

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO INDUSTRIAL

CONTROLE INTELIGENTE DE ROBÔS

por: Antonio Joaquim Melo Carvalhal

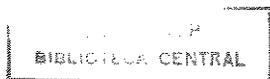
orientador: Prof. Dr. Fernando A. Campos Gomide

Tese apresentada à Faculdade de Engenharia Elétrica da Universidade Estadual de Campinas-UNICAMP - como parte dos requisitos para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA.

*Este documento corresponde à
versão final da tese de mestrado
de Antonio Joaquim Melo Carvalhal.*

Fernando A. Campos Gomide
Prof. Dr. FERNANDO GOMIDE
DCA/FEE/UNICAMP

FEVEREIRO 1988



À Elisabete,
Ao Roberto ,
Ao William.

AGRADECIMENTOS,

- . Ao Prof. Dr. Fernando A. C. Gomide pela orientação, estímulo e amizade.
- . Ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - pelo apoio financeiro na parte inicial deste trabalho.
- . à FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo - pelo apoio financeiro para sua conclusão.
- . à UNICAMP e à quantidade enumerável de colegas da Engenharia Elétrica, pela discussão, apoio, companheirismo e colaboração.
- . Ao CTI/IA - Centro Tecnológico para Informática/Instituto de Automação, pelo suporte no desenvolvimento da etapa inicial deste trabalho.
- . aos meus familiares pelo incentivo.
- . à Huda e à Cristina pela montagem final deste trabalho.
- . à todos aqueles que direta ou indiretamente contribuíram para a sua realização.

ÍNDICE

CAPÍTULO 1 - INTRODUÇÃO.....	1.1
CAPÍTULO 2 - MODELAGEM CINEMÁTICA DE ROBÔS INDUSTRIAIS.....	2.1
2.1 - Introdução.....	2.1
2.2 - Robôs Industriais.....	2.1
2.3 - Equações dos Ligamentos e Matrizes de Transformação.....	2.2
2.3.1 - Especificação das Matrizes "A".....	2.2
2.3.2 - Equações de Controle de Punho.....	2.6
2.4 - Equações Cinemáticas de Controle.....	2.5
2.4.1 - Solução Para o Punho.....	2.5
2.4.2 - Solução Para o Braço.....	2.11
2.4.3 - Solução Para o Robô.....	2.14
2.5 - Resumo.....	2.19
CAPÍTULO 3 - MODELAGEM DINÂMICA DE ROBÔ INDUSTRIAL.....	3.1
3.1 - Introdução.....	3.1
3.2 - As equações Diferenciais de Movimento.....	3.1
3.3 - Notação Por Combinadores.....	3.7
3.3.1 - Conjunto de Primitivas.....	3.9
3.3.2 - Operações Básicas Com Combinadores.....	3.11
3.3.3 - Implementação.....	3.17
3.4 - Resumo.....	3.27
CAPÍTULO 4 - RESULTADOS COMPUTACIONAIS	
4.1 - Introdução.....	4.1
4.2 - O Robô.....	4.1
4.3 - Parâmetros e Matrizes.....	4.1
4.4 - Modelo Simbólico.....	4.2
4.5 - Resultados.....	4.3
4.6 - Conclusão.....	4.7

CAPÍTULO 5 - CONTROLE HIERÁRQUICO DE ROBÔS	5.1
5.1 - Introdução	5.1
5.2 - Níveis Hierárquicos de Sistema de Controle	5.2
5.2.1 - O Nível Organizador (ORCOH)	5.3
5.2.2 - O Nível Coordenador (CORCOH)	5.8
5.2.3 - O Nível Supervisor (SURCOH)	5.12
5.2.4 - Sistema de Visão e Tato	5.17
5.3 - Conclusão	5.18
CAPÍTULO 6 - CONCLUSÃO	6.1
REFERÊNCIAS BIBLIOGRÁFICAS	R.1

CAPÍTULO 1 - INTRODUÇÃO

CAPÍTULO 1 - INTRODUÇÃO

O robô industrial, como é concebível ainda na atualidade, tem suas características decorrentes de duas origens: o dispositivo comandado à distância e a máquina ferramenta controlada numericamente.

O dispositivo comandado à distância foi desenvolvido durante a 2.^a Guerra Mundial para manipular materiais radioativos.

O operador, separado do material radioativo por parede de concreto, controlava com manipuladores externos, mãos do sistema em contato com o material.

O aperfeiçoamento de sistemas em que os manipuladores externos, tinham sua força realimentada a partir das mãos mecânicas em contato com o material, além do desenvolvimento de máquinas de controle numérico, reunia sistemas servo controlados às técnicas de computação digital.

As tarefas realizadas pelos robôs nesta fase passaram a ser de simples como abrir e fechar garra, carga e descarga, troca de objetos de local, às mais complexas, sendo que a trajetória a ser seguida pelo conjunto punho/garrado robô para a realização dessas tarefas era inicialmente pré gravada em memória.

A realização dos cálculos de transformação de coordenadas do objeto a ser alcançado através de computador acoplado ao sistema robótico e a demonstração da possibilidade de identificação e de localização de objetos por sistemas com câmeras acopladas, somados a outros sensores, além de permitirem a realização de tarefas mais complexas, delineou o caminho das pesquisas posteriores.

E estas o foram na direção do sistema robótico inteligente, equipado o mais possível com sensores e tanto quanto possível próximo dos parâmetros de desempenho do controle humano total.

O controle inteligente de robôs tornou-se desta maneira a tendência atual da pesquisa na área.

Esta pesquisa procura identificar e reduzir as limitantes de desempenho do sistema diretamente ligadas com seu comportamento cinemático e dinâmico, bem como o desenvolvimento e aperfeiçoamento de sensores em particular a integração completa desses conjuntos.

O aperfeiçoamento dessas características, pode ser pesquisado analisando-se então, para o sistema robótico,

- . o seu controle cinemático
- . o seu controle dinâmico
- . o seu controle integrado, com utilização de sensores, de forma hierárquica

Este trabalho, objetivando-se estudar os itens anteriores, tem a seguinte organização:

- . O Segundo Capítulo, Controle Cinemático e de Posição do Conjunto punho/garra, procura-se mostrá-lo através da técnica de solução composta para os subsistemas de posicionamento e orientação.
- . O Terceiro Capítulo, Análise e Implementação de Programa Gerador do Modelo Dinâmico Para Robô Industrial, é desenvolvido utilizando-se da ferramenta computacional denominada combinadores, criada para utilização em linguagem do tipo FP ("Functional Programming"), sendo Prolog a linguagem de programação escolhida.
- . A seguir a este Capítulo, mostram-se os resultados computacionais obtidos pela utilização do modelo anterior, implementado em um determinado tipo de robô.
- . No Quinto Capítulo é delineada a proposição de um modelo global de controle hierárquico capaz de permitir a conjugação de técnicas atualizadas no campo de reconhecimento, comparação e tomada de decisão com as tradicionais de controle clássico do "hardware" existente.

Acerca do primeiro item, Controle Cinemático e de Posição do Conjunto Manipulador punho/garra, vários artigos tem sido publicados, [2],[3],[4],[5], sendo que a técnica empregada por Paul, R.P. em [2] e Paul, R.P.; Shimano, B; Meyer G. em [4] procura obter a solução para o controle da junta sequencialmente.

Cada variável sucessiva é isolada a partir da mão/ferramenta pela pré-multiplicação das matrizes "A".

A técnica empregada em [3], [5] e [6] e utilizada neste trabalho trata separadamente cada um dos subsistemas, braço manipulador e garra/ferramenta, reduzindo erros e multiplicações tediosas.

Este resultado conduz à diminuição do número de cálculos computacionais, além de permitir de modo claro a definição do primeiro problema a ser analisado nos manipuladores: o de controle de posição.

O segundo item, o de análise e implementação de modelo dinâmico para robô industrial, vem sendo pesquisado a partir do nascimento de dinâmica computacional de robô em 1973[22].

Esta, considerada como uma síntese da mecânica clássica com o processamento simbólico[24],[29], tem procurado resolver problemas práticos de forma a obter modelos cinemáticos e dinâmicos precisos para mecanismos robóticos.

Em 1984, Vukobratovic e Kiricanski,[33], utilizando um programa de modelagem simbólico implementado em LISP, mostraram a eficiência computacional do modelo dinâmico para robôs.

Além destes, implementados em LISP Clássico, ou linguagens especiais como REDUCE existem outros como [23],[30].

Neste item, é desenvolvido o modelamento dinâmico seguindo o princípio de Lagrange-Euler acrescido do ferramental de combinadores descritos em[10],[11],[31],[32], programado em linguagem Prolog.

A este conjunto de formulações acrescentou-se o trabalho realizado por Siqueira Neto, J.L.;Alves, J.B.M; Pereira A.E.C [25].

Neste trabalho[25], baseado em[7], procurou-se gerar o modelo dinâmico completo com a ajuda da ferramenta de combinadores[31], sendo este item o desenvolvimento do referido trabalho e acrescenta a ele os seguintes aspectos:

- . desenvolvimento dos termos de aceleração inercial , para matrizes simétricas nxn.
- . geração dos termos de aceleração de Coriolis e centrífuga.
- . geração dos termos de aceleração gravitacional.
- . geração de ambiente integrado de "software" em "Prolog",

permitindo a interação com o usuário.

No modelo foram considerados todos os termos para n juntas, $1 \leq n \leq 6$, incluindo todos os acoplamentos existentes.

No último item, propõe-se um modelo global de controle hierárquico, composta de três níveis[14],[16],[26].

O primeiro nível, considera-se como aquele de maior inteligência: considera todas as atividades dos sensores, determina tarefas e gerencia os níveis inferiores.

O segundo nível, está intimamente ligado ao controle da trajetória a ser seguido, de acordo com as informações do nível superior e a realimentação do nível inferior.

O último nível, o de menor "inteligência", contém o controle direto das juntas, utilizando o microprocessamento amostrado dos sinais.

Este modelo proposto difere daquele mostrado em[14] por concentrar o processamento dos sinais dos sensores no nível Organizador do Controle Hierárquico (ORCOH) o de maior inteligência.

Esta proposição procura liberar o nível intermediário, Coordenador de Controle Hierárquico (CORCOH) para o controle de trajetória levando-o a processar os algoritmos que permitam evitar as interferências no caminho.

CAPÍTULO 2 - MODELAGEM CINEMÁTICA DE
ROBÔS INDUSTRIAIS

2.1 INTRODUÇÃO

Este capítulo tem como finalidade apresentar uma introdução aos robôs industriais, o ferramental de modelagem matricial básico, e equações cinemáticas de controle.

O aspecto cinemático é desenvolvido através da técnica de solução composta para os subsistemas de posicionamento e de orientação.

2.2 ROBÔS INDUSTRIAIS

Um robô industrial, no conceito do Robot Institute of America é:

"Um operador reprogramável e multifuncional projetado para manipular material, peças, ferramentas ou dispositivos especializados, através de movimentos variáveis programados para a realização de uma variedade de tarefas".

O desempenho do robô pode ser medido em termos de:

- . precisão
- . repetitividade
- . velocidade
- . capacidade de carga e/ou força

A precisão é definida como a habilidade do robô em alcançar um pré-determinado ponto do seu espaço de trabalho na presença de imperfeições estruturais (como atritos mecânicos, imprecisão de seus atuadores e sensores).

A repetitividade é definida como a habilidade do robô em executar o mesmo movimento preciso várias vezes a uma dada velocidade e capacidade de carga e/ou força, medindo a produtividade do robô.

Genêricamente um robô industrial consiste de vários corpos rígidos chamados ligamentos ("links") conectados em série por juntas rotacionais (de deslocamento angular) ou prismáticas (de deslocamento linear). Um dos terminais de cadeia é conectada à base de suporte enquanto que a outra é livre e equipada para manipular objetos ou realizar tarefas diversas. O movimento das juntas resulta no movimento relativo de seus ligamentos.

Mecanicamente, um robô é composto de um braço (ou conjunto primário) e um conjunto secundário composto de punho e ferramenta.

O robô deve alcançar o seu objetivo dentro de seu volume de trabalho.

O volume de trabalho é uma região de influência na qual o seu conjunto primário consegue posicionar a unidade do conjunto secundário em qualquer ponto interno.

O conjunto secundário consiste tipicamente de movimentos de três graus de liberdade, os quais localizam ou posicionam o punho em relação à peça.

O conjunto secundário, em um robô típico de seis graus de liberdade, comporta-se como mecanismo de orientação, enquanto que o conjunto primário é o mecanismo de posicionamento.

Em termos de características de geometria e de movimento de seu braço, os robôs industriais podem ser incluídos em uma das cinco categorias básicas abaixo:

- . de coordenadas cartesianas (três eixos lineares - fig 2.1)
- . de coordenadas cilíndricas (dois eixos lineares e um eixo rotacional - fig 2.2)
- . de coordenadas esféricas ou polares (um eixo linear e dois eixos rotacionais - fig 2.3)
- . de coordenadas articuladas ou de revolução (três eixos rotacionais - fig 2.4)
- . mistos

2.3 EQUAÇÕES DOS LIGAMENTOS E MATRIZES DE TRANSFORMAÇÃO

O relacionamento entre todos os ligamentos dos conjuntos primário e secundário é representado por meio de matrizes.

Uma matriz de transformação A_i descreve a translação e rotação entre o sistema de coordenadas do ligamento i e o do ligamento $i-1$. Desta forma, para um robô de seis graus de liberdade.

