáveis das juntas.

Do ponto de vista do equacionamento cinemático, só são necessários os dados das dimensões de um elo, relacionados com a localização da próxima junta, são comuns as simplificações na descrição dos elos, como a notação de Denavit-Hartenberg, porém, as consideramos desnecessárias no contexto deste trabalho, estas simplificações visam apenas um método de representação das dimensões dos elos por parâmetros obtidos de forma padronizada.

Modelamento Cinemático Direto :

Aplicamos a aritmética das mudanças de sistemas de coordenadas, vista anteriormente, na cadeia cinemática, composta pelos  $n$  sistemas de coordenadas, e obtemos a posição e a orientação do sistema de coordenadas da garra, medida em relação à base fixa do robô, como sendo :

$$[i \ j \ k]_{N/0/0} = [i \ j \ k]_{1/0/0} \cdot [i \ j \ k]_{2/1/1} \cdot [i \ j \ k]_{3/2/2} \cdot \dots \cdot [i \ j \ k]_{N/N-1/N-1},$$

$$\mathbf{r}_{N/0/0} = \mathbf{r}_{N/N-1/0} + \dots + \mathbf{r}_{3/2/0} + \mathbf{r}_{2/1/0} + \mathbf{r}_{1/0/0}.$$

Note que, os vetores das translações de um sistema de coordenadas em relação ao anterior estão expressos no sistema de coordenadas da base, para tanto, estes devem sofrer as mudanças de sistemas de onde são conhecidos até a base, ou seja :

$$\begin{aligned} \mathbf{r}_{N/0/0} &= [i \ j \ k]_{1/0/0} \cdot [i \ j \ k]_{2/1/1} \cdot [i \ j \ k]_{3/2/2} \cdot \dots \cdot [i \ j \ k]_{N-1/N-2/N-2} \cdot \mathbf{r}_{N/N-1/N-1} + \\ &+ \dots + \\ &+ [i \ j \ k]_{1/0/0} \cdot [i \ j \ k]_{2/1/1} \cdot \mathbf{r}_{3/2/2} + \\ &+ [i \ j \ k]_{1/0/0} \cdot \mathbf{r}_{2/1/1} + \mathbf{r}_{1/0/0} \end{aligned}$$

Como exemplo, podemos tomar o manipulador da figura III.1 :

Neste manipulador planar de três graus de liberdade, podemos ver os sistemas de coordenadas afixados aos elos, de forma que a representação de um sistema em relação ao anterior seja a mais simples possível, para que isto facilite o seu equacionamento.

Equacionando o manipulador, temos :

$$\mathbf{r}_{4/3/3} = \begin{bmatrix} l_3 \\ 0 \\ 0 \end{bmatrix}_3, \quad [i \ j \ k]_{4/3/3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_3,$$

O sistema de coordenadas 4 representa a garra, estando fixo a esta, e se localiza na extremidade do elo 3, a uma distância  $l_3$  da origem deste, sobre o eixo X, e possui a mesma orientação que o sistema de coordenadas 3, analogamente escrevemos :

$$\mathbf{r}_{3/2/2} = \begin{bmatrix} l_2 \\ 0 \\ 0 \end{bmatrix}_{/2}, \quad [i \ j \ k]_{3/2/2} = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 \\ \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/2},$$

$$\mathbf{r}_{2/1/1} = \begin{bmatrix} l_1 \\ 0 \\ 0 \end{bmatrix}_{/1}, \quad [i \ j \ k]_{2/1/1} = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/1},$$

$$\mathbf{r}_{1/0/0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{/0}, \quad [i \ j \ k]_{1/0/0} = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/0},$$

Após a descrição dos elos do manipulador, basta aplicar as relações anteriores para obtermos :

$$\mathbf{r}_{4/0/0} = \begin{bmatrix} -(\sin\theta_1 \cdot ((l_2 + l_3 \cdot \cos\theta_3) \cdot \sin\theta_2 + l_3 \cdot \cos\theta_2 \cdot \sin\theta_3)) + \\ \cos\theta_1 \cdot ((l_2 + l_3 \cdot \cos\theta_3) \cdot \sin\theta_2 + l_3 \cdot \cos\theta_2 \cdot \sin\theta_3) + \\ 0 \\ + \cos\theta_1 \cdot (l_1 + \cos\theta_2 \cdot (l_2 + l_3 \cdot \cos\theta_3) - l_3 \cdot \sin\theta_2 \cdot \sin\theta_3) \\ + \sin\theta_1 \cdot (l_1 + \cos\theta_2 \cdot (l_2 + l_3 \cdot \cos\theta_3) - l_3 \cdot \sin\theta_2 \cdot \sin\theta_3) \\ 0 \end{bmatrix}_{/0},$$

$$[i \ j \ k]_{4/0/0} = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}_{/0},$$

Não transcrevemos as passagens intermediárias destes cálculos, pois utilizamos o software Mathematica ( um programa com ferramentas de matemática, que trabalha com equações literais ) para obter estes resultados, sendo que na solução obtida, conferimos todos os passos do desenvolvimento realizado, uma vez que o programa é conduzido pelo programador neste desenvolvimento. A listagem deste exemplo constou do anexo do relatório anterior, observando-se as particularidades do programa, com relação à diferente notação, neste utilizada.

Obtemos assim, a posição e a orientação da garra em função dos ângulos das juntas  $\theta_1$ ,  $\theta_2$  e  $\theta_3$ , sendo que estas as variáveis possibilitam os três graus de liberdade do manipulador. neste caso específico, do robô planar, os três graus de liberdade são a posição da garra no plano X-Y e a orientação da garra em relação à base em torno do eixo Z; além disso, influem no modelo cinemático as dimensões dos braços do manipulador  $l_1$ ,  $l_2$ , e  $l_3$ .

Note-se que as dimensões dos braços do manipulador são importantes na definição do volume de trabalho do robô, ou seja, no projeto do manipulador o alcance da garra no espaço do



ambiente de trabalho é o ponto de partida para o projetista, sendo assim, o modelo cinemático direto é uma importante ferramenta na visualização do volume de trabalho do robô.

A noção de modelo cinemático direto, levanta o problema fundamental da robótica, pois os atuadores ( motores, redutores, correias, cilindros hidráulicos, etc. ) atuam sobre as variáveis das juntas, assim, é necessário que o modelo cinemático direto converta os valores destas variáveis, nos valores de posição e orientação da garra no sistema de coordenadas do ambiente de trabalho do robô, para que este possa interagir com os outros objetos ( operador, peças, máquinas, equipamentos, etc. ), portanto, surge o problema de se controlar os atuadores diretamente, porém, atuando indiretamente sobre as juntas e mais indiretamente ainda, sobre a garra no espaço cartesiano do ambiente de trabalho.

Decorre disso, a idéia de localização do manipulador no espaço dos atuadores, no espaço das juntas, e no espaço cartesiano, já que em cada um destes espaços, pode-se, com um conjunto de variáveis distintas em significado e valor, definir completamente a localização do manipulador.

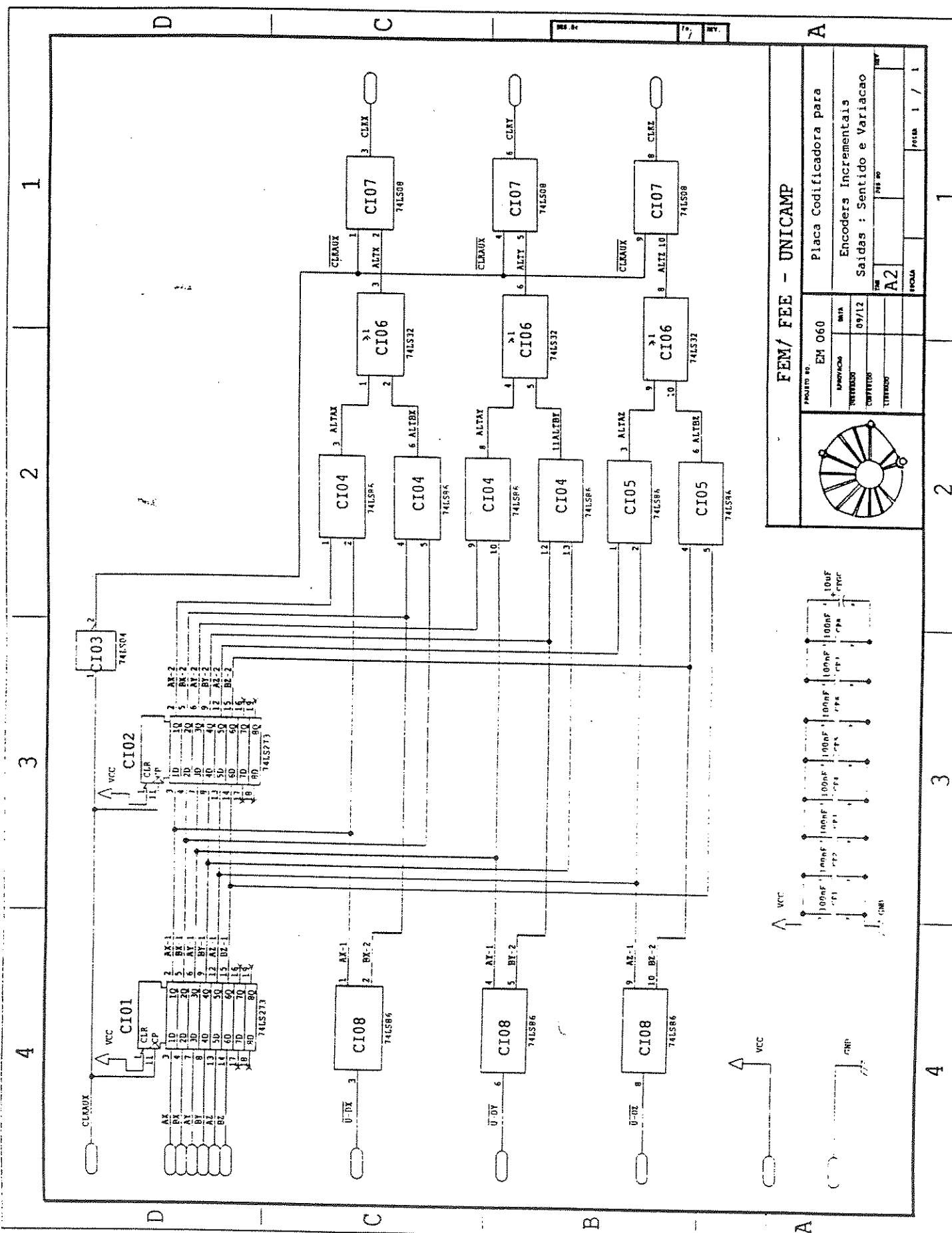
Portanto, o modelo cinemático direto mapeia a localização do manipulador do espaço das juntas para o espaço cartesiano do sistema de coordenadas da base, ou seja, é uma transformação que leva de um vetor de posição das juntas ( no exemplo planar, um vetor  $3 \times 1$ , composto por  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  ) para um vetor posição e uma matriz de orientação da garra ( no exemplo planar, pode-se pensar em um vetor  $3 \times 1$ , composto pelas coordenadas X e Y, e o ângulo de orientação final da garra ).

## A.IV - Cinemática Inversa de Manipuladores

Outra questão que surge, é como podemos obter os valores das variáveis das juntas, necessárias para que a garra do robô esteja em uma localização desejada do espaço cartesiano da base, para que este possa realizar uma tarefa com sua garra, nesta posição.

Este mapeamento é feito pelo Modelo Cinemático Inverso, ou seja, uma transformação que leva uma localização no espaço cartesiano da base para valores das variáveis das juntas correspondente.

Infelizmente, devido às não linearidades do modelo cinemático direto, não podemos simplesmente invertê-lo para obter o modelo cinemático inverso, este é um problema mais difícil, que deve ser solucionado caso a caso.



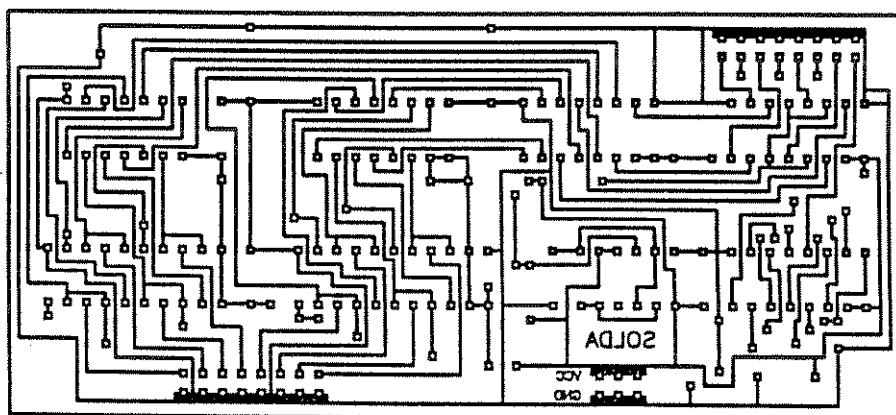


Figura B.I.2 - Lay-Out da Placa de Circuito Impresso da Placa Decodificadora de Sinais de Encoders - Trilhas do Lado da Solda

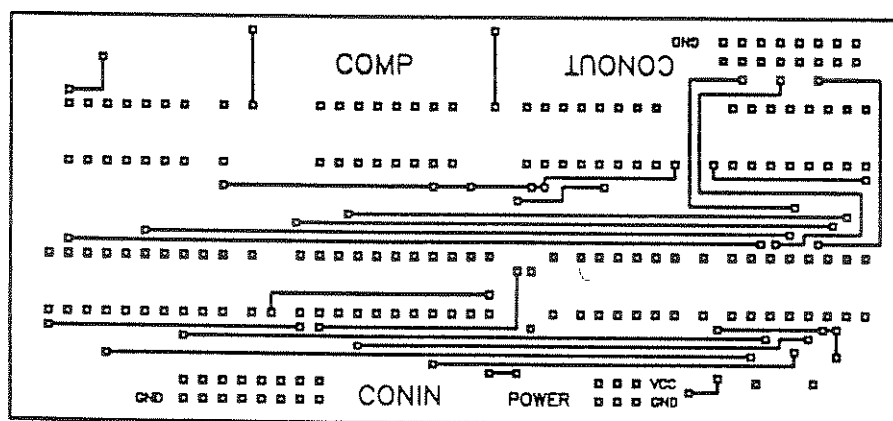


Figura B.I.3 - Lay-Out da Placa de Circuito Impresso da Placa Decodificadora de Sinais de Encoders - Trilhas do Lado dos Componentes

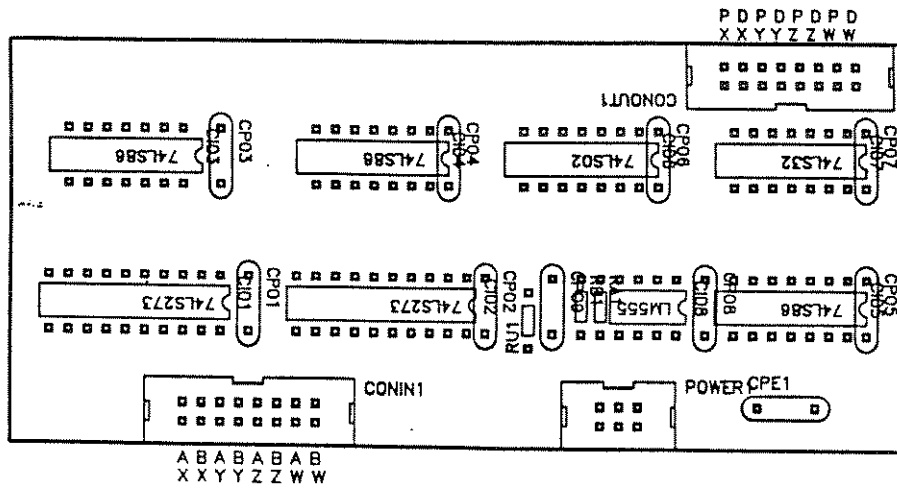


Figura B.L.4 - Lay-Out da Placa de Circuito Impresso da Placa Decodificadora de Sinais de Encoders - Posicionamento dos Componentes



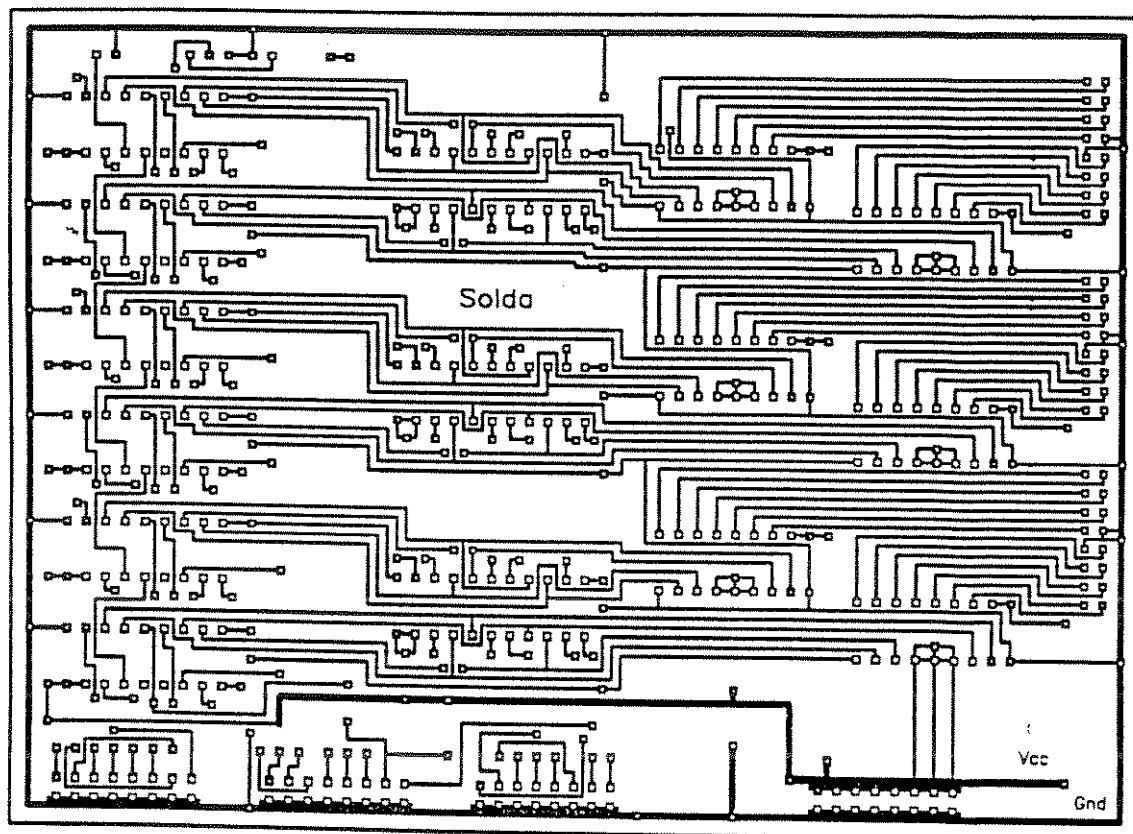


Figura B.II.2 - Lay-Out da Placa de Circuito Impresso da Placa de Contadores Digitais - Trilhas do Lado da Solda