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6$$

2.3.1 ESPECIFICAÇÃO DAS MATRIZES "A"

As juntas são de revolução ou prismáticas, e a direção de torção entre os ligamentos está em eixos paralelos ou perpendiculares. Um relacionamento simples permite descrever as transformações entre ligamentos. As rotações e convenções que se apresentam seguem [1],

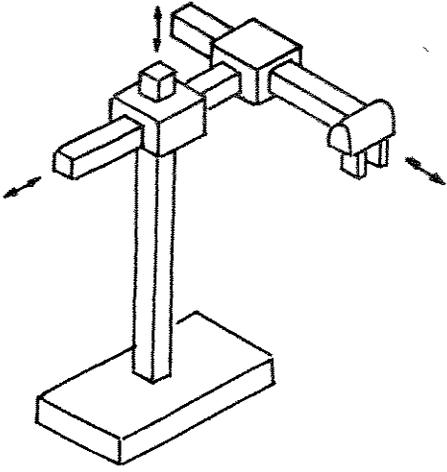


fig 2.1

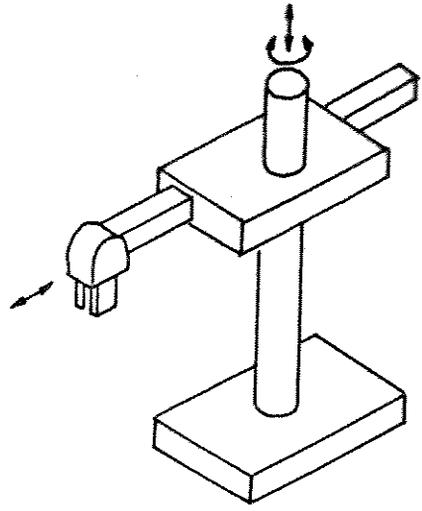


fig 2.2

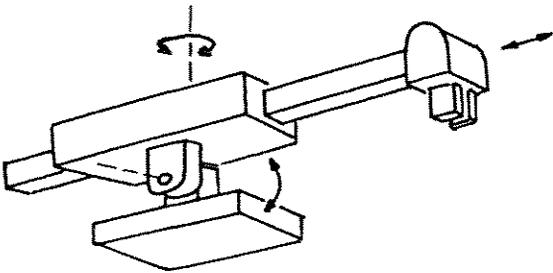


fig 2.3

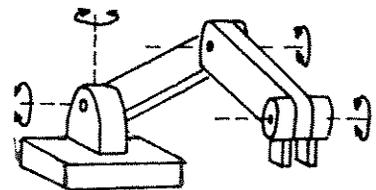


fig 2.4

[2] e [3] e permitem determinar o sistema de coordenadas, fig.2.5, para a definição das matrizes A, ou seja:

- . alinhar os eixos X de todos sistemas de coordenadas na mesma direção do sistema de coordenadas da base fixa.
- . as juntas rotacionais executando revolução em torno do eixo Z respectivo.
- . as juntas prismáticas executando translação ao longo do eixo Z respectivo.

Um ligamento é comumente apresentado por parâmetros fixos conhecidos como parâmetros de Denavit e Hartenberg (DHP), o comprimento do link como a_i , o ângulo de rotação como α_i e como parâmetros do movimento, θ_i a variável rotacional e d_i a translacional.

A regra descrita, em casos de orientação do conjunto secundário, pelos ângulos RPY ("roll", "pitch", "yaw") e braço cartesiano é algo modificada [3].

De um modo geral, a matriz de transformação homogênea A_i , entre o ligamento $i-1$, conforme [1], é dada por:

$$A_i = \text{Rot}(Z, \theta_i) \text{Trans}(0, 0, d_i) \text{Trans}(a_i, 0, 0) \text{Rot}(X, \alpha_i)$$

A ordem de transformação, aqui é da esquerda para direita, devido ao relacionamento entre os ligamentos sucessivos:

$$A_i = \begin{pmatrix} C\theta & -S\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

determinando

$$A_i = \begin{pmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

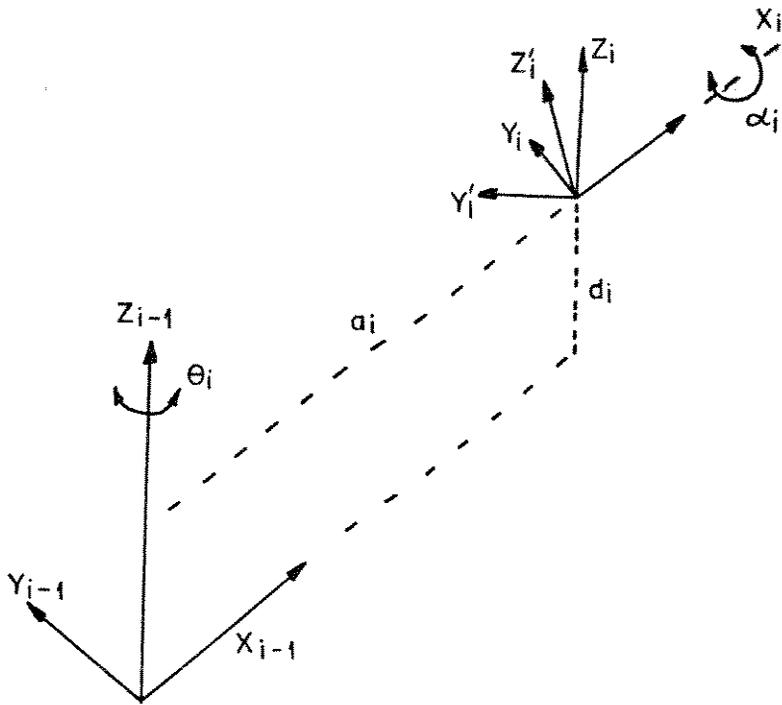


fig 2.5

onde

$$c\theta_i = \cos\theta_i, \quad s\theta_i = \sin\theta_i, \quad c\alpha_i = \cos\alpha_i, \quad s\alpha_i = \sin\alpha_i, \quad a_i c\theta_i = a_i \cos\theta_i, \\ a_i s\theta_i = a_i \sin\theta_i, \quad a_i = 0 \quad \text{para juntas prismáticas.}$$

2.3.2 EQUAÇÕES DE CONTROLE DE PUNHO

A orientação do robô, determinada pelo sistema secundário (punho), é frequentemente especificada por uma sequência de rotações em torno dos eixos X, Y e Z.

Pode ser analisada usando-se dois tipos de ângulos: RPY e ângulos de Euler.

Nos ângulos de Euler, Fig. 2.6, utilizados no robô PUMA 560, toda orientação possível é descrita de:

- . rotação de um ângulo θ_1 , em torno do eixo Z
- . rotação de um ângulo θ_2 , em torno do eixo Y
- . rotação de um ângulo θ_3 , em torno do eixo Z

o que permite definir:

$$T_{\text{EULER}} = \begin{pmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta_2 & 0 & s\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ -s\theta_2 & 0 & c\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ = \begin{pmatrix} c\theta_1 c\theta_2 c\theta_3 - s\theta_1 s\theta_3 & -c\theta_1 c\theta_2 s\theta_3 - s\theta_1 c\theta_3 & c\theta_1 s\theta_2 & 0 \\ s\theta_1 c\theta_2 c\theta_3 + c\theta_1 s\theta_3 & -s\theta_1 c\theta_2 s\theta_3 + c\theta_1 c\theta_3 & s\theta_1 s\theta_2 & 0 \\ -s\theta_2 c\theta_3 & s\theta_2 s\theta_3 & c\theta_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{eq2.1}) \\ = \begin{pmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta_2 & 0 & s\theta_2 & 0 \\ s\theta_2 & 0 & -c\theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ = A_1 A_2 A_3$$

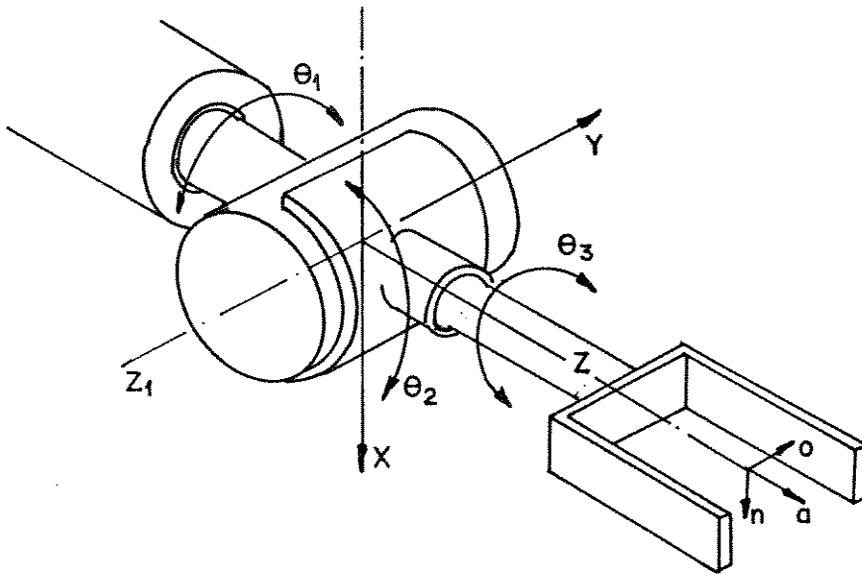


fig 2.6

2.4 EQUAÇÕES CINEMÁTICAS DE CONTROLE

O movimento do robô no seu espaço de trabalho é um movimento composto, parte realizado pelo seu conjunto primário e parte pelo seu conjunto secundário.

Desta forma,

(Movimento Total) = (Movimento do Braço) . (Movimento do Punho)

isto é,

$${}^0T = ({}^0T_{\text{braço}}) ({}^{\text{braço}}T_{\text{punho}})$$

para um robô de seis graus de liberdade

$${}^0T = (A_1 A_2 A_3) (A_4 A_5 A_6) \quad (\text{eq 2.2})$$

onde o superescrito 0 representa o sistema de coordenadas de referência, no caso a base e T descreve a posição do punho-ferramenta com respeito ao sistema de coordenadas da base.

Necessita-se então resolver a equação cinemática geral, eq2.2, em termos das variáveis da juntas, [4], [5], [6] de forma a que o sistema de controle possa atuar sobre elas.

A técnica empregada em [4] obtém a solução de maneira sequencial. Cada variável sucessiva é isolada a partir do punho-ferramenta pela pré-multiplicação de matrizes sucessivas.

Nota-se uma complexidade e uma grande frequência de multiplicações matriciais induzindo erros.

A solução composta permite a minimização destes problemas e um ganho computacional.

2.4.1. SOLUÇÃO PARA O PUNHO

Utilizando os ângulos de Euler (eq2.1)

$$T_E = A_1 A_2 A_3,$$

o que permite obter

$$\bar{A}_1^{-1} \cdot T_E = A_2 A_3$$

considerando

$$T_E = \begin{pmatrix} n_x & 0_x & a_x & 0 \\ n_y & 0_y & a_y & 0 \\ n_z & 0_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

resulta em:

$$\begin{pmatrix} c_1 n_x + s_1 n_y & c_1 0_x + s_1 0_y & c_1 a_x + s_1 a_y & 0 \\ -n_z & -0_z & -a_z & 0 \\ -s_1 n_x + c_1 n_y & -s_1 0_x + c_1 0_y & -s_1 a_x + c_1 a_y & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} c_2 c_3 & -c_2 s_3 & s_2 & 0 \\ s_2 c_3 & -s_2 s_3 & -c_2 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{eq 2.3})$$

igualando-se os termos respectivos da eq 2.3, obtemos $-s_1 a_x + c_1 a_y = 0$

$$\text{tg} \theta_1 = \frac{a_y}{a_x}, \quad \theta_1 = \text{tg}^{-1} \frac{a_y}{a_x} \quad (\text{eq 2.4})$$

similarmente:

$$\theta_2 = \text{tg}^{-1} \frac{c_1 a_x + s_1 a_y}{a_z} \quad (\text{eq 2.5})$$

$$\theta_3 = \operatorname{tg}^{-1} \left[\frac{-s_1 n_x + c_1 n_y}{-s_1 0_x + c_1 0_y} \right]$$

da eq 2.4

$$\operatorname{sen} \theta_1 = \frac{a_y}{\pm \sqrt{a_x^2 + a_y^2}}$$

e

$$\operatorname{cos} \theta_1 = \frac{a_x}{\pm \sqrt{a_x^2 + a_y^2}}$$

Substituindo na eq 2.5

$$\theta_2 = \operatorname{tg}^{-1} \frac{\pm \sqrt{a_x^2 + a_y^2}}{a_z}$$

2.4.2 SOLUÇÃO PARA O BRAÇO

A solução para um robô de revolução (tipo PUMA 560), permite mostrar que:

$$T_{\text{revolução}} = A_1 A_2 A_3 \quad \text{sendo:}$$

$$A_1 = \begin{pmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T_{\text{revolução}} = \begin{pmatrix} n_x & 0_x & a_x & p_x \\ n_y & 0_y & a_y & p_y \\ n_z & 0_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \bar{p} = \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}$$

resolvendo temos:

$$c_1 (a_3 c_{23} + a_2 c_2) = p_x$$

$$s_1 (a_3 c_{23} + a_2 c_2) = p_y$$

$$a_3 s_{23} + a_2 s_2 = p_z$$

onde s_{23} e c_{23} são respectivamente $\sin(\theta_2 + \theta_3)$ e $\cos(\theta_2 + \theta_3)$

A divisão $\frac{p_y}{p_x}$ permite obter

$$\theta_1 = \operatorname{tg}^{-1} \frac{p_y}{p_x}$$

Igualando p_x e p_y e somando

$$c_1^2 (a_3 c_{23} + a_2 c_2)^2 = p_x^2$$

$$s_1^2 (a_3 c_{23} + a_2 c_2)^2 = p_y^2$$

$$(a_3 c_{23} + a_2 c_2)^2 = p_x^2 + p_y^2$$

Fazendo-se $\alpha = \sqrt{p_x^2 + p_y^2}$

$$c_{23} = \frac{\alpha - a_2 c_2}{a_3} \quad (\text{eq 2.6})$$

como $s_{23}^2 + c_{23}^2 = 1$ e $s_{23} = \frac{p_z - a_2 s_2}{a_3}$ (eq 2.7)

obtemos

$$p_2 s_2 + \alpha c_2 = \frac{\alpha^2 + p_z^2 + a_2^2 - a_3^2}{2a_2}$$

Fazendo $\beta = \frac{\alpha^2 + p_z^2 + a_2^2 - a_3^2}{2a_2}$

temos:

$$p_z s_2 + \alpha c_2 = \beta \quad (\text{eq 2.8})$$

considerando

$$p_z = r \cos \phi$$

$$\alpha = r \sin \phi, \quad r = \sqrt{p_z^2 + \alpha^2}, \quad \phi = \text{tg}^{-1} \frac{\alpha}{p_z}$$

e substituindo na eq 2.8

$$r \cos \phi \sin \theta_2 + r \sin \phi \cos \theta_2 = \beta$$

$$\theta_2 = \text{tg}^{-1} \frac{\pm \beta}{\sqrt{p_z^2 + \alpha^2 - \beta^2}} - \text{tg}^{-1} \frac{\alpha}{p_z}$$

Para obter a solução de θ_3 , dividimos eq 2.7 por 2.6,

$$\frac{s_{23}}{c_{23}} = \frac{\frac{p_z - a_2 s_2}{a_3}}{\frac{\alpha - a_2 c_2}{a_3}}$$

$$\text{tg}(\theta_2 + \theta_3) = \frac{p_z - a_2 s_2}{\alpha - a_2 c_2}$$

$$\theta_3 = \text{tg}^{-1} \frac{p_z - a_2 s_2}{\alpha - a_2 c_2} - \theta_2$$

2.4.3 SOLUÇÃO PARA O ROBÔ

Utilizando a técnica da composição (eq 2.2),

$${}^0T_6 = (A_1 \ A_2 \ A_3) (A_4 \ A_5 \ A_6)$$

onde

$${}^0T_3 = A_1 \ A_2 \ A_3 = {}^0T_{\text{braço}}$$

e

$${}^3T_6 = A_4 \ A_5 \ A_6 = \text{braço}T_{\text{punho}}$$

$${}^0T_6 = \begin{pmatrix} N_x & O_x & A_x & P_x \\ N_y & O_y & A_y & P_y \\ N_z & O_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left[\begin{array}{c|c} \text{MOR} & \text{VP} \\ \hline 0 & 1 \end{array} \right]$$

$\text{MOR}_{3 \times 3}$ = matriz de orientação

$\text{VP}_{3 \times 1}$ = vetor de posição

$${}^0T_3 = \begin{pmatrix} n_{3x} & o_{3x} & a_{3x} & p_{3x} \\ n_{3y} & o_{3y} & a_{3y} & p_{3y} \\ n_{3z} & o_{3z} & a_{3z} & p_{3z} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left[\begin{array}{c|c} \text{MOR}_{3\phi} & \text{VP}_{3\phi} \\ \hline 0 & 1 \end{array} \right]$$

$${}^3T_6 = \begin{pmatrix} n_{6x} & o_{6x} & a_{6x} & p_{6x} \\ n_{6y} & o_{6y} & a_{6y} & p_{6y} \\ n_{6z} & o_{6z} & a_{6z} & p_{6z} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \left(\begin{array}{c|c} \text{MOR}_{63} & \text{VP}_{63} \\ \hline 0 & 1 \end{array} \right)$$

COMO

$${}^0T_6 = {}^0T_3 \cdot {}^3T_6$$

$$= \left(\begin{array}{c|c} \text{MOR}_{30} & \text{VP}_{30} \\ \hline 0 & 1 \end{array} \right) \cdot \left(\begin{array}{c|c} \text{MOR}_{63} & \text{VP}_{63} \\ \hline 0 & 1 \end{array} \right)$$

$$= \left(\begin{array}{c|c} \text{MOR}_{30} \text{ MOR}_{63} & \text{MOR}_{30} \text{VP}_{63} + \text{VP}_{30} \\ \hline 0 & 1 \end{array} \right)$$

A parte rotacional será

$$R = \text{MOR}_{30} \text{ MOR}_{63}$$

e a parte translacional

$$P = \text{MOR}_{30} \text{VP}_{63} + \text{VP}_{30} \text{ (eq 2.9)}$$

A matriz resultante $MOR_{30} VP_{63}$ é composta como:
 MOR_{30} = matriz de orientação, conectando os sistemas de coordenadas 3 e 0.

A sua pós-multiplicação por VP_{63} converte-a na equivalente com referência ao sistema de coordenadas 0.

Desta maneira $MOR_{30} VP_{63} = VP_{60}$, a posição do centro do punho, com referência ao sistema 0.

A eq 2.9, pode ser escrita,

$$\bar{P} = VP_{30} + VP_{60}$$

. Solução para os ângulos $\theta_1, \theta_2, \theta_3$ de controle do braço manipulador

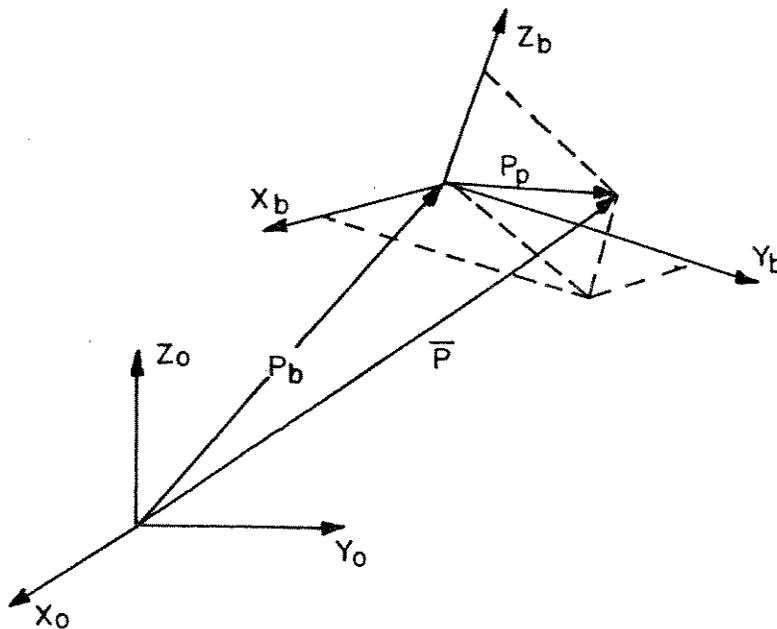


fig 2.7

O exame da translação e da rotação do braço manipulador, analisando-se a parte da translação inicialmente, mostra que, fig 2.7 :

(translação total) = (translação do braço) + (translação do punho)

$$\bar{P} = P_{\text{braço}} + P_{\text{punho}} \quad (\text{eq 2.10})$$

$$\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} = \begin{pmatrix} P_{\text{braço},x} \\ P_{\text{braço},y} \\ P_{\text{braço},z} \end{pmatrix} + \begin{pmatrix} P_{\text{punho},x} \\ P_{\text{punho},y} \\ P_{\text{punho},z} \end{pmatrix}$$

A fig 2.7 mostra a relação entre P , $P_{\text{braço}}$, P_{punho} , sendo que P_{punho} pode ser expresso em termos de direção do vetor A , e magnitude de p_{punho} , fig 2.8.

O ângulo de azimuth, ϕ , e o ângulo polar, θ , do vetor A com respeito ao sistema de coordenadas de base, pode ser determinado como:

$$\phi = \text{tg}^{-1} \frac{A_y}{A_x}$$

$$\theta = \text{tg}^{-1} \frac{\sqrt{A_x^2 + A_y^2}}{A_z}$$

e

$$P_{\text{punho},x} = |P_{\text{punho}}| \text{sen}\theta \cos\phi$$

$$P_{\text{punho},y} = |P_{\text{punho}}| \text{sen}\theta \text{sen}\phi$$

$$P_{\text{punho},z} = |P_{\text{punho}}| \text{cos}\theta$$

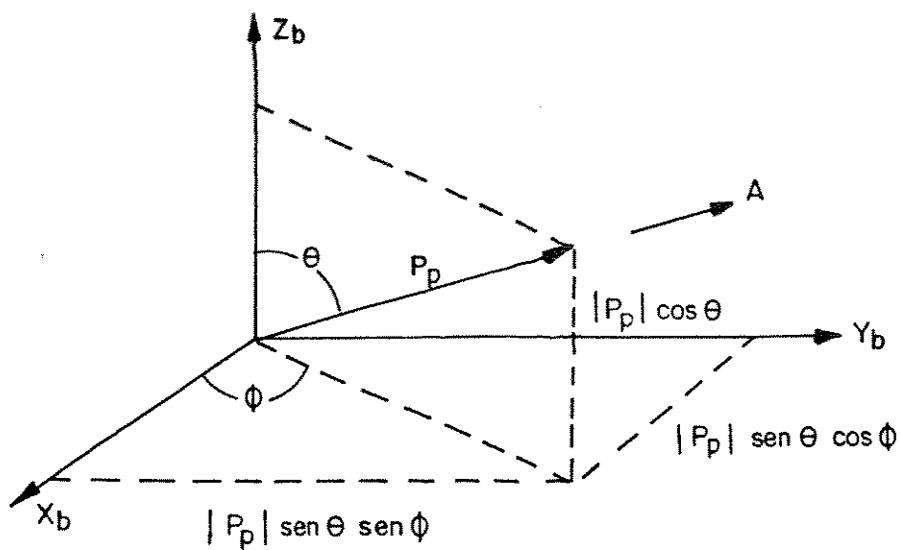


fig 2.8

Da eq 2.10, temos

$$P_{\text{braço}} = \bar{P} - P_{\text{punho}}$$

ou

$$\begin{pmatrix} P_{\text{braço},x} \\ P_{\text{braço},y} \\ P_{\text{braço},z} \end{pmatrix} = \begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix} - \begin{pmatrix} P_{\text{punho}} \begin{vmatrix} \text{sen}\theta \cos\phi \\ \text{sen}\theta \text{sen}\phi \\ \cos\theta \end{vmatrix} \end{pmatrix}$$

O vetor $P_{\text{braço}}$ é medido a partir da origem ao ponto de conexão braço/punho.

É interessante notar que o robô conforme visto na fig 2.7 não apresenta translação na ponta do braço.

Considerando um deslocamento adicional, ao longo do eixo z, por exemplo d_4 temos:

$$P_{\text{braço}} = {}^0T_{\text{braço}} \begin{pmatrix} 0 \\ 0 \\ d_4 \\ 1 \end{pmatrix}$$

ou

$$P_{\text{braço}} = \begin{pmatrix} c_1 s_2 s_3 d_4 + c_1 c_2 a_2 - s_1 d_2 \\ s_1 s_2 s_3 d_4 + s_1 c_2 a_2 + c_1 d_2 \\ c_2 s_3 d_4 - a_2 s_2 \\ 1 \end{pmatrix}$$

Igualando respectivamente

$$c_1 s_{23} d_4 + c_1 c_2 a_2 - s_1 d_2 = P_{\text{braço},x} = P_{b,x}$$

$$s_1 s_{23} d_4 + s_1 c_2 a_2 + c_1 d_2 = P_{\text{braço},y} = P_{b,y}$$

$$c_{23} d_4 - a_2 s_2 = P_{\text{braço},z} = P_{b,z}$$

multiplicando ambos os lados da equação por $-s_1$ e c_1 , e adicionando-as temos

$$-s_1 c_1 s_{23} d_4 - s_1 c_1 c_2 a_2 + s_1^2 d_2 = -s_1 P_{b,x}$$

$$c_1 s_1 s_{23} d_4 + s_1 c_1 c_2 a_2 + c_1^2 d_2 = c_1 P_{b,y}$$

$$P_{b,x} s_1 + P_{b,y} c_1 = d_2$$

que é da forma $-As\theta + Bc\theta = D$

como

$$\theta = \operatorname{tg}^{-1} \frac{B}{A} - \operatorname{tg}^{-1} \frac{\pm D}{\sqrt{A^2 + B^2 - D^2}}$$

temos

$$\theta_1 = \operatorname{tg}^{-1} \frac{P_{b,y}}{P_{b,x}} - \operatorname{tg}^{-1} \frac{\pm d_2}{\sqrt{P_{b,x}^2 + P_{b,y}^2 - d_2^2}}$$

novamente,

Igualando os valores respectivos

$$c_1 s_{23} d_4 + c_1 c_2 a_2 - s_1 d_2 = P_{\text{braço},x} = P_{b,x}$$

$$s_1 s_{23} d_4 + s_1 c_2 a_2 + c_1 d_2 = P_{\text{braço},y} = P_{b,y}$$

$$c_{23} d_4 - a_2 s_2 = P_{\text{braço},z} = P_{b,z}$$

Multiplicando ambos os lados das equações por s_1 e c_1 e adicionando-as temos

$$\theta_2 = \text{tg}^{-1} \frac{\alpha}{P_{bz}} - \text{tg}^{-1} \frac{\pm\beta}{\sqrt{\alpha^2 + P_{bz}^2 - \beta^2}}$$

onde

$$\beta = \frac{\alpha^2 + a_2^2 + P_{bz}^2 - d_4^2}{2a_2}$$

$$\alpha = P_{bx} c_1 + P_{by} s_1$$

e

$$\theta_3 = \text{tg}^{-1} \frac{\alpha - a_2 c_2}{P_{bz} + a_2 s_2} - \theta_2$$

. Solução para os ângulos $\theta_4, \theta_5, \theta_6$ de controle do punho.