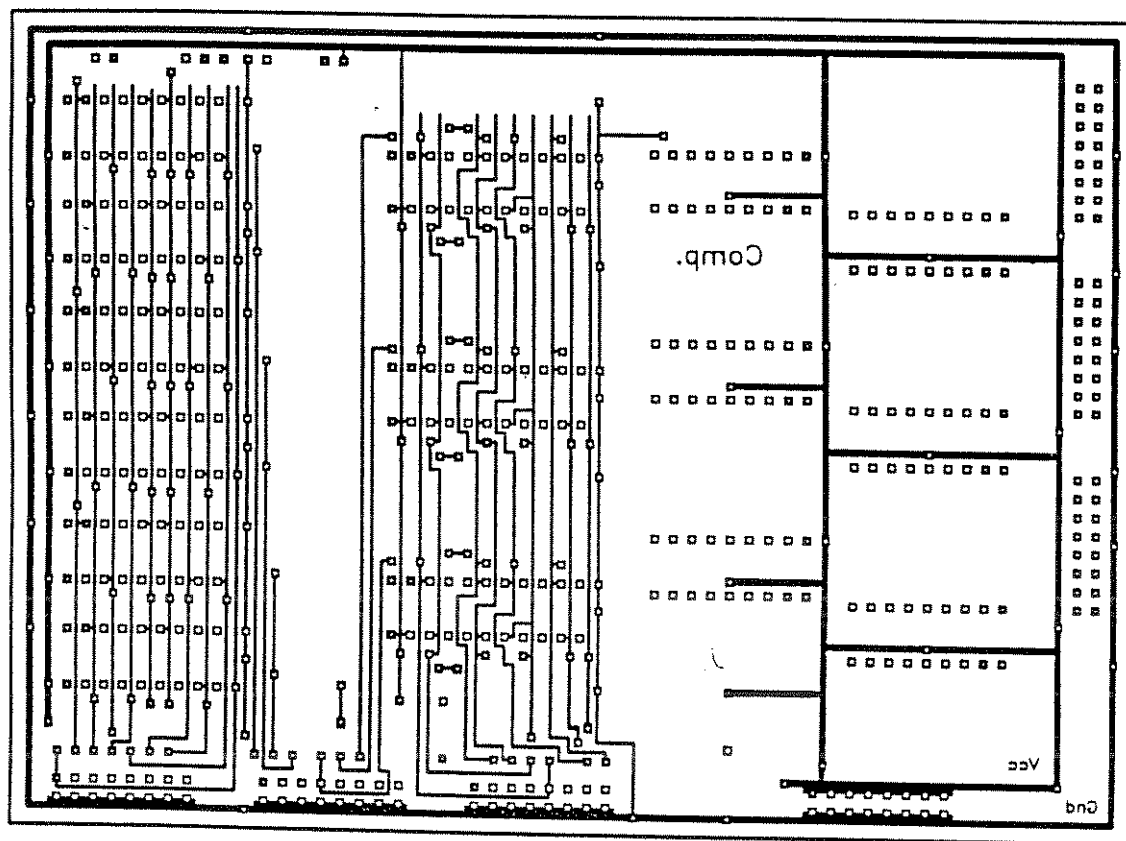


Figura B.II.3 - Lay-Out da Placa de Circuito Impresso da Placa de Contadores Digitais - Trilhas do Lado dos Componentes

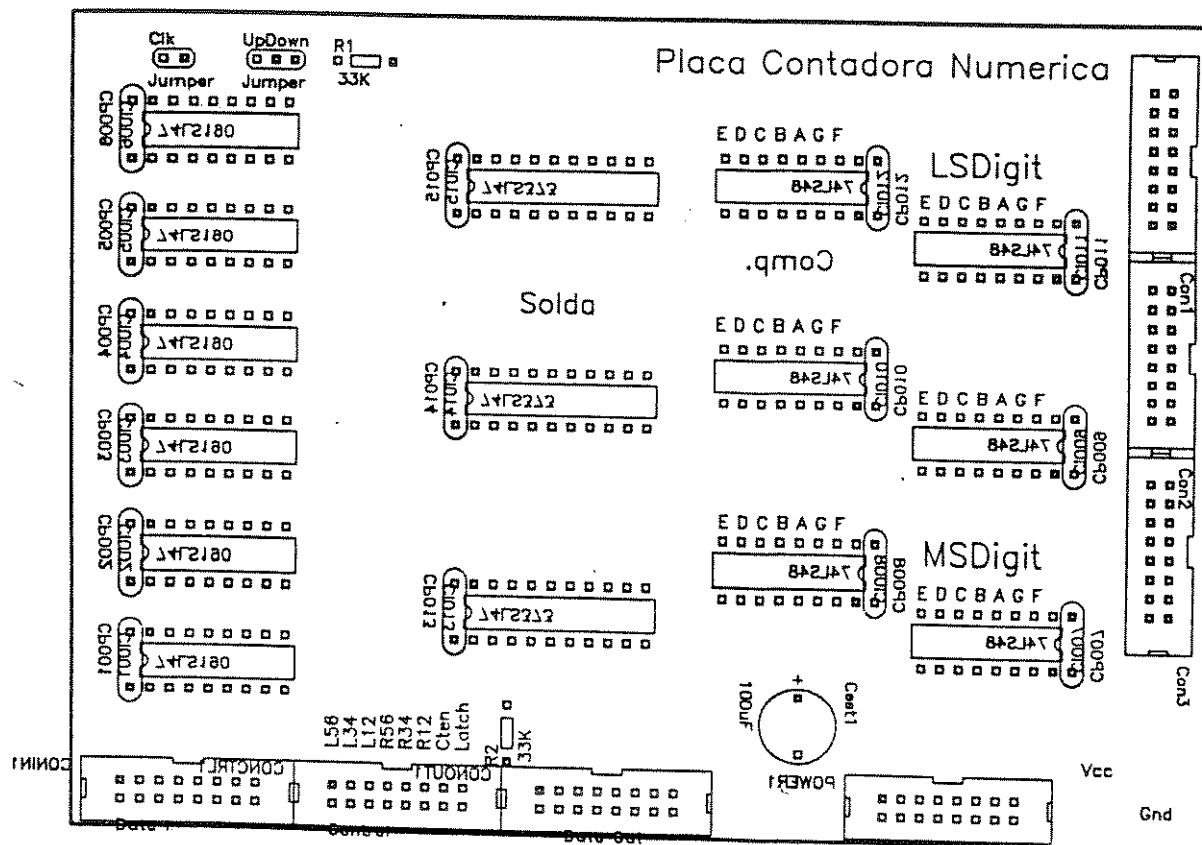


Figura B.II.4 - Lay-Out da Placa de Circuito Impresso da Placa de Contadores Digitais - Posicionamento dos Componentes

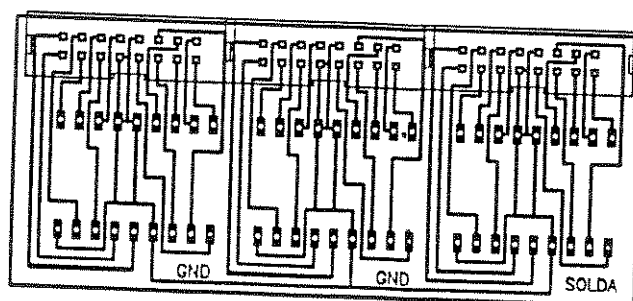
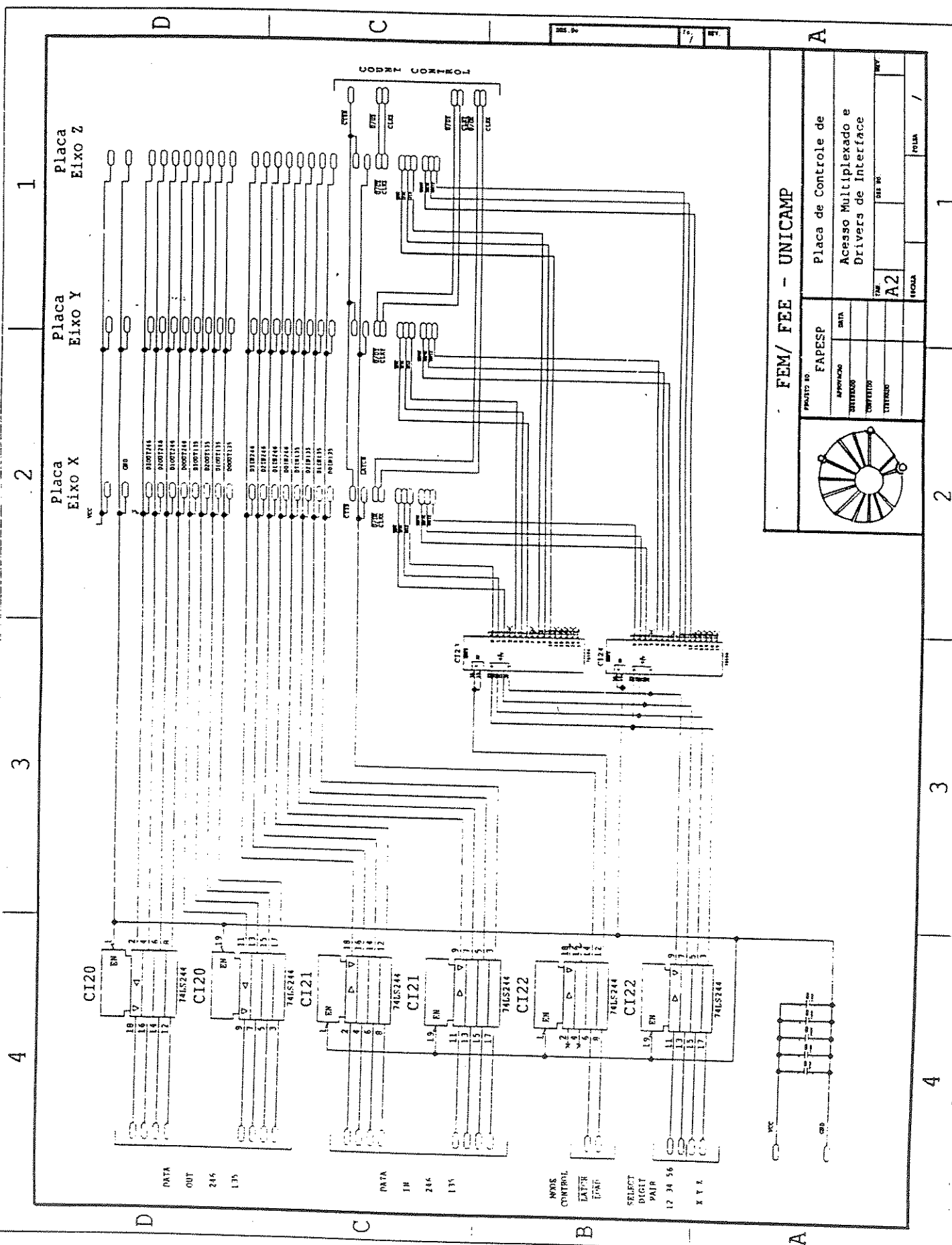