A rotação total da mão ferramenta é o produto de duas sub- rotações i.e,

(rotação total) = (rotação do braço) (rotação do punho)

$$R = {}^0R_{\text{braço}} \cdot {}^3R_{\text{punho}}$$

$${}^3R_{\text{punho}} = {}^0R_{\text{braço}}^{-1} \cdot R \quad (\text{eq 2.11}),$$

onde ${}^3R_{\text{punho}}$ é a rotação em termos dos ângulos de Euler,

$${}^3R_{\text{punho}} = \begin{pmatrix} n_{\text{punho},x} & 0_{\text{punho},x} & a_{\text{punho},x} \\ n_{\text{punho},y} & 0_{\text{punho},y} & a_{\text{punho},y} \\ n_{\text{punho},z} & 0_{\text{punho},z} & a_{\text{punho},z} \end{pmatrix}$$

$$\begin{pmatrix} n_{p,x} & 0_{p,x} & a_{p,x} \\ n_{p,y} & 0_{p,y} & a_{p,y} \\ n_{p,z} & 0_{p,z} & a_{p,z} \end{pmatrix} = \begin{pmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 \\ -s_5c_6 & s_5s_6 & c_5 \end{pmatrix}$$

A inversa de ${}^0R_{\text{braço}}$

$${}^0R_{\text{braço}} = \begin{pmatrix} c_1c_{23} & s_1c_{23} & -s_{23} \\ -s_1 & c_1 & 0 \\ c_1s_{23} & s_1s_{23} & c_{23} \end{pmatrix}$$

e substituindo-se na eq 2.11 obtem-se

$$\theta_4 = \operatorname{tg}^{-1} \frac{a_{\text{punho},y}}{a_{\text{punho},x}}$$

ou $\theta_4 = \theta_4 + 180^\circ$ e

$$\theta_5 = \operatorname{tg}^{-1} \frac{c_4 a_{\text{punho},x} + s_4 a_{\text{punho},y}}{a_{\text{punho},z}}$$

e
$$\theta_6 = \operatorname{tg}^{-1} \frac{-s_4 n_{\text{punho},x} + c_4 n_{\text{punho},y}}{-s_4 o_{\text{punho},x} + c_4 o_{\text{punho},y}}$$

2.5 RESUMO

Neste capítulo, além da introdução à robótica, foi analisado o aspecto da modelagem cinemática do robô industrial de forma metódica.

Onde a orientação é fundamental, para o caso do sistema punho/mão ferramenta, a multiplicação inversa permite o isolamento das variáveis de controle.

Onde é fundamental o posicionamento, cada variável do braço foi explicitada.

De um modo geral os itens analisados foram:

- a) determinação dos ângulos polares e azimutais θ e ϕ do conjunto punho/mão ferramenta;
- b) determinação dos componentes do vetor $P_{\text{braço}}$, usando θ e ϕ ;
- c) ângulos θ_1, θ_2 e θ_3 de controle do braço, usando os componentes do vetor $P_{\text{braço}}$, foram determinados e
- d) variáveis de controle, $\theta_4, \theta_5, \theta_6$ do conjunto punho/mão ferramenta foram resolvidas, usando-se os ângulos $\theta_1, \theta_2, \theta_3$.

CAPÍTULO 3 - MODELAGEM DINÂMICA DE
ROBÔ INDUSTRIAL

CAPÍTULO 3

MODELAGEM DINÂMICA DE ROBÔ INDUSTRIAL

3.1 INTRODUÇÃO

Este capítulo apresenta inicialmente as formulações teóricas das equações diferenciais de movimento.

A opção por uma destas formulações (Lagrange-Euler), define a forma a ser tomada pela modelagem dinâmica.

Em seguida é definida uma notação matricial, baseada na ferramenta computacional de combinadores, introduzida em linguagem tipo FP ("Functional Programming").

O objetivo, utilizando os itens anteriormente definidos, é a concretização da implementação de um programa gerador do modelo dinâmico de robô industrial, o qual é escrito em linguagem PROLOG.

3.2 AS EQUAÇÕES DIFERENCIAIS DE MOVIMENTO

As informações fundamentalmente necessárias na modelagem da dinâmica de um robô, são conseguidas a partir das equações diferenciais de movimento.

Estas tem origem basicamente em quatro diferentes formulações: a de Lagrange-Euler, de Newton-Euler, de Lagrange-recursivo e a de D'Alembert generalizado.

Destas, duas são usadas mais sistematicamente e obtem-se melhores resultados: Lagrange-Euler e Newton-Euler.

Em nosso estudo abordaremos a formulação de Lagrange-Euler, particularmente a apresentada através dos trabalhos realizados em [7] [8].

A equação dinâmica de movimento excluindo a dinâmica do dispositivo eletrônico de controle e a fricção da transmissão, é um conjunto de equações diferenciais aplicadas de segunda ordem, não linear.

No caso de um manipulador com seis graus de liberdade, é mostrado [8] que ela é altamente não linear, consistindo de cargas inerciais, reações de acoplamento entre juntas (Coriolis e centrífuga) e efeitos de carga gravitacional.

De um modo geral, cada termo de torque ou força gravitacional e inercial depende da configuração (posição) instantânea dos ligamentos:

Por outro lado, os termos de torque ou força de reação gerados pela velocidade dependem também da velocidade instantânea.

A aplicação de formulação de Lagrange-Euler, conjuntamente com a representação matricial de Denavit-Hartenberg [1] resulta em um algoritmo compacto, descrevendo a equação de movimento do manipulador.

3.11 A FORMULAÇÃO DE LAGRANGE-EULER

As equações de movimento de L-E,

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i, \quad i = 1, 2, \dots, n \quad (\text{eq 3.1})$$

$$L = K - P$$

onde

L = Função Lagrangeana = Energia Cinética (K) - Energia Potencial (P),

K = Energia cinética total do sistema,

P = Energia potencial total do sistema,

q_i = Coordenada generalizada do sistema

$q_i = \theta_i$ para juntas rotacionais

$q_i = d_i$ para juntas translacionais

τ_i = Força (ou torque) generalizada aplicado ao sistema

\dot{q}_i = Derivada primeira em relação ao tempo da coordenada generalizada q_i

Então

$$L = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^i \sum_{k=1}^i T_r (U_{ij} J_i U_{ik}^t) \dot{q}_j \dot{q}_k +$$

$$+ \sum_{i=1}^n m_i g (A_0^i \bar{r}_i)$$

visto que,

$$P_i = -m_i g r_0^i = -m_i g (A_0^i \bar{r}_i), \quad i = 1, 2, \dots, n$$

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n T_r \left[\sum_{p=1}^i \sum_{r=1}^i U_{ip} J_i U_{ir}^t \dot{q}_p \dot{q}_r \right]$$

$$= \frac{1}{2} \sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \left[T_r (U_{ip} J_i U_{ir}^t) \dot{q}_p \dot{q}_r \right]$$

resultando em

$$\tau_i = \sum_{j=1}^n \sum_{k=1}^j T_r (U_{jk} J_j U_{ji}^t) \ddot{q}_k +$$

$$+ \sum_{j=1}^n \sum_{k=1}^j \sum_{m=1}^j T_r (U_{jkm} J_j U_{ji}^t) \dot{q}_k \dot{q}_m -$$

$$- \sum_{j=1}^n m_j g U_{ji} \bar{r}_j, \quad i = 1, 2, \dots, n$$

ou

$$\tau_i = \sum_{k=1}^n D_{ik} \ddot{q}_k + \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m + G_i \quad (\text{eq 3.2})$$

para $i = 1, 2 \dots n$

As equações de Lagrange-Euler, eq 3.2, em uma forma matricial compacta [7] apresentam-se:

$$\tau = D(\theta)\ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta), \quad (\text{eq 3.3})$$

onde

. τ é o vetor de torque ou força generalizada aplicado aos atuadores das juntas

. D é a matriz, $n \times n$, simétrica, de aceleração inercial, onde

$$D_{ik} = \sum_{j=\text{MAX}(i,k)}^n \text{Tr} (U_{jk} J_j U_{ji}^t), \quad \text{para} \quad (\text{eq 3.4})$$

$$i, k = 1, 2 \dots n$$

. $H(\theta, \dot{\theta})$ é o vetor $n \times 1$ de força centrífuga ou de Coriolis onde

$$H(\theta, \dot{\theta}) = (H_1, H_2, \dots, H_n)^t \text{ e}$$

$$H_i = \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m \text{ para } i=1, 2, \dots, n \text{ (eq 3.5)}$$

$$H_{ikm} = \sum_{j=\text{MAX}(i,k,m)}^n \text{Tr}(U_{jkm} J_j U_{ji}^t) \text{ (eq 3.6)}$$

para $i, k, m = 1, 2, \dots, n$

. $G(\theta)$ é um vetor $n \times 1$ de carga gravitacional,

$$G(\theta) = (G_1, G_2, \dots, G_n)^t \quad e$$

$$G_i = \sum_{j=i}^n (-m_j g U_{ji} \bar{r}_j) \text{ para } i = 1, 2, \dots, n \text{ (eq 3.7)}$$

Nas eq. 3.4, 3.6 e 3.7 temos que

$$U_{ij} = \frac{\partial A_0^i}{\partial q_j} \text{ (eq 3.8)}$$

onde A_0^i é a matriz 4×4 , homogênea, das coordenadas de transformação que relaciona as coordenadas dos ligamentos da base (ligamento zero) as do ligamento i , definida como

$$A_0^i = A_0^1 A_1^2 \dots A_{i-2}^{i-1} A_{i-1}^i \text{ (eq 3.9)}$$

e A_{i-1}^i é a matriz 4x4, homogênea, genérica, das coordenadas de transformação que relaciona às do ligamento i-1.

Na eq 3.8, q_j é a coordenada generalizada do sistema e

$$U_{ij} = 0 \quad \text{para } j > i \quad (\text{eq 3.10})$$

Na eq 3.4 J_j é a matriz 4x4 de pseudo-inércia do ligamento j, e

$$J_j = \int r_j r_j^t dm \quad (\text{eq 3.11})$$

onde

$$r_j = (x_j, y_j, z_j, 1)^t$$

é o ponto fixo do ligamento j e expresso em coordenadas homogêneas com respeito ao j-ésimo sistema de coordenadas do sistema.

Na eq 3.6

$$U_{jkm} = \frac{\partial U_{jk}}{\partial q_m}$$

Na eq 3.7 m_j é a massa do ligamento j.

g é o vetor linha de gravidade isto é $g = (g_x, g_y, g_z, 0)$, expresso no sistema de coordenadas de base.

$\bar{r}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i, 1)^t$ é o centro de massa do ligamento i expresso em termos do i-ésimo sistema de coordenadas.

$$T_r A = \sum_{i=1}^n a_i i$$

$$(a, b, c)^t = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

n é o número de graus de liberdade do robô.

A_{i-1}^i é definida genericamente como

$$A_{i-1}^i = \begin{pmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (\text{eq 3.12})$$

3.3 NOTAÇÃO POR COMBINADORES

A notação matricial é uma maneira compacta de expressar as equações de movimento de Lagrange-Euler.

Ela é, porém, um sistema de notação baseado em matrizes com índices.

Um algoritmo computacionalmente eficiente para operar com matrizes, não o será utilizando iterações indexadas.

Isto se deve ao fato de que não é necessário um acesso aleatório ao elemento matricial, para realizar operações como multiplicação de matrizes.

As operações com matrizes podem então ser vistas como operações com sequências de números ou objetos e tratados como tal, de forma a tornarem-se eficientes.

Desta forma é necessária uma ferramenta para implementar as operações com matrizes e vetores.

A essa ferramenta denominaremos combinadores, introduzidos em [10], como um subconjunto de uma linguagem FP ("Functional Programming").

Esta ferramenta acrescida de definições de recursividade, facilita a manipulação de funções.

A álgebra particular dos combinadores linguagem FP permite:

- . programas com semântica extremamente simples
- . aparecimento frequente de expressão não recursiva para muitas funções definidas recursivamente.
- . programa com exibição de estrutura hierárquica clara.

Um programa FP consiste de:

- . um conjunto de funções primitivas (por exemplo, operadores aritméticos como +, -, ., etc)
- . um conjunto de realizadores de funções (PFOs), análogos a laços tipo while condicionais, etc.
- . um domínio de objetos como por exemplo inteiros, caracteres, sequências etc.
- . PFOs são operações pré-definidas em funções.

Uma operação condicional de FP como $f \rightarrow g:L$ é similar a operações conhecidas como `if...then...else` exceto pelo fato de que em FP, a operação envolve apenas as funções f , g e L , que podem ser primitivas, funções definidas em expressões construídas usando PFOs.

Em resumo, toda FPO utiliza um certo número de funções como argumento e retorna uma única função.

Toda função de FP, utiliza um objeto único como entrada e produz um resultado único, definindo a relação objeto \rightarrow objeto

Todo sistema de funções de FP, utiliza a operação aplicação, a qual, dada uma função e um objeto, define o resultado aplicação

cando a função ao objeto.

Assim a notação $f:x$ é usada para representar a aplicação da função f ao objeto x .

Definir um sistema FP, torna necessário especificar um conjunto de primitivas, um conjunto de PFOs e um conjunto de objetos.

Os conjuntos a seguir definidos são exemplos de primitivas presentes em um sistema FP e servem para a compreensão da implementação desenvolvida a seguir.

3.3.1 CONJUNTO DE PRIMITIVAS

SOMA, SUBTRAÇÃO, MULTIPLICAÇÃO, ETC (+, -, *, etc)

Quando x é da forma $\langle y, z \rangle$ e y, z são números
então $+$: x executa a soma de y e z
se não $+$: $x = \perp$

Caracter menor do que, menor do que ou igual, etc (mdq, mdq/e, etc)
Quando $x = \langle y, z \rangle$ e y, z são números
então se $y > z$ então mdq : $x = T$ se não mdq = F

Obs.

. as outras funções são definidas similarmente.

Função e, ou, não

Quando $x = \langle T, T \rangle$ então e: $x = T$

Quando $x = \langle F, T \rangle$ ou $x = \langle T, F \rangle$

ou $x = \langle F, F \rangle$, então e : $x = F$

Função nulo (lista vazia)

Quando $x = \langle \rangle$ então nulo : $x = T$

Quando $x = \perp$ então nulo : $x = \perp$

de outro modo nulo : $x = F$

Função coloca à esquerda (coloca à esquerda - ce)

Quando $x = \langle y, \langle \rangle \rangle$, então $ce : x = \langle y \rangle$

Quando $x = \langle y, \langle z_1, z_m \rangle \rangle$,

então $ce : x = \langle y, z, z_m \rangle$

Função coloca à direita (coloca à direita - cr)

Quando $x = \langle \langle \rangle, y \rangle$, então $cr : x = \langle y \rangle$

Quando $x = \langle \langle z_1, z_m \rangle, y \rangle$,

então $cr : x = \langle z_1 \dots z_m, y \rangle$

Função Tranpor (trans)

Quando $x = \langle \langle \rangle, \dots \langle \rangle \rangle$ então $trans : x = \langle \rangle$

Quando $x = \langle x_1, \dots, x_m \rangle$, onde $x_i = \langle x_{i1}, \dots, x_{ik} \rangle$

para $1 \leq i \leq m$ então $trans : x = \langle z_1, z_k \rangle$

onde $z_j = \langle x_{1j}, \dots, x_{mj} \rangle$ para $1 \leq j \leq k$

se não $trans : x = \perp$

Distribui à esquerda (de)

Quando $x = \langle y, \langle \rangle \rangle$ então $de : x = \langle \rangle$

Quando $x = \langle y, \langle z_1 \dots z_m \rangle \rangle$,

então $de : x = \langle \langle y, z_1 \rangle \dots \langle y, z_m \rangle \rangle$

Distribui à direita (dr)

Quando $x = \langle \langle \rangle, y \rangle$ então $dr : x = \langle \rangle$

Quando $x = \langle \langle z_1 \dots z_m \rangle, y \rangle$

então $\langle \langle z_1, y \rangle, \dots, \langle z_m, y \rangle \rangle$

Obs:

- . outras funções são definidas similarmente
- . o caracter \perp significa objeto indefinido ou erro.

Também PFOs são definidos como:

Função Composição

$(f \circ g) : x = f : (g : x)$

Função Construção

$$(f_1, f_2, \dots, f_m) : x = \langle f_1 : x, f_2 : x, \dots, f_m : x \rangle$$
Função Condição

$$(p \rightarrow f : g) : x \text{ se } (p:x) \text{ é T então } f : x$$

$$\text{se não se } (p : x) \text{ é F então } g : x$$

$$\text{se não } \perp$$
Função Aplica a Todo (apt)

$$\alpha f : x = \text{se } \langle \rangle \text{ então } \langle \rangle$$

$$\text{se } x = y_1 \dots y_m, \text{ então } \langle f : y_1 \dots f : y_m \rangle$$

3.3.2 OPERAÇÕES BÁSICAS COM COMBINADORES

A utilização da ferramenta dos combinadores em conjunto com a linguagem Prolog fornecerá vários instrumentos básicos observadas as seguintes considerações:

- . Vetores serão representados por listas.
- . Matrizes serão listas dos vetores.
- . Dados dois vetores, o predicado (denominado Comb) fornece uma lista de pares.
- . Cada par contém os elementos de dois vetores indexados igualmente.

ou seja:

$$? - \text{comb} ([2, 5, a, d], [3, g, 4, 0], c).$$

$$c = ([2, 3], [5, g], [a, 4], [d, 0])$$

genêricamente comb é:

$$\text{comb} ([], [], []) : - !.$$

$$\text{comb} ([H1/T1], [H2/T2], [[H1, H2] / T12]) : -$$

$$\text{comb} (T1, T2, T12).$$

- o predicado de (distribui à esquerda) e dr (distribui à direita) formam todos os pares contendo um objeto (sempre o mesmo) e um elemento do vetor, ou seja:

? - de (3, [a,b,c], X)
 x = [3, a], [3, b], [3, c]
 ? - dr (3, [a, b, c], X)
 x = [a, 3], [b, 3], [c, 3]

É possível dizer que de distribui o objeto à esquerda dos elementos do vetor, assim com dr o faz a direita.

de modo geral:

de (x, [], []): - ! .
 de (x, [H/T], [[x,H]/XT]) : - dr (X,T,XT) .
 dr (X, [], []): - ! .
 dr (X, [H/T], [[H,X]/TX]): - dr (X,T,TX) .

- o predicado aplica-a-todo (apt) aplica uma função especificada a todo elemento de um vetor. De modo geral,

apt (F_n, [], []): - ! .
 apt (F_n, [Y/Z], [F_nY/F_nZ]) : -
 ap (F_n, Y, F_nY),
 apt (F_n, Z, F_nZ) .

onde F_n é a função especificada e ap é um predicado com três argumentos: o nome de uma função, um objeto e o resultado da aplicação da função ao objeto, por exemplo:

ap (quadrado, X, XX) : - XX é X * X

particularizando:

? - apt (quadrado, [2, 4, 5], Q)

Q = [4, 16, 25].

. o predicado red acumula o resultado da aplicação de uma função aos argumentos de cada dois elementos de um vetor, i.e.,

? - red (+, [2, 5, 3, 4], X).

X = 14

de modo geral:

red (F_n , [Y, Z], R) : - ap(F_n , [Y, Z], R), ! .

red (F_n , [Y/Z], R) : - red(F_n , Z, RZ),

ap (F_n , [Y, RZ], R), ! .

Usando estas primitivas, é possível realizar-se manipulações algébricas, como:

a) transposição:

ap(trans, X, []) : - linha-nula (X), ! .

ap(trans, M, [TL/TR]) : - apt (cabeça, M, TL),

apt (cauda, M, TM),

ap (trans, TM, TR).

ap (cabeça, [X/T], X).

ap (cauda, [X/T], T).

linha-nula ([]): - !.

linha-nula ([]/T): - linha-nula (T).

b) produto escalar de vetores:

pode ser dado por:

ap (produto-escalar, [V₁, V₂], V): -

comb (V₁, V₂, CV),

apt (*, CV, MV),

red (+, MV, V).

A atuação efetiva do predicado ap tem como condição fundamental a definição de clausulas de Horn, para "ap(+,...)" e "ap(*,...)".

Isto porque torna-se necessária a realização de operações elementares e simplificações.

Para cada um desses predicados, existem mais de 100 clausulas de Horn, algumas delas como:

ap (+, [0,X], X) :- true.

ap (+, [log(X), log(Y)], Z) :-

ap (*, [X, Y], Z₁),

ap (log, Z₁, Z).

ap (+, [X, Y], R): - R is X + Y, R \= err.

ap (*, [X, 0], 0).

ap (*, [1, X], X).

```

ap (*, [X, X ^(-1)], 1).
ap (*, [X, Y], Z): - Z is X * Y, Z \= err.
ap (*, [-X, -Y], X * Y).

```

.

.

.

Com o conhecimento de como realizar o produto escalar de vetores, não há dificuldade para definir o produto de um vetor por uma matriz,

```

ap (vXm, [V, M], VM): -
de (V, M, CMV),
apt(escalar, CMV, VM).

```

A definição de um produto matricial, segue,

```

ap (mXm, [M1, M2], MM): -
ap (trans, M2, M3), ! ,
dr (M3, M1, CM), ! ,
apt(vXm, CM, MM).

```

O predicado `ap`, como visto, pode ter milhares de clausulas; deste modo torna-se necessário a indexação do primeiro argumento para torná-lo eficiente.

Ou então, armazenar as clausulas de `ap` em uma tabela de consulta.

Considerando-se a implementação em Arity Prolog, isto pode ser feito definindo-se o predicado:

```

ap (Fn, Arg, Res): - recorded (Fx, [Arg, Res, Corpo], Ref),
Corpo.

```

como exemplo podemos ter:

```

recordz (+, [[0,X], X, true], Ref).
recordz (+, [[log(X), log(Y)], Z,
(ap (*, [X, Y], Z1); ap(log, Z1, Z))], Ref).
recordz(+, [[X,Y], R, R is X + Y, R = err]], Ref).

```

O predicado recordz grava um determinado objeto no final da base de dados.

3.3.3 IMPLEMENTAÇÃO

A partir dos parâmetros de Denavit-Hartenberg (DHP) para cada ligamento, a geração das matrizes A_{i-1}^i , $i=1, 2, \dots, n$ são realizadas por:

```

gerador-de-matrizes-Ai ([ ], [ ]):-!.
gerador-de-matrizes-Ai ([DHP/Outro-DHP],[ Ai/Outro Ai] ):-
monta-matriz-Ai (DHP, Ai),
gerador-de-matrizes-Ai (Outro-DHP, Outro Ai).

```

onde o predicado monta-matriz-A_i, como o próprio nome diz, monta as matrizes A_{i-1}^i uma por vez, conforme definida na eq 3.12.

Uma vez operada a lista de matrizes A_{i-1}^i , são geradas as matrizes A_0^i conforme definido na eq. 3.9, usando-se os predicados:

```

gerador-de-matrizes-A0 ([ ], [ ], [ ]):-!.
gerador-de-matrizes-A0([Ai/Outro-Ai], Aux, [ A0/Outro-A0 ]):-
ap(mXm,[Aux, Ai], A0),
gerador-de-matrizes-A0 (Outro-Ai, A0, Outro-A0).

```

É necessário observar que Aux no gerador-de matrizes-A ϕ , em sua primeira chamada é a matriz identidade.

A partir das matrizes A_0^i , $i = 1, 2 \dots n$ das listas de variáveis para cada junta, são geradas as matrizes U_{ij} para $j \leq i$, desde que $U_{ij} = 0$ para $j > i$.

Assim os predicados que se seguem, permitem a determinação de U_{ij} ,

gerador-de-matrizes-U ([], [], []): - ! .

gerador-de-matrizes-U([A₀/Outro-A₀], [V/outro-V],[Lista-U/Outra-Lista-U]):-

de (V, [A₀/Outro-A₀], VA₀),

apt(deriva-matriz, VA₀, Lista-U),

gerador-de-matrizes-U(Outro-A,Outro-V,Outra-Lista-U).

O resultado da aplicação de predicado gerador-de-matrizes-U, é na realidade uma lista de U_{ij} , tal que:

Lista de $U_{ij} = [[U_{11}, U_{21} \dots U_{n1}], [U_{22}, \dots U_{n2}], \dots [U_{nn}]]$

Uma vez geradas as matrizes U_{ij} , o próximo passo será a execução da mesma etapa para as matrizes $D(\theta)$, cujos elementos estão definidos em (eq 3.4).

A expansão da eq 3.4, permite observar a realização das operações de traço e somatório, gerando D_{ik} .

Utilizando todo o U_{ij} já gerado e tendo J_i como entrada podemos construir o gerador das matrizes D_{ik} .

gera-linhas-da-matriz-D(-, [], [], []):-! .

gera-linhas-da-matriz-D(Sc,[RTI/ORTI], [LRU/OLRU] [LRD/OLRD]):-

calcula-linha-de-DS(Sc,[RTI/ORTI],[LRU/OLRU],LRD),

X is Sc + 1

gera-linhas-da-matriz-D(X, ORTI, OLRU, OLRD).

calcula-linha-de Ds(__, [], [], []) :-!.

calcula-linha-de-DS(Sc, [RTI/ORTI], [LRU/OLRU], [RD/ORD]):-

gera-matrizes-JU(Sc, [RTI/ORTI], LRU, LRJU),

gera-matrizes-UJU(LRU, LRJU, LRUJU),

aptodo(Traço-uju, LRUJU, Traços),

red(+, Traços, D), [D] = Traços,

recordz(d, D, RD),

calcula-linha-de-DS(Sc, ORTI, OLRU, ORD), !.

gera-matrizes-UJU(LRU, LRJU, LRUJU):-

comb(LRU, LRJU, Comb),

aptodo(mXm-uju, Comb, LRUJU), !.

gera-matrizes-JUt(Sc, __, __, LRJU):-

recorded(ju, Sc, Ref),

nref(Ref, NRef),

erase(NRef),

recupera(Ref, LRJU), !.

gera-matrizes-JU(Sc, LRTI, LRU, LRJU): -

comb(LRTI, LRU, Comb),

recordz(ju, Sc, __),

aptodo(mXm-ju, Comb, LRJU),

grava(jut, LRJU, Re),

grava(jul, LRJU, __).

Na tabela 3.1, pode ser analisada a expansão da equação (3.4) esquematicamente, considerando-se a estrutura gerada da lista para as U_{ij} e para as matrizes de pseudo-inércia J_i ,

		$J_1 U_{11}^t$	$J_2 U_{21}^t$...	$J_n U_{n1}^t$
D_{11}	U_{11}	*			
	U_{21}		*		
	\vdots			\ddots	
	U_{n1}				*
D_{12}	U_{22}		*		
	\vdots			\ddots	
	U_{n2}				*
\vdots	...				
D_{1n}	U_{nn}				*

		$J_2 U_{22}^t$...	$J_n U_{n2}^t$
D_{22}	U_{22}	*		
	\vdots		\ddots	
	U_{n2}			*
\vdots	...			
D_{2n}				*

		$J_n U_{nn}^t$
D_{nn}	U_{nn}	*

tabela 3.1

```

recupera (Ref, [RJU/ORJU]):-
nref (Ref, RJU),
instance (RJU, Obj),
nonvar (Obj), not integer (Obj),
recupera (RJU, ORJU).

```

```

monta-matriz-D ([Penultima-LRD,Última-LRD], MRD):-
ap (cauda, Penultima-LRD, [RD]),
ae (RD, última-LRD, Aux),
ae (Penultima-LRD, [Aux], MRD),!.

```

```

monta-matriz-D ([LRD/OLRD], MRD):-
monta-matriz-D (OLRD, Aux),
ap (cauda, LRD, Resto-LRD),
embaralha (Resto-LRD, Aux, Lista),
ae (LRD, Lista, MRD).

```

```

embaralha ([ ], [ ], [ ]):-
embaralha ([H/T], [[X/Y]/W], [[H, X/Y] / Z):-
embaralha (T, W, Z).

```

O predicado `monta-matriz-D`, realiza a montagem da lista D_{ik} conforme a tabela 3.1, levando em consideração que $k \geq 1$, uma vez que $D(\theta)$ é simétrica.

O cálculo do vetor de força centrífuga ou de Coriolis, $H(\theta, \dot{\theta})$, através de H_i , dado pela eq 3.5 torna necessário inicialmente a solução de H_{ikm} , dado pela eq. 3.6.