Figura B.II.5 - Lay-Out Impresso da Placa de Displays Digitais - Trilhas do Lado da Solda





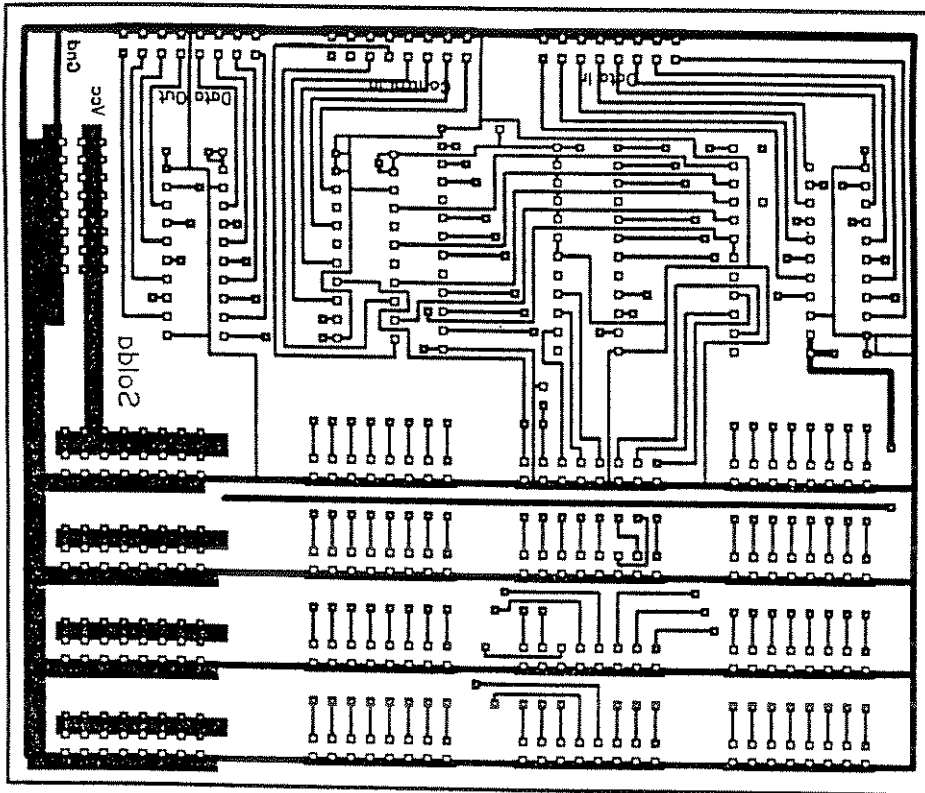


Figura B.III.2 - Lay-Out da Placa de Circuito Impresso da Placa de Acesso Multiplexado - Trilhas do Lado da Solda

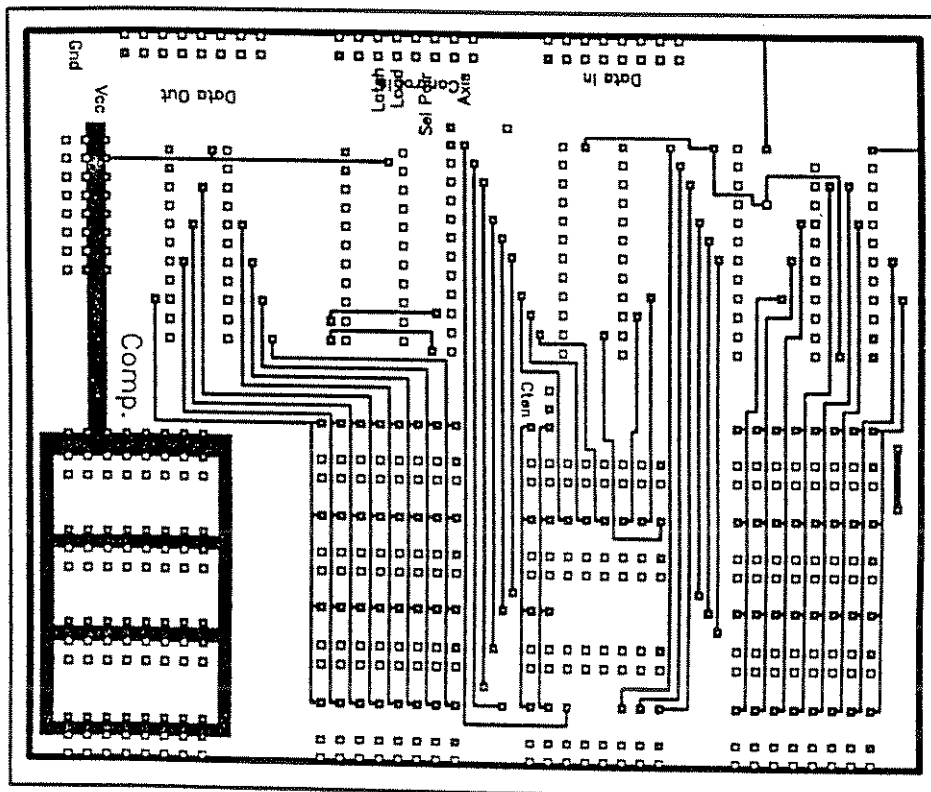


Figura B.III.3 - Lay-Out da Placa de Circuito Impresso da Placa de Acesso Multiplexado - Trilhas do Lado dos Componentes

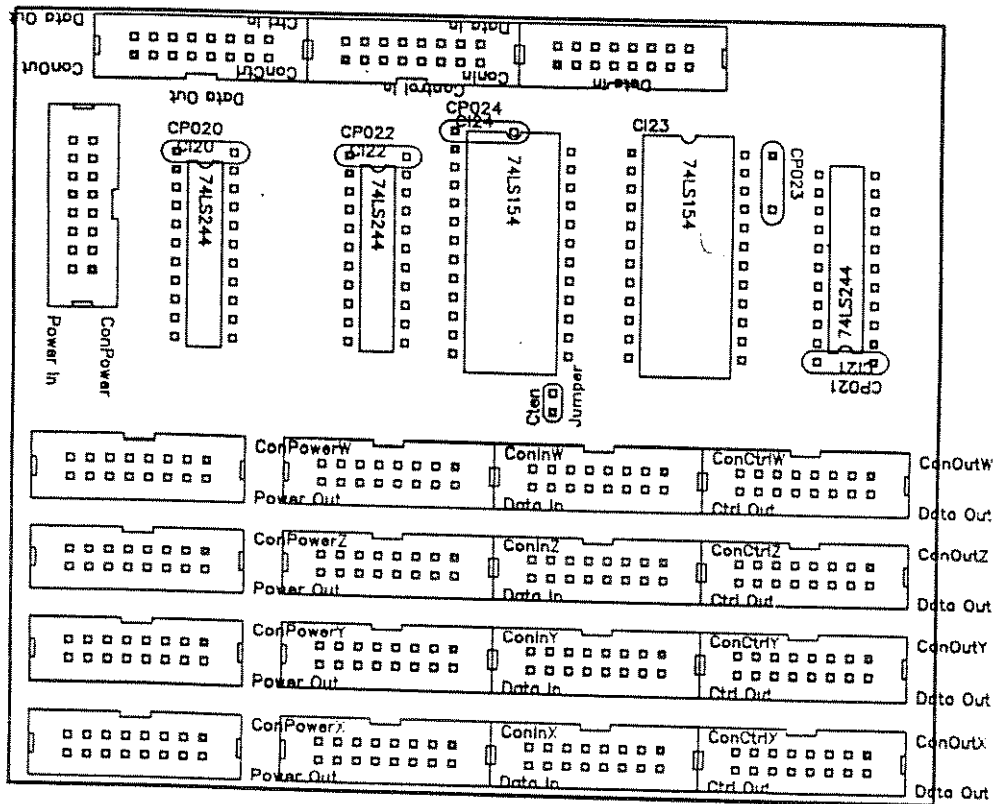


Figura B.III.4 - Lay-Out da Placa de Circuito Impresso da Placa de Acesso Multiplexado - Posicionamento dos Componentes

1

2

3

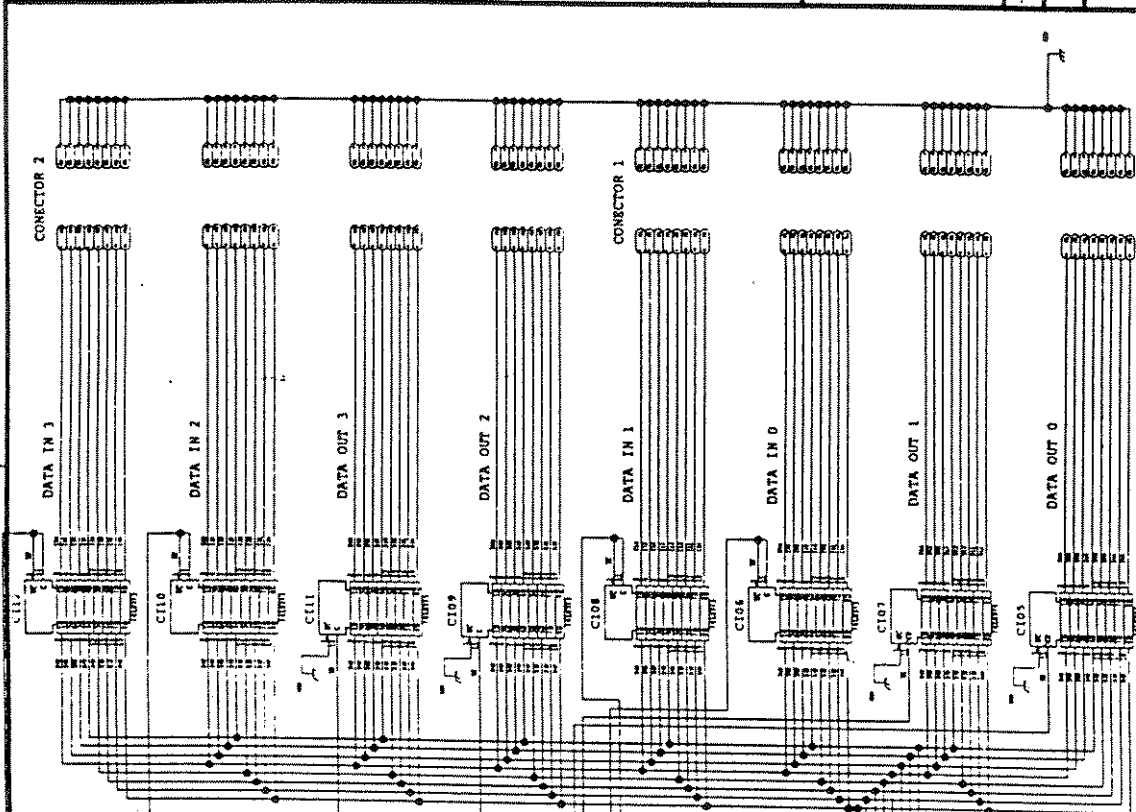
4

D

C

MSB, LS

A



FEM - UNICAMP

Placa de Interface Paralela

para o IBMPC/XT/AT,  
com 2x( 2x8 out, 2x8 in)

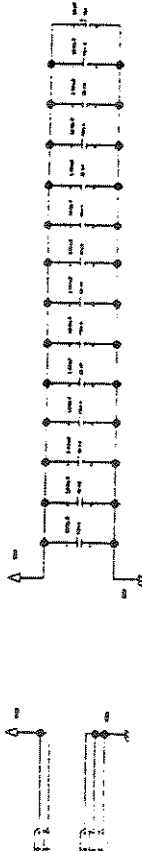
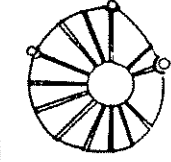
FAPESP

APROVADO DATA 05/93

REVISÃO 01

PROJETO

DATA



A

1

2

3

4

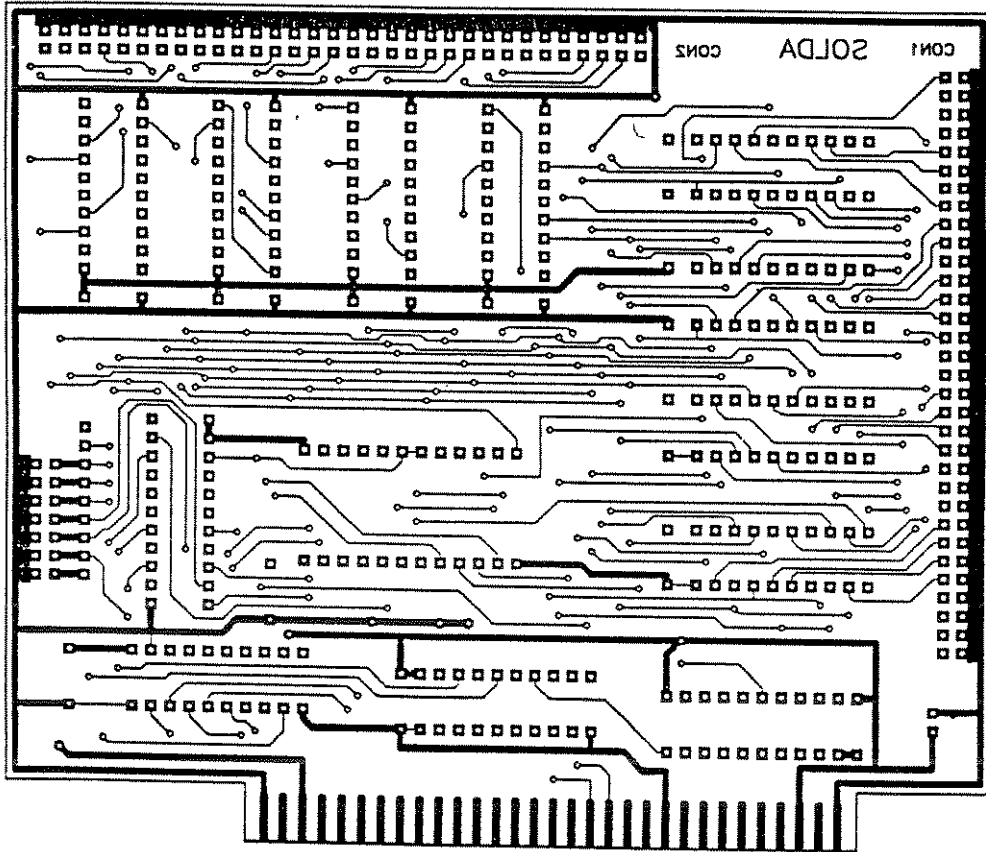


Figura B.IV.2 - Lay-Out da Placa de Circuito Impresso da Placa de Interface Digital - Trilhas do Lado da Solda

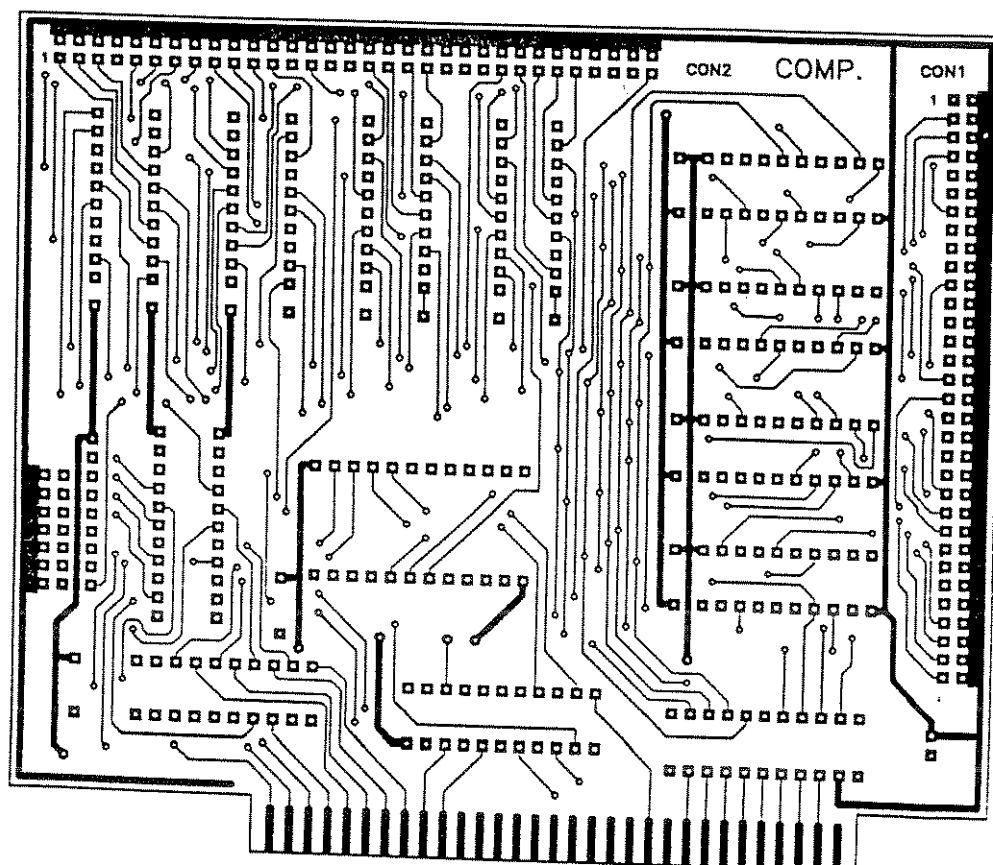


Figura B.IV.3 - Lay-Out da Placa de Circuito Impresso da Placa de Interface Digital - Trilhas do Lado dos Componentes