Como no caso da matriz $D(\theta)$, a expansão realizada é esquematizada pela tabela 3.2, conforme segue:

		$J_1 U_{11}^t \quad J_2 U_{21}^t \quad \dots \quad J_n U_{n1}^t$			
		U_{111}	*		
H_{111}		U_{211}		*	
		\vdots			
		U_{n11}			*
		...			
H_{112}		U_{212}		*	
		\vdots			
		U_{n12}			*
		...			
H_{11n}		U_{n1n}			*
		...			
		U_{221}		*	
		\vdots			
H_{121}		U_{n21}			*
		...			
		U_{n2n}			*
		...			
H_{1nr}		$U_{n rn}$			*
		...			
		U_{222}		*	
		\vdots			
H_{221}		U_{n22}			*
		...			
		U_{n2n}			*
		...			
H_{2nr}		$U_{n rn}$			*
		...			
		U_{22n}			*
		...			
H_{nrn}		$U_{n rn}$			*
		...			
		$U_{n n n}$			*
		...			

tabela 3.2

Como foi visto anteriormente a solução de H_{ikm} conforme a eq 3.6, necessita dos vetores de U_{jkm} , caracterizado como sistema de derivadas de $U(SDU)$.

```
gera-lista-de-SDU([ ], [ ], [ ]):-!.
```

```
gera-lista-de-SDU([LRU/OLRU],[V/VV],[LSDU/OLSDU]):-
```

```
gera-lista-de-SDU1(LRU,[V/VV],LSDU),
```

```
gera-lista-de-SDU(OLRU,VV,OLSDU).
```

```
gera-lista-de-SDU1([ ], [ ], [ ]):-!.
```

```
gera-lista-de-SDU1([LRUA/LRUB],[V/VV],[LSDUA/LSDUB]):-
```

```
de(V,[LRUA/LRUB],VLRU),
```

```
aptodo(deriva-matriz-heter,VLRU,LSDUA),
```

```
gera-lista-de-SDU1(LRUB,VV,LSDUB).
```

A próxima etapa realiza a geração final dos termos da força de Coriolis, através dos predicados,

```
calcula-linha-de-HS(__,[ ], [ ], [ ]):-!.
```

```
calcula-linha-de-HS(Sc,[LSDU/OLSDU],[LJUt/OLJUt],[HS/OHS]):-
```

```
calcula-H([LSDU/OLSDU],LJUt,HS),
```

```
processa-lista-de-SDU(Sc,[LSDU/OLSDU],[NSDU/OLSDU]),
```

```
W is Sc + 1
```

```
calcula-nova-linha-de-HS(W,[NSDU/OLSDU],OLSJUt,OHS).
```

```
calcula-nova-linha-de-HS(__,[ ], __,[ ]):-!.
```

```
calcula-nova-linha-de-HS(W,[NSDU/OLSDU],OLJUt,[N/OHS]):-
```

```
calcula-nova-linha(NSDU,OLJUt,H),
```

```
calcula-nova-linha-nova-de-HS(W,OLSDU,OLJUt,OHS)
```

```

calcula-nova-linha-nova-de-H (__,[ ], [ ], [ ]):-!.
calcula-nova-linha-nova-de-HS(__,[OLSDUA/OLSDUB],OLJUT,OHS):-
calcula-nova-linha-nova-de-HHS(__,OLSDUA, OLJUT, OHS).

```

```

calcula-nova-linha-nova-de-HHS(__, [ ], [ ], [ ]):-!.
calcula-nova-linha-nova-de-HHS(__,[OLSDUA/OLSDUB] [OLJUtA/OLJUtB],OHS:-
comb (OLSDUA, OLJUtA, Comb1),
aptodo (mXm_hs, Comb1, DUJUN1),
aptodo (Traço_hs, DUJUN1, Traço),
red(+, Traço, OHS).

```

```

calcula-nova-linha ([ ],__,[ ]):-!.
calcula-nova-linha ([NSDUA/NSDUB], [OLJUtA/OLJUtB], [NH/ONH]):-
comb (NSDUA, OLJUtA, Comb),
aptodo (mXm_hs, Comb, DUJUN),
aptodo (Traço_hs, DUJUN, Traço),
red (+, Traço, NH),
calcula-nova-linha (NSDUB, [OLJUtA/OLJUtB], ONH).

```

```

calcula-H ([ ], [ ], [ ]):-!.
calcula-H([LSDU/OLSDU], [LJUt/OLJUt], [H/OH]):-
calcula-linha-dos-elementos-de-H (LSDU,[LJUt/OLJUt],H,
calcula-H(OLSDU, OLJUT, OH).

```

```

calcula-linha-dos-elementos-de-H([ ], [ ], [ ]):-!.
calcula-linha-dos-elementos-de-H(LSDU/OLSDU, LJUt/OLJUt,
H/OH):-
comb (LSDU, LJUt/OLJUt, Comb),
aptodo (mXm_hs, Comb, DUJU),

```

```

aptodo(traco_hs, DUJU, Traço),
red(+, Traço, H),
calcula-linha-dos-elementos-de-H(OLSDU, OLJUt, OH).

```

```

processa-lista-de-SDU( $\phi$ , OLSDU, OLSDU):- ,
processa-lista-de SDU(Sc, [LSDU/OLSDU], [NSDU/OLSDU]):-
processa-SDU(Sc, LSDU, NSDU),
Z is Sc - 1
processa-lista-de-SDU(Z, OLSDU, OLSDU).

```

```

processa-SDU( $\phi$ , OLSDU; OLSDU):- .
processa-SDU(Sc, [LSDU/OLSDU], [NDU/OLSDU]):-
ap(cauda, LSDU, NDU),
X is Sc - 1,
processa-SDU(X, OLSDU, OLSDU).

```

Para o calculo dos termos do vetor de aceleração de gravidade necessita-se calcular cada G_i como expresso na eq 3.7, a partir de uma lista da massa de cada ligamento, do vetor g , da lista de U_{ij} como descrita na eq. 3.8, e da lista de centro de massa de cada ligamento. Exceto para U_{ij} , que é calculado como mostrado anteriormente, todos os outros dados são considerados como entrada ao programa principal pelo usuário.

Assim o calculo de $G(\theta)$, é realizado com,

```

gera-gravidade([ ], __, [ ], [ ], [ ]):-!.
gera-gravidade([M/Outros-Ms], VG, [LRU/OLRU], [R/Outros-RS],-
[ RG/Outros-RGS ]):-

```

```
comb(LRU, [R/Outros-RS ], C1),  
aptodo(mXm-ur, C1, LRUr),  
de(VG, LRUr, C2),  
aptodo(mXm-mr, C2, LgUr),  
Comb([ M/Outros-MS ], LgUr, C3)  
aptodo(eXm, C3, LmgUr),  
(red (+, LmgUr, G),  
[[[G]]] = LmgUr),  
recordz(g, G, RG),  
gera-gravidade(Outros-MS, VG, OLRU, Outros-RS,Outros-RGS)
```

3.4 RESUMO

Neste capítulo, foram mostrados os termos determinantes da equação dinâmica de movimento de um robô industrial, utilizando-se a formulação de Lagrange-Euler.

A explicitação desta equação em seus termos componentes permite a visualização de que estes podem ser implementados de forma independente, utilizando-se notação matricial.

Constata-se também que matrizes resultantes podem ser manipulados como objetos uma vez que não é necessária manipulação em sua estrutura interna.

Esta conclusão permite a utilização da álgebra de combinadores, resultando em uma implementação fácil e clara.

Por fim, a linguagem Prolog, operando predicados desenvolvidos na álgebra anteriormente exposta, mostra a potencialidade da ferramenta.

CAPÍTULO 4 - RESULTADOS COMPUTACIONAIS

4.1 - INTRODUÇÃO

Este capítulo procurar mostrar os resultados obtidos no processamento das equações dinâmicas de um robô industrial segundo a formulação de Lagrange-Euler utilizando-se de modelo dinâmico simbólico programado em Prolog.

4.2 - O ROBÔ

O robô utilizado é composto de variáveis rotacionais nas juntas de nºs 1 e 2 e ligamentos de comprimento l , conforme a fig. 4.1.

Na figura estão também identificados os eixos de orientação dos referidos ligamentos e juntas onde:

- L0 - ligamento 0
- L1 - ligamento 1
- L2 - ligamento 2
- θ_1 - variável da junta 1
- θ_2 - variável da junta 2
- P - ponto da ferramenta

4.3 - PARÂMETROS E MATRIZES

Os parâmetros das juntas estabelecidas para este caso, estão na tabela a seguir:

JUNTA i	PARÂMETROS DAS JUNTAS			
	θ_i	d_i	a_i	α_i
1	θ_1	0	a_1	0
2	θ_2	0	a_2	0

As matrizes de transformação são derivadas utilizando-se os parâmetros acima definidos e a forma geral de A_{i-1}^i onde

LEGENDA

L_0 = Ligamento 0

L_1 = " 1

L_2 = " 2

J_1 = Junta 1

J_2 = Junta 2

P = Ponto de ferramenta

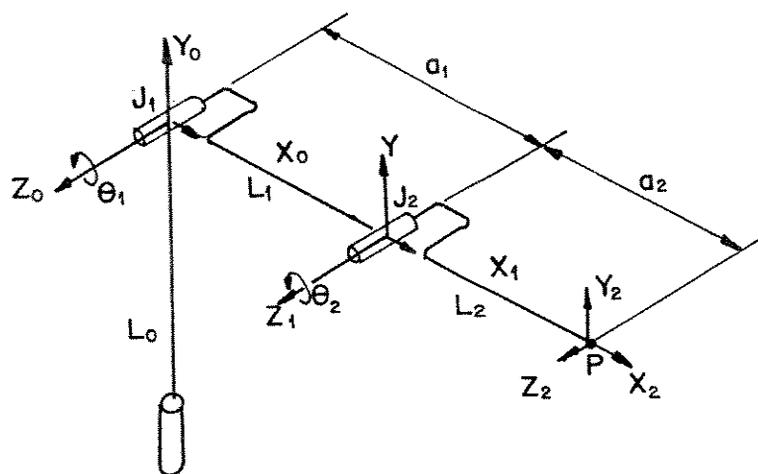


Fig 4.1

$$A_{i-1}^i = \begin{pmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

e tomam a seguinte forma:

$$A_0^1 = \begin{pmatrix} c_1 & s_1 & 0 & a_1 c_1 \\ s_1 & c_1 & 0 & a_1 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_1^2 = \begin{pmatrix} c_2 & s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

As matrizes A_0^1 e A_1^2 são matrizes rotacionais.

4.4 - MODELO SIMBÓLICO

As equações dinâmicas de movimento do robô, expressa na forma matricial compacta, programadas em Prolog, são:

$$\tau = D(\theta) \ddot{\theta} + H(\theta, \dot{\theta}) + G(\theta)$$

onde

$D(\theta)$ = matriz de aceleração inicial simétrica, cujos elementos são

$$D_{ik} = \sum_{j=\max(i,k)}^n \text{Tr}(U_{jk}^J U_{ji}^t) \text{ para } i,k=1,2,\dots,n;$$

$H(\theta, \dot{\theta})$ = vetor de aceleração de Coriolis e força centrífuga
 $= (H_1, H_2, \dots, H_n)^t$

$$H_i = \sum_{k=1}^n \sum_{m=1}^n H_{ikm} \dot{q}_k \dot{q}_m, \quad i = 1, 2, \dots, n$$

$$H_{ikm} = \sum_{j=\text{MAX}(i,k,m)}^n {}^T_r(U_{jkm} J_j U_{ji}^t), \quad i, k, m = 1, 2, \dots, n;$$

$G(\theta)$ = vetor de força gravitacional = $(G_1, G_2, \dots, G_n)^t$

$$G_i = \sum_{j=i}^n (-m_j g U_{ji} \bar{r}_j) \quad \text{para } i=1, 2, \dots, n$$

4.5 - RESULTADOS

Os resultados obtidos, na forma simbólica referem-se aos termos anteriores considerando-se os seguintes dados do robô:

- . m_1 = massa do ligamento 1
- . m_2 = massa do ligamento 2
- . ℓ_1 = comprimento do ligamento 1
- . ℓ_2 = comprimento do ligamento 2
- . $X_0 Y_0 L_0$ - base
- . $X_1 Y_1 Z_1$ - coordenadas de referência 1
- . $X_2 Y_2 Z_2$ - coordenadas de referência 2
- . matriz de inércia J_1

$$J_1 = \begin{pmatrix} \frac{1}{3} m_1 \ell_1^2 & 0 & 0 & -\frac{1}{2} m_1 \ell_1 \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ -\frac{1}{2} m_1 \ell_1 & 0 & 0 & m_1 \end{pmatrix}$$

onde $J_{111} = \frac{1}{3} m_1 \ell_1^2$, $J_{114} = -\frac{1}{2} m_1 \ell_1$

$J_{141} = -\frac{1}{2} m_1 \ell_1$ e $J_{144} = m_1$

ficando a entrada da matriz de inércia no programa, para o 1º grau de liberdade:

$$1^{\text{a}} \text{ linha da matriz: } J_{111}, 0, 0, J_{114}$$

$$2^{\text{a}} \text{ linha da matriz: } 0, 0, 0, 0$$

$$3^{\text{a}} \text{ linha da matriz: } 0, 0, 0, 0$$

$$4^{\text{a}} \text{ linha da matriz: } J_{141}, 0, 0, J_{144}$$

. matriz de inércia J_2

$$J_2 = \begin{pmatrix} \frac{1}{3} m_2 l_2^2 & 0 & 0 & -\frac{1}{2} m_2 l_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} m_2 l_2 & 0 & 0 & m_2 \end{pmatrix}$$

onde $J_{211} = \frac{1}{3} m_2 l_2^2$, $J_{214} = -\frac{1}{2} m_2 l_2$, $J_{241} = -\frac{1}{2} m_2 l_2$, $J_{244} = m_2$
da mesma maneira, para o 2º grau de liberdade:

$$1^{\text{a}} \text{ linha da matriz: } J_{211}, 0, 0, J_{214}$$

$$2^{\text{a}} \text{ linha da matriz: } 0, 0, 0, 0$$

$$3^{\text{a}} \text{ linha da matriz: } 0, 0, 0, 0$$

$$4^{\text{a}} \text{ linha da matriz: } J_{241}, 0, 0, J_{244}$$

ou

$$J_1 = \begin{pmatrix} J_{111} & 0 & 0 & J_{114} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ J_{141} & 0 & 0 & J_{144} \end{pmatrix}$$

$$J_1 = [[J_{111}, 0, 0, J_{114}], [0, 0, 0, 0], [0, 0, 0, 0], [J_{141}, 0, 0, 0, J_{144}]] \quad \text{eq. 4.1}$$

e

$$J_2 = \begin{bmatrix} J_{211} & 0 & 0 & J_{214} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ J_{241} & 0 & 0 & J_{244} \end{bmatrix}$$

$$J_2 = [[J_{211}, 0, 0, J_{114}], [0, 0, 0, 0], [0, 0, 0, 0], [J_{41}, 0, 0, J_{44}]] \text{ eq. 4.2}$$

$$J = [[[J_{111}, 0, 0, J_{114}], [0, 0, 0, 0], [0, 0, 0, 0], [J_{141}, 0, 0, 0, J_{144}]], \\ [[J_{211}, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [J_{241}, 0, 0, J_{244}]]] \text{ eq. 4.3}$$

Na eq. 4.2 estão os parâmetros de entrada das matrizes de inércia no programa de modelagem dinâmica colocadas em sua forma final.