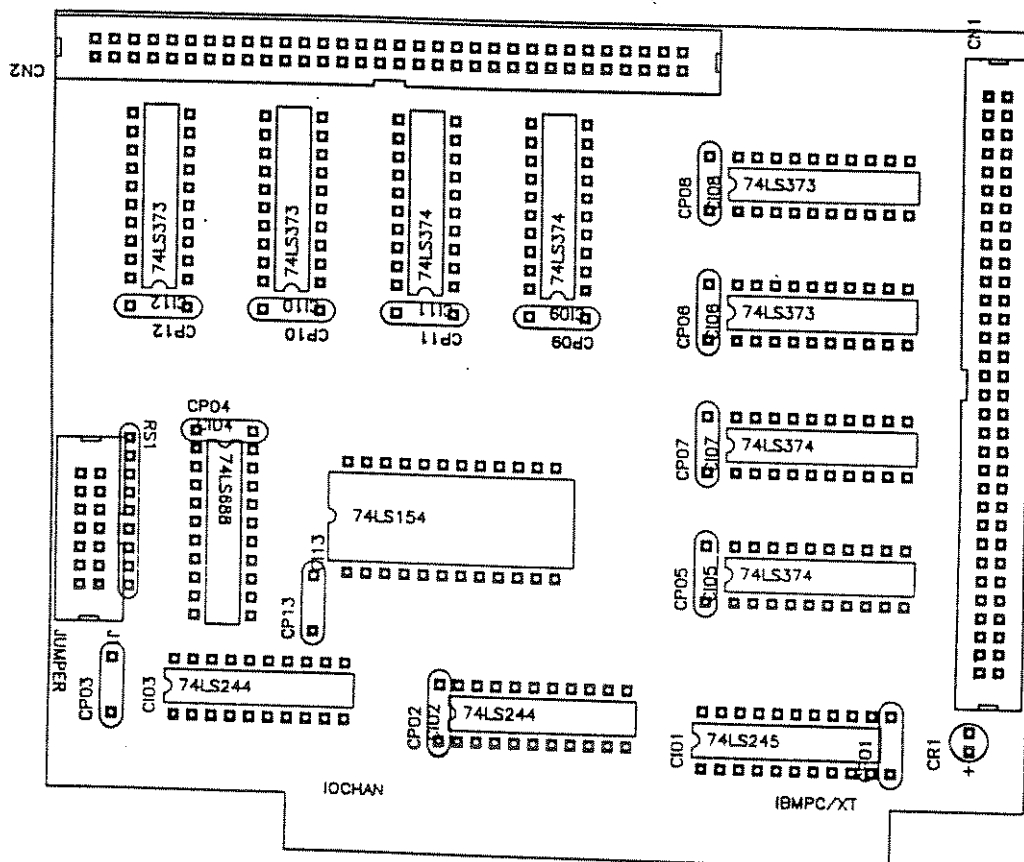
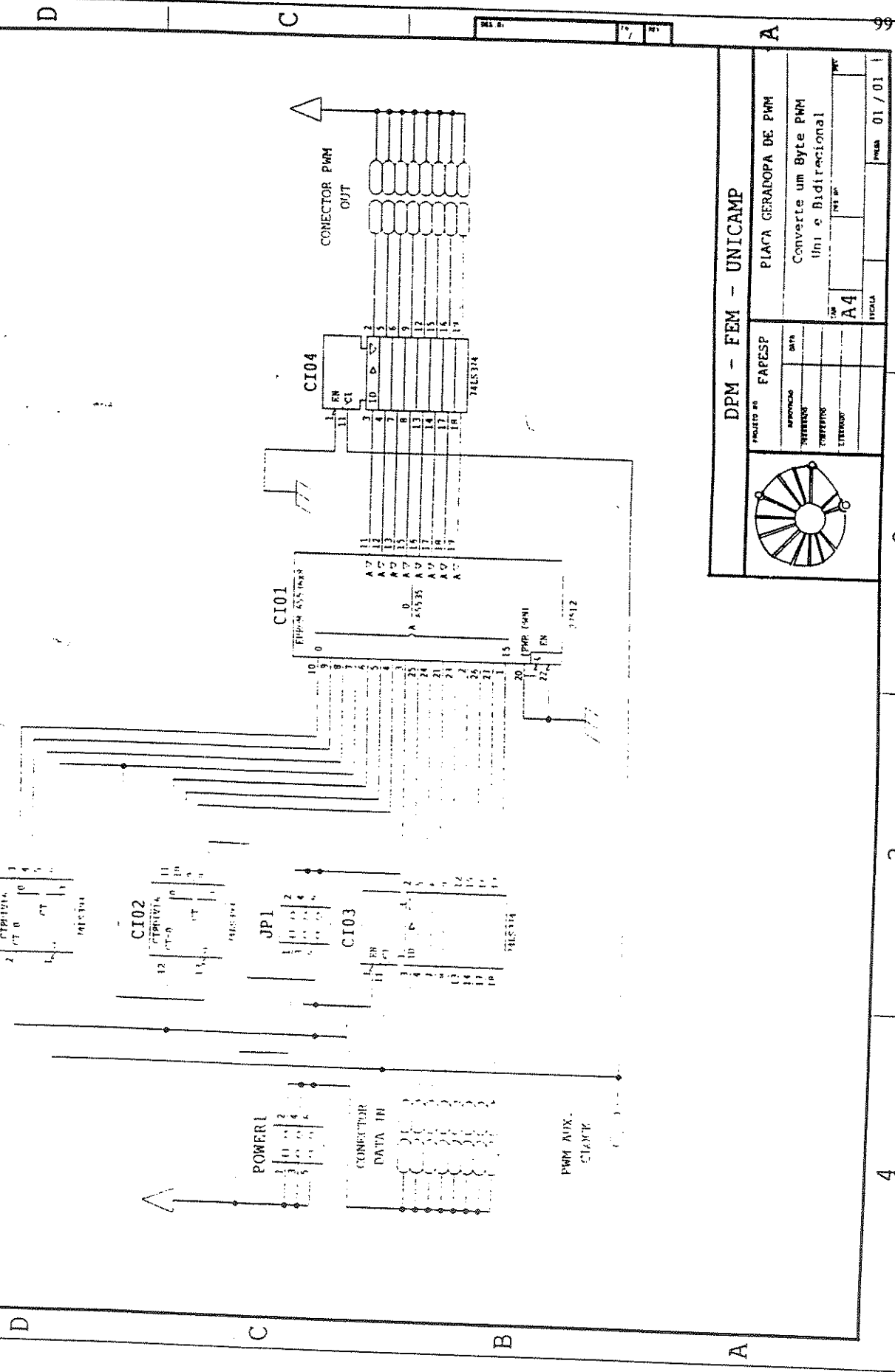


Figura B.IV.4 - Lay-Out da Placa de Circuito Impresso da Placa de Interface Digital - Posicionamento dos Componentes



1 2 3 4



DPM - FEM - UNICAMP	
PIACA GERADORA DE PMM	
Converte um Byte PMM Uni e Bidirecional	
A4	
Ficha 01 / 01	

1 2 3 4

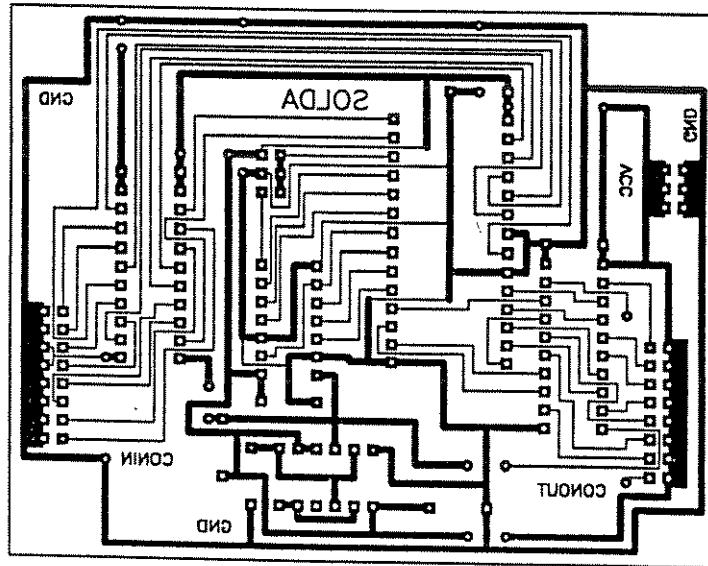


Figura B.V.2 - Lay-Out da Placa de Circuito Impresso da Placa Geradora de PWM - Trilhas do Lado da Solda

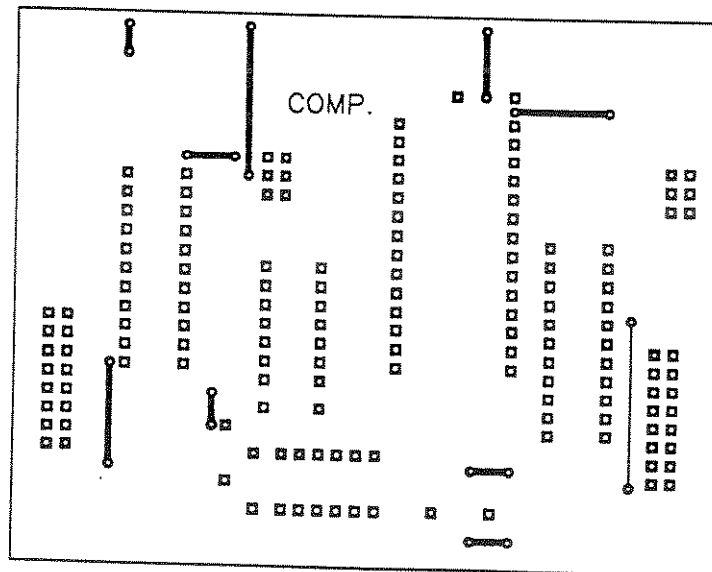


Figura B.V.3 - Lay-Out da Placa de Circuito Impresso da Placa Geradora de PWM - Trilhas do Lado dos Componentes

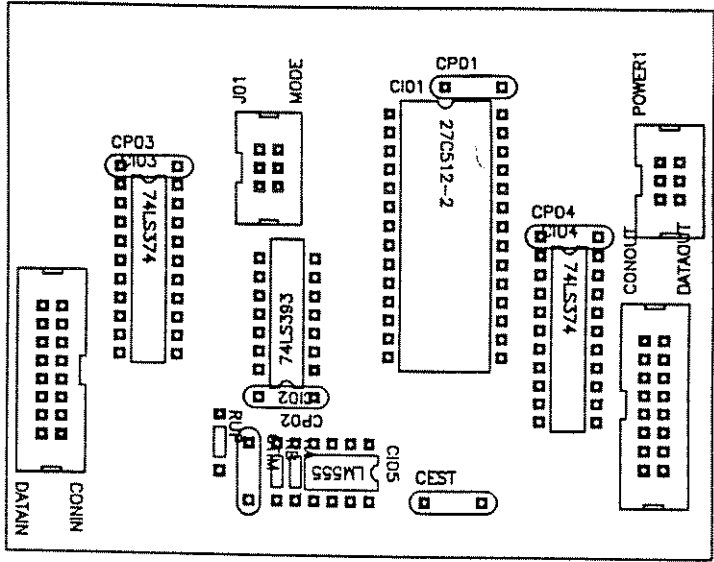
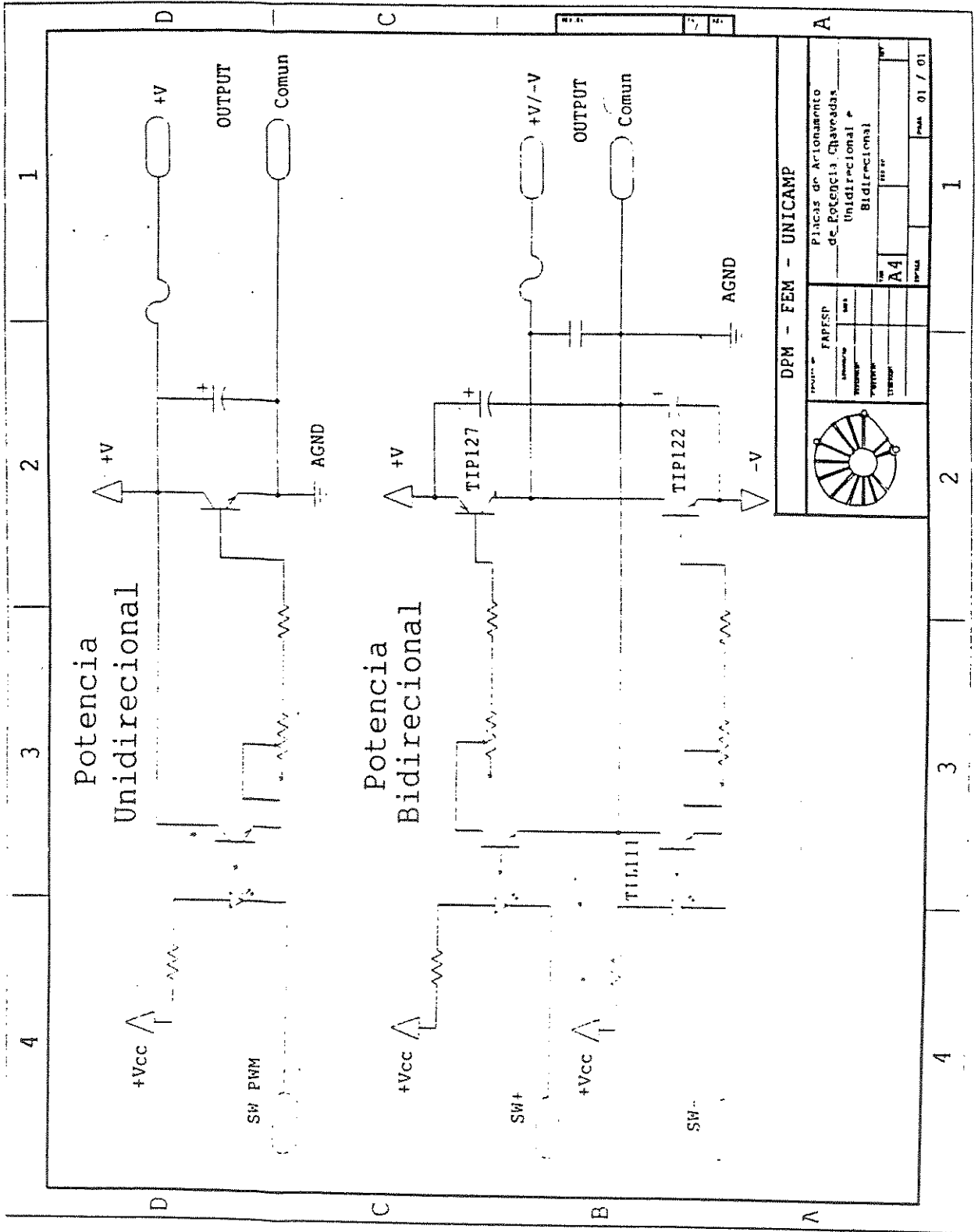
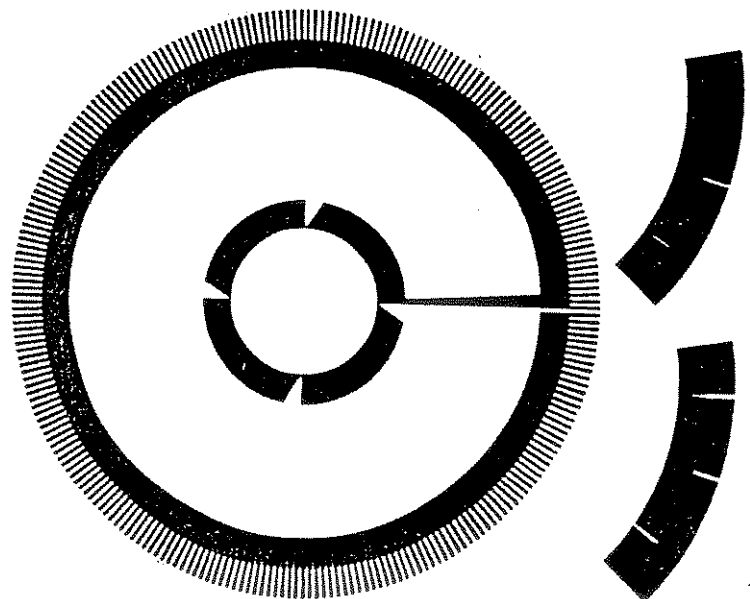
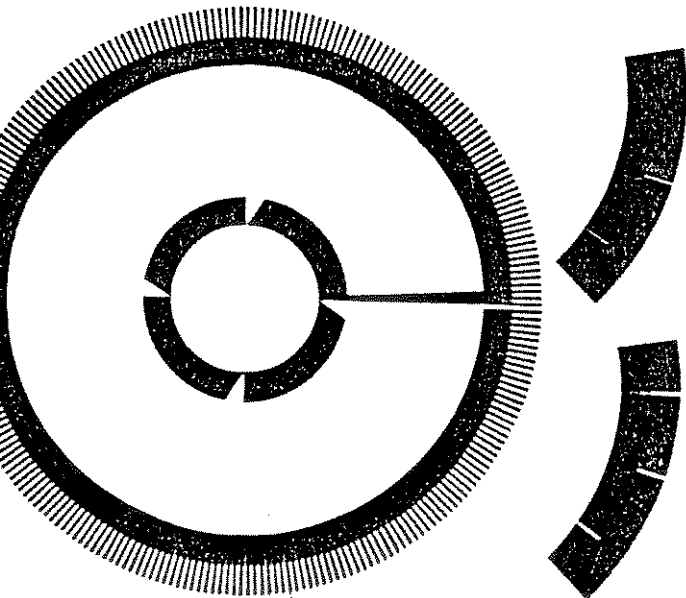
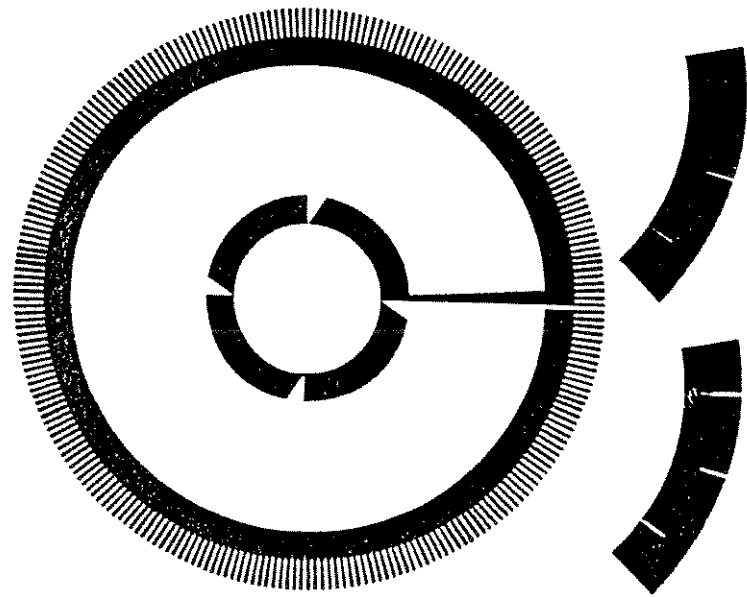
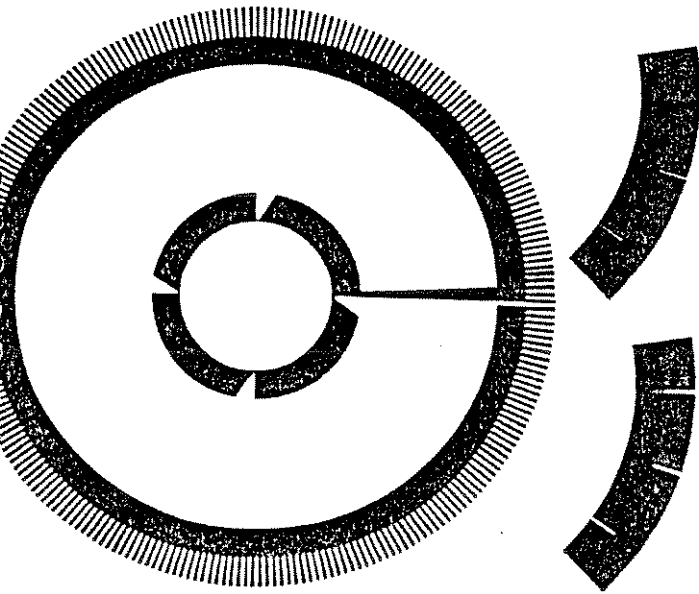
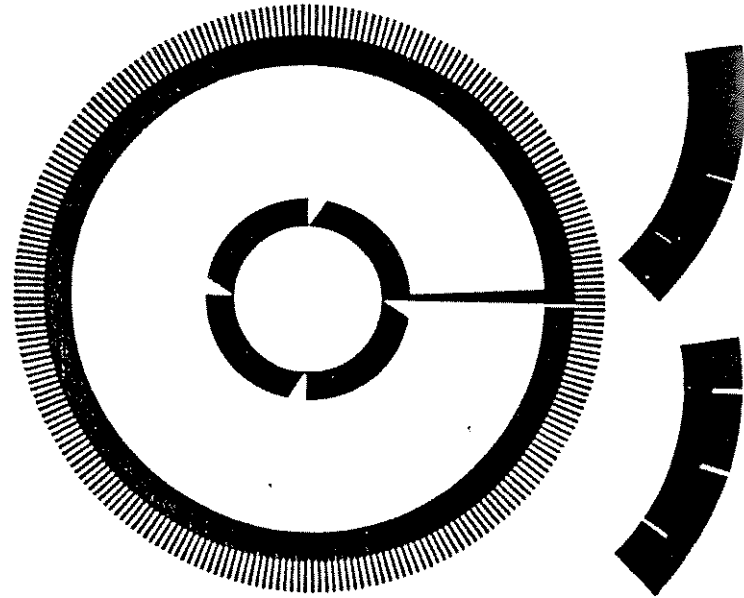
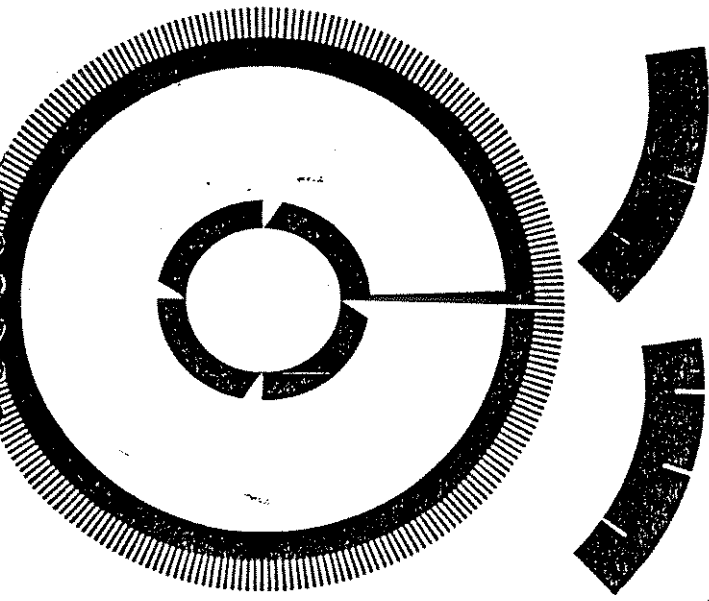