. Centro de massa dos ligamento 1 e 2 onde

$$\bar{r}_i = [\bar{x}_i, \bar{y}_i, \bar{z}_i, 1]^t$$

obtem-se:

$$\bar{r}_1 = [[r1x], [0], [0], [1]] \text{ eq.4.4}$$

$$\bar{r}_2 = [[r2x], [0], [0], [1]] \text{ eq.4.5}$$

$$\bar{r} = [[[r1x], [0], [0], [1]], [[r2x], [0], [0], [1]]] \text{ eq.4.6}$$

onde a eq. 4.6 é a matriz de entrada do centro de massa dos ligamentos 1 e 2 no programa de modelagem dinâmica, na sua forma final.

Obtem-se a partir das equações 3.1 a 3.11, programadas no modelo dinâmico, as saídas seguintes:

$D(\theta)$ = matriz de aceleração normal

$G(\theta)$ = vetor de força gravitacional

$H(\theta, \dot{\theta})$ = vetor de força centrífuga,

compactamente:

$$D_{11} = T_r(U_{11} J_{11} U_{11}^t) + T_r(U_{21} J_2 U_{21}^t) \text{ eq. 4.6}$$

$$D_{12} = T_r(U_{22} J_2 U_{21}^t) = D_{21} \text{ eq. 4.7}$$

$$D_{22} = \text{Tr}(U_{22} J_2 U_{22}^t) \quad \text{eq. 4.8}$$

$$G_1 = m_1 g U_{11} \bar{r}_1 + m_2 g U_{21} \bar{r}_2 \quad \text{eq. 4.9}$$

$$G_2 = m_2 g U_{22} \bar{r}_2 \quad \text{eq. 4.10}$$

$$H_1 = H_{111} + H_{112} + H_{122} = U_{111} J_1 U_{11}^t + U_{211} J_2 U_{21}^t + \\ U_{212} J_2 U_{21}^t + U_{222} J_2 U_{21}^t \quad \text{eq. 4.11}$$

$$H_2 = H_{211} + H_{212} + H_{222} = U_{211} J_2 U_{22}^t + \\ U_{212} J_2 U_{22}^t + U_{222} J_2 U_{22}^t \quad \text{eq. 4.12}$$

onde:

$$U_{11} = \begin{bmatrix} [-\text{sen}(\text{teta1}), -\text{cos}(\text{teta1}), 0, -\ell * \text{sen}(\text{teta1})] \\ [\text{cos}(\text{teta1}), -\text{sen}(\text{teta1}), 0, \ell * \text{cos}(\text{teta1})] \\ [0, 0, 0, 0] \\ [0, 0, 0, 0] \end{bmatrix}$$

$$U_{21} = \begin{bmatrix} [-\text{sen}(\text{teta1}+\text{teta2}), -\text{cos}(\text{teta1}+\text{teta2}), 0, -\ell * \text{sen}(\text{teta1}) * \\ (1+\text{cos}(\text{teta2})+\text{cos}(\text{teta1}) * \text{sen}(\text{teta2}))] \\ [\text{cos}(\text{teta1}+\text{teta2}), -\text{sen}(\text{teta1}+\text{teta2}), 0, \ell * (\text{cos}(\text{teta1}+\text{teta2})+\text{cos}(\text{teta2}))] \\ [0, 0, 0, 0] \\ [0, 0, 0, 0] \end{bmatrix}$$

$$U_{22} = \begin{bmatrix} [-\text{sen}(\text{teta1}+\text{teta2}), -\text{cos}(\text{teta2}+\text{teta1}), 0, -\ell * \text{sen}(\text{teta2}+\text{teta1})] \\ [\text{cos}(\text{teta1}+\text{teta2}), -\text{sen}(\text{teta1}+\text{teta2}), 0, \ell * \text{cos}(\text{teta1}+\text{teta2})] \\ [0, 0, 0, 0] \\ [0, 0, 0, 0] \end{bmatrix}$$

e

U_{111} = derivada de U_{11} em relação à variável θ_1

U_{211} = derivada de U_{21} em relação à variável θ_1

U_{212} = derivada de U_{21} em relação à variável θ_2

U_{222} = derivada de U_{22} em relação à variável θ_2

4.6 - CONCLUSÃO

Os resultados obtidos mostram a reconhecida complexidade do modelo dinâmico completo de um robô industrial.

Desta forma pode-se avaliar a razão pela qual os modelos são simplificados a critério do pesquisador segundo a orientação dada à pesquisa.

CAPÍTULO 5 - CONTROLE HIERÁRQUICO
DE ROBÔS

CAPÍTULO 5 - CONTROLE HIERÁRQUICO DE ROBÔS

5.1 - INTRODUÇÃO

Básicamente um robô constitui-se de dois grandes sistemas: o sistema mecânico e o de controle.

As primeiras gerações de robôs apresentavam-os como um sistema mecânico, dotado de manipulador manual e sensores limitadores ou mecanismos ajustáveis.

Estes sensores posicionavam o conjunto manipulador de maneira pré-programada[12], inflexível.

A evolução do conjunto mecânico e de controle, na direção de um sistema integrado e "inteligente" tem como premissa o aperfeiçoamento do controle, e dos sistemas de sensoramento que permitam informações mais precisas, em maior número, e com maior confiabilidade.

A visão integrada[27] do conjunto mecânico e de controle de um sistema robótico, conforme fig. 5.1., mostra a necessária consideração de vários aspectos quais sejam :

- . processamento de informações (sinais) fornecidos pelos sensores;
- . armazenamento e transferência dessas informações;
- . cálculos e conversões de pontos de trajetória e seu planejamento;
- . utilização de comandos e técnicas de aprendizagem;
- . utilização de modelos dinâmicos inversos;
- . controle direto das juntas.

A realização das etapas descritas está ligada ao desenvolvimento de um sistema inteligente dotado de um controle flexível e preciso durante o movimento do sistema robótico.

No que concerne ao sistema de sensoramento há que contar-se com sensores de alta capacidade (visuais, táteis), de média capacidade (como os de proximidade) e primitivos (tecômetros, potenciômetros).

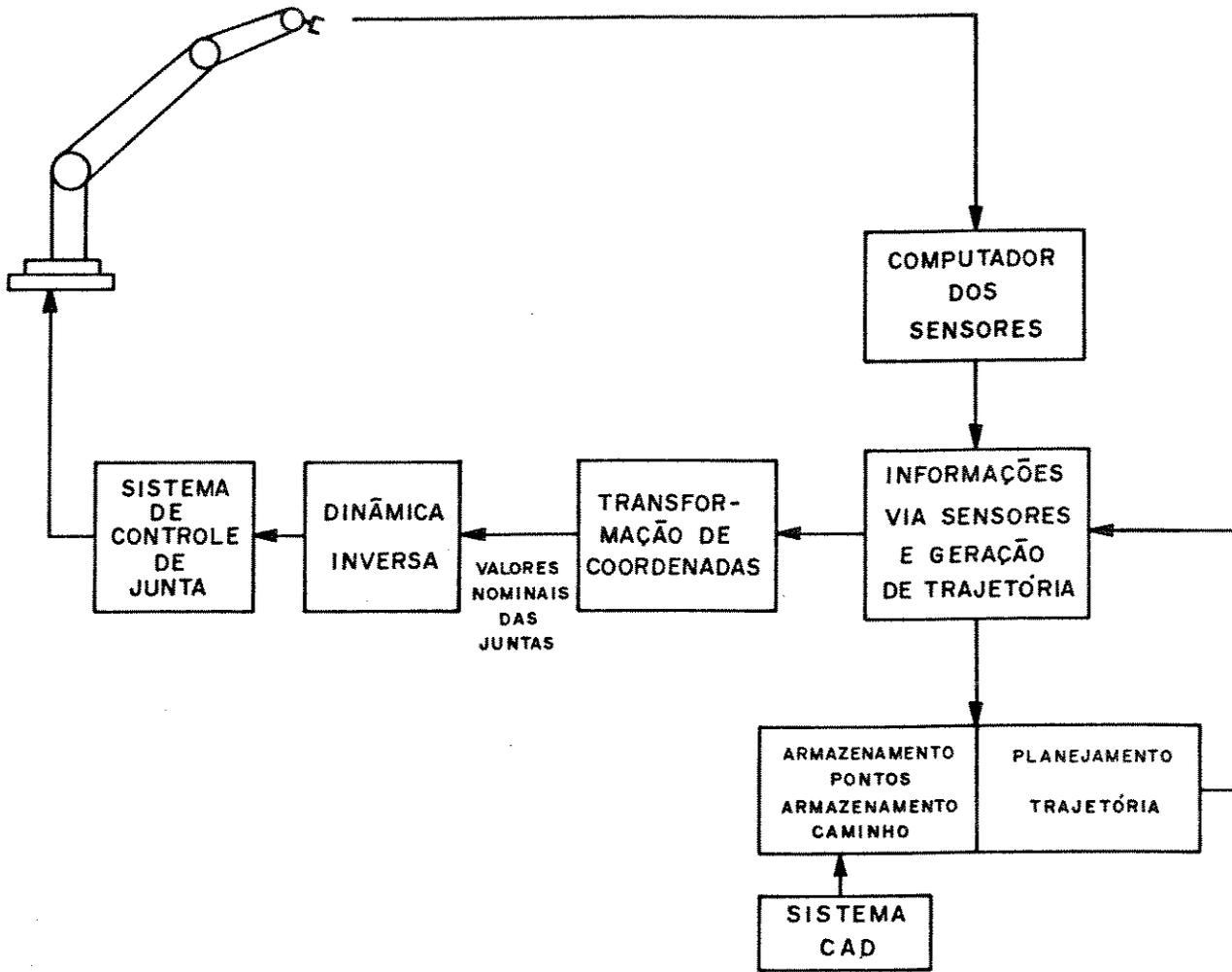


Fig 5.1

Este complexo, tem porém apenas a capacidade de melhor "sentir" os efeitos ambientais, sem que possa discernir acerca de les.

Deste modo, é necessário acrescentar-se um sistema específico de comparação e aprendizagem e que ajude em tomada de decisões.

Este sistema pode ser definido como fazendo parte de um sistema de reconhecimento inteligente.

Em suma, um robô pertencente à última geração, pode ser considerado como subdivido nos seguintes subsistemas[13], [9].

- . sistema mecânico
- . sistema de controle
- . sistema sensorial
- . sistema de reconhecimento

A pesquisa [13],[14],[15],[21],[26]de uma estrutura de atuação para os subsistemas anteriormente descritos, em especial para os de controle, sensoramento e reconhecimento, está voltada para uma forma hierárquica, composta de vários níveis, onde cada nível deverá ter suas tarefas específicas e critérios de autonomia e correlação com os demais.

Desta forma, pode-se considerar como objetivo deste capítulo a formulação de um sistema de controle, estruturado na forma hierárquica, e que incorpore recentes metodologias teóricas desenvolvidas, baseadas em rigorosa modelagem matemática e subsequente análise dos processos físicos associados.

Este sistema de controle será tanto mais eficiente quanto os seus resultados apresentados aproximarem-se das características de desempenho do controle realizado diretamente pelo ser humano.

5.2 - NÍVEIS HIERÁRQUICOS DE SISTEMA DE CONTROLE

De um modo geral, vamos considerar o sistema, fig. 5.2., composto de três macro-níveis[14]que podem ser denominados:

- . Nível 1 - Organizador (ORCOH)
- . Nível 2 - Coordenador (CORCOH)

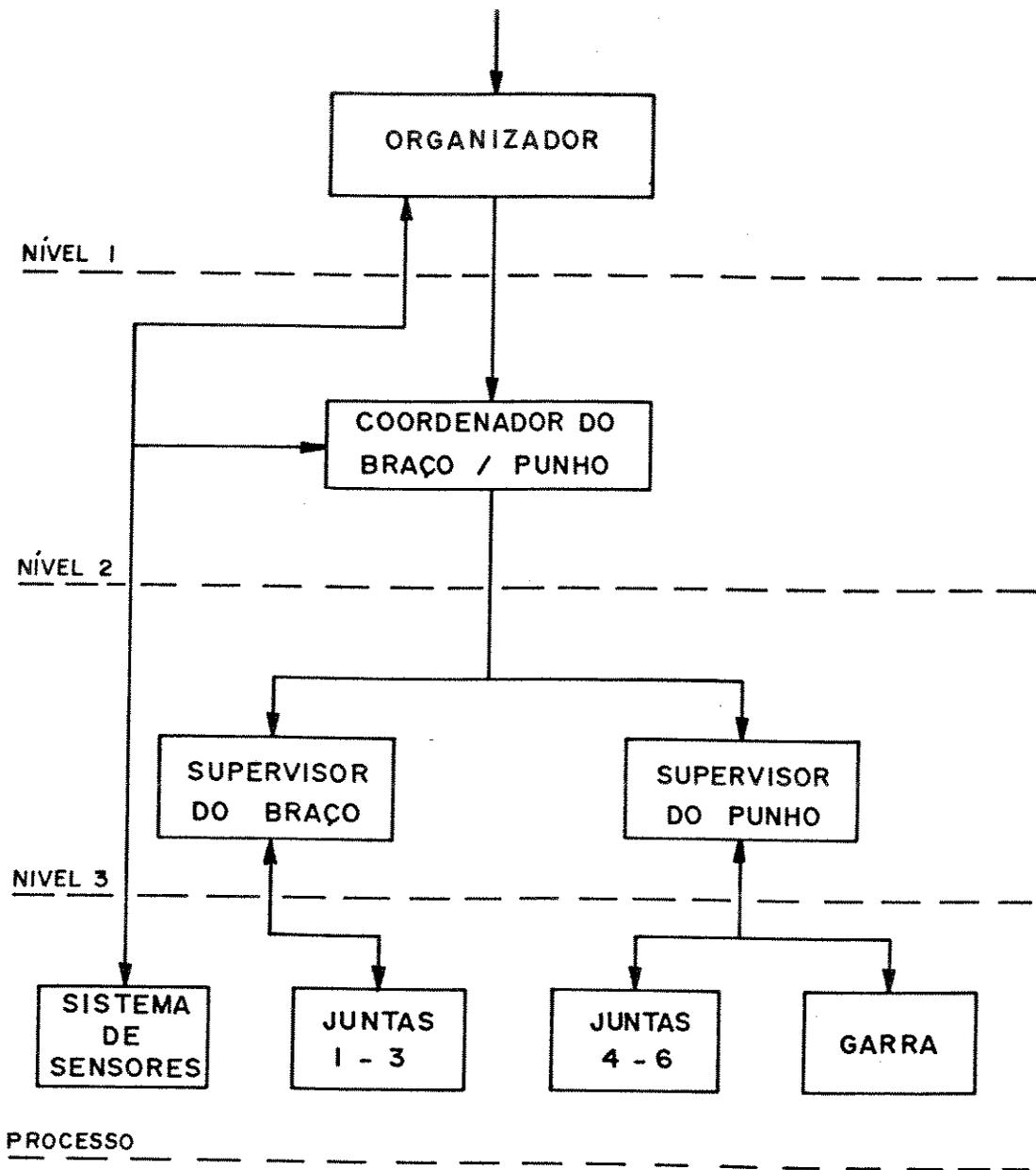


Fig 5.2

. Nível 3 - Supervisor (SURCOH)

Neste níveis algumas premissas fundamentais precisam ser consideradas [16]:

- . os níveis de 3 para 1 apresentam uma redução de precisão com aumento de inteligência, culminando com o nível organizador;
- . a informação processada de um nível particular não é diretamente disponível aos outros níveis de estrutura;
- . as informações fluem em dois sentidos: do nível mais alto (nível 1) para o nível mais baixo (nível 3) e vice versa;
- . um nível emite comandos apenas para o nível imediatamente inferior;
- . a realimentação das informações baseadas em um determinado nível é fechada no nível imediatamente superior;
- . o nível mais baixo (supervisor) consiste de vários controladores projetados para efetivar o controle usando "hardware" existente;
- . o próximo nível (coordenador) utiliza ao máximo as características dos micro-computadores para de posse das informações geradas pelo nível organizador e a realimentação fornecida pelo nível supervisor, posicionar o sistema na trajetória desejada;
- . o nível mais alto (organizador) tem como meta o controle do sistema como um todo, recebendo informações do sensoramente e utilizando programas, dados e sistemas de reconhecimento e tomada de decisão.

5.2.1 - O NÍVEL ORGANIZADOR (ORCOH)

O nível organizador de controle hierarquico (ORCOH) é o de mais alta hierarquia e maior "inteligencia" na estrutura proposta.

Suas atividades compreendem:

- . reconhecimento de situações complexas no ambiente;
- . discriminação das tarefas e sub-tarefas a serem realizadas;
- . envio de comando sequencial das tarefas ao nível coor
denador;
- . organização e consulta de biblioteca de dados e padrões;
- . coordenação das informações recebidas e enviadas do sistema de sensores;
- . envio de comando(s) ao nível coordenador baseado nas informações recebidas dos sensores.

No reconhecimento das situações complexas no ambiente, o nível organizador, atua como uma combinação homem-máquina.

A alteração dos parâmetros de um determinado ambiente e até mesmo do ambiente [17] leva à alteração dos índices de desempenho a serem utilizados.

Esta mudança deve ser reconhecida pelo "reconhecedor" de padrão, embutido neste nível, fig. 5.3.

A informação para este reconhecimento, dada pelos parâmetros do ambiente, é obtido do conjunto sensorial, em especial do sistema de visão.

O ambiente, de modo geral caracterizado por um comportamento aleatório, implica na necessidade de existir neste nível, um autômato de aprendizagem [18], que selecionará uma ação (tarefa) entre um conjunto finito possível e atualizará sua estratégia.

A tarefa será realizada como uma sequência de ações elementares (como movimentos e leituras de câmera).

Torna-se necessário para que haja eficiência a preparação "a priori" de uma sequência de ações calculadas para produzir um efeito desejado.

Este processo conhecido como "problem solving" e "planning" [17] é realizado antes que a ação do robô seja iniciada.

Evidentemente, para que haja planejamento, o sistema robótico deve conhecer (estimar) o efeito de suas ações.

O conhecimento dos efeitos das ações permite estabelecer a correlação das ações com o ambiente.

O processo "problem-solving" utiliza informação armaze

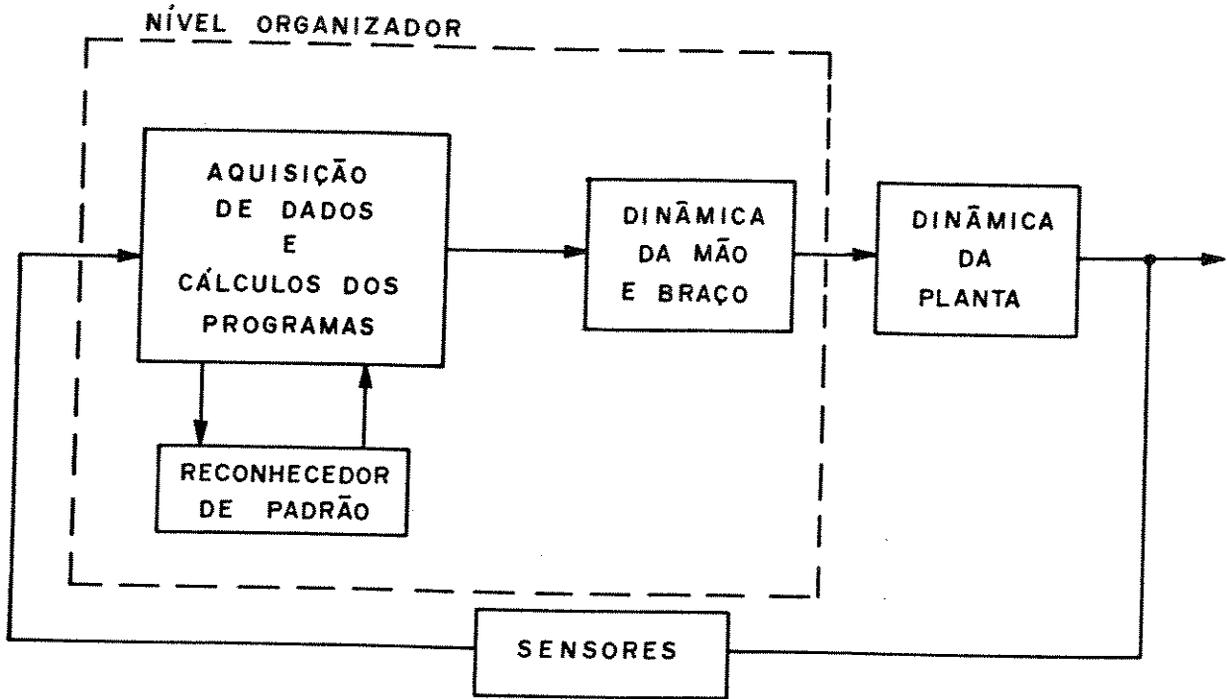


Fig 5.3

da na forma de um modelo para calcular qual sequência de ações conduzirá o processo controlado ao estado desejado.

Como o ambiente muda, pelas próprias ações do robô, ou por outras razões o modelo deve ser atualizado, sendo então retidas essas alterações.

Novas informações sobre o ambiente devem ser adicionadas, ao modelo, atualizando-o em um processo de aprendizagem[21]

Em alguns casos, o ambiente pode ser expresso em termos de um pequeno número de situações (estados).

Nestes casos o método de programação dinâmica é utilizado para determinar a sequência de ações (política de controle ótimo) necessárias para atingir o estado desejado[19].

Presentemente, não existem algoritmos disponíveis para decisões ótimas[17] aplicáveis à grande número de tarefas, em ambientes complexos.

Técnicas heurísticas para "problem solving" tem sido propostas [20] recentemente, direcionadas com as pesquisas em inteligência artificial.

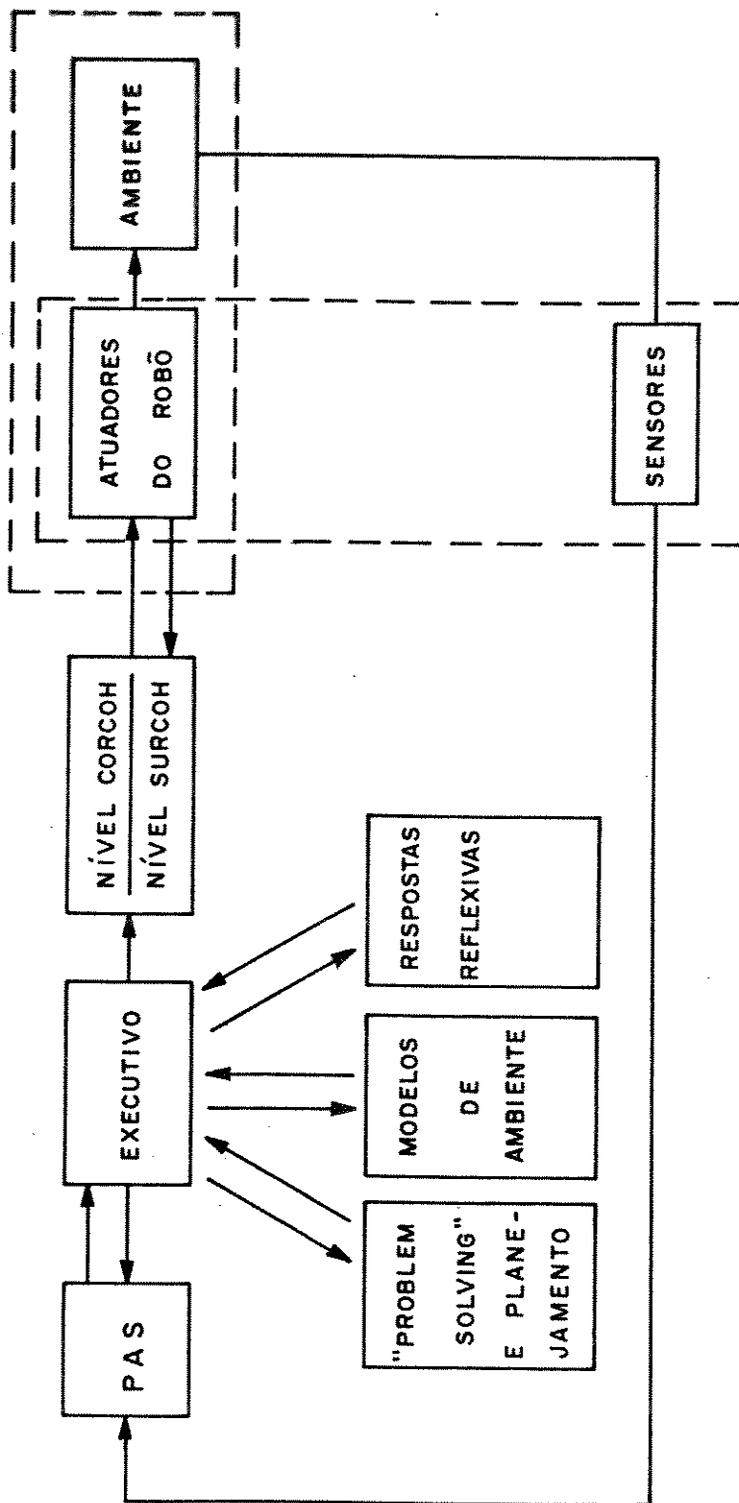
De forma a sintetizar o anteriormente exposto, pode mos considerar o nível ORCOH, como sendo aquele que realiza a execução das tarefas combinando:

- . decisões de alta inteligência, geralmente executadas por controlador humano.
- . atividades que requerem inteligência relativamente menor, usualmente executadas pelo computador atuando como controlador.

A Figura 5.4, esclarece graficamente o exposto, determinando os componentes modulares responsáveis pelas ações.

O bloco de respostas reflexivas é similar à um controlador convencional, o qual executa tarefas correspondentes a soluções pré-estabelecidas para situações simples e previsíveis do ambiente.

Todas as informações, derivadas dos blocos correspondentes são procuradas em uma placa, aqui denominada de "executiva", a qual, por sua vez interage com a placa PAS, o processador de acoplamento de sensores.



PAS = PROCESSADOR DE ACOPLAMENTO DOS SENSORES

Fig 5.4

Consideramos, sob o ponto de vista da implementação como fornecedores de informação à este nível, e pertencentes ao sistema inteligente de sensores (SSI), os seguintes componentes:

- . sensores de visão
- . sensores de tato
- . sensores de força/movimento
- . sensores de distancia e força/pega

A comunicação feita com o sistema de sensores inteligentes (SSI) é feita através da placa do processador de acoplamento de sensores (PAS), fig. 5.5.

As instruções ao SSI são consideradas de duas espécies:

- . comandos
- . solicitações

Os comandos iniciam ações no sistema sensorial, sendo os dados obtidos através do PAS.

As comunicações são realizadas paralelamente com as operações normais nos processadores PNC1 e PNC2 (nível CORCOH