Figura B.V.4 - Lay-Out da Placa de Circuito Impresso da Placa Geradora de PWM - Posicionamento dos Componentes







## Referências Bibliográficas

- [1] SNYDER, W.E. *Industrial Robots : Computer Interfacing and Control* . Prentice Hall, Inc, 1985.
- [2] SNYDER, W.E., and SCHOTT, G. *Using Optical Shaft Encoders*. Robotics Age. 1980.
- [3] BARNEY, G.C. *Intelligent Instrumentation - Microprocessor Applications in Measurement and Control* . Prentice Hall, 1988.
- [4] FU, K.S., GONZALES, R.C. *Robotics - Control, Sensing, Vision and Intelligence*. Mc Graw - Hill, Inc, 1987.
- [5] ASTROM, K., WITTENMARK, B., (1984). *Computer Controlled Systems*. Prentice Hall, 1984.
- [6] KOREM, Y. *Robotics for Engineers*. Mc Graw Hill, 1988.
- [7] EGGBRECHT, L.C. *Interfacing to the IBM Personal Computer* , Howard W. Sams & CONTROLE OTIMO , 1983.

- [8] CRAIG, J.J. *Introduction to Robotics: Mechanics & Control*. Addison - Wesley Publishing, Inc, 1986.
- [9] CRAIG, J.J. *Adaptive Control of Mechanical Manipulators*. Addison Wesley Reading, Mass, 1988.
- [10] PAUL, R.P. *Robot Manipulators - Mathematics, Programming and Control*. The MIT Press Series in Artificial Intelligence, 1982.
- [11] WELLS, D.A. *Theory and Problems of Lagrangian Dynamics*, McGraw-Hill Book Company, 1967.
- [12] PHILLIPS, C.L., NAGLE, H.T. *Digital Control System, Analysis and Design*. Prentice Hall, Inc, 1984.
- [13] GIBSON, J.E. *Nonlinear Automatic Control*. Mac Graw Hill Book Company, 1963.
- [14] CLARKE, D.W. *PID Algorithms and Their Implementation*, O.U.E.L. Report No. 1482/83 Oxford, Eng, 1983.
- [16] *Software 2900 Programming System*. (Software de Programação de Circuitos Lógicos e Memórias EPROM), Data I/O Corporation.
- [17] RIEDEL, N.K., MCANINCH, D.A., FICHER, C., GOLDSTEIN, N.B. *A Signal Processing Implementation for an IBM-PC-Based Workstation*. IEEE MICRO, October 1985.



- [18] GUTTAG, K.M., ALBERS, T.M., ASAL, M.D., ROSE, K.G. *The TMS34010: An Embedded Microprocessor*. IEEE MICRO, June 1988.
- [19] *TMS32010 User's Guide*. (Digital Signal Processor Products), Texas Instruments Inc.
- [20] *Digital Signal Processing Applications with the TMS320 Family*. (Theory, Algorithms and Implementations), Texas Instruments Inc.
- [21] SMIAROWSKI JR, A., ANDERSON, J.N. *A Fast Computer Architecture for the Control of Robots*. Computers in Electronics Engineering, Pergamon Press, v. 17, n. 3, 1991
- [22] *RISC Data Book*. (The R3000 RISC Processor Family), Integrated Device Technology, Inc, 1991.
- [23] MIRAPURI, S.S., WOODACRE, M.S., VASSEGHI, N.,. *The Mips R4000 Processor*. IEEE MICRO, April 1992.
- [24] DIEFENDORF, K., ALLEN, M. *Organization of the Motorola 88110 Superscalar RISC Microprocessor*, IEEE MICRO, April 1992.
- [25] MARSHALL, T. *Real World RISCs*. (Microprocessadores Am29000 e Motorola 88000), BYTE, May 1988
- [26] LOWELL, B. *The Am29000 as an Embedded Controller* , Dr. Dobb's Journal, March 1992

- [27] MANN, D. *Unix and the Am29000*. IEEE MICRO, February 1992
- [28] SIMPSON, D. *Embedded RISCs Take Off*. Systems Integration, August 1989
- [29] ROBERTS, C.E. *A Risc Processor for Embedded Applications Within an ASIC*. (Microprocessor VL86C010RISC), IEEE MICRO, October 1991
- [30] CATES, R. *Processor Architecture Considerations for Embedded Controller Applications*. June 1988.
- [31] PURKISER, C., KARDACH, J. *The Intel 376 Family for Embedded Processor Applications*. IEEE MICRO. June 1988
- [32] ATKINS, M. *Performance and the i860 Microprocessor*. IEEE MICRO, October 1991
- [33] RYAN, D.P. *Intel's 80960: An Architecture Optimized for Embedded Control*. IEEE MICRO, June 1988
- [34] HUTCHISON, D. *Local Area Network Architectures*. Addison-Wesley Publishing Company, 1987
- [35] *Data Transmission and Control Circuits Selection Guide*. Texas Instruments Inc.
- [37] HERVELLA, C. *Projeto e Desenvolvimento de um Sistema para Aquisição, Monitoramento e Controle de Sistemas Mecânicos Via Computador*. Relatório FAPESP, Projeto Número: 91/2301-7, 1992

- [38] MIRANDA, M.F. *Controle de um Servomecanismo por um Microcomputador Dedicado: Uma Contribuição ao Estudo de Controladores para Robôs Industriais*. Tese de Mestrado, UNICAMP, 1992
- [39] FAYAN, B.L. *Projeto e Concepção de um Supervisor de Controle Para um Robô Industrial*. Tese de Mestrado, UNICAMP, 1992
- [40] DIAS, C.H. *Realização Experimental de um Supervisor de Controle para um Robô Industrial*. Tese de Mestrado, UNICAMP, 1992.
- [41] SOUZA, J.P. *Procedimento Automático para Aquisição e Tratamento do Movimento de um Robô - Uma Contribuição ao Estudo de Controladores Não Lineares*. Tese de Mestrado, UNICAMP, 1992
- [42] ROSÁRIO, J.M. *Projeto e Implementação de Um Novo Tipo de Controlador em Robótica Industrial*. FAPESP, Projeto Número: 90/0959-2, 1992
- [43] ROSÁRIO, J.M. *Projeto e Desenvolvimento de um Sistema para Supervisor de Controle para um Robô Industrial*, FAPESP, Projeto Número: 91/2301-7, 1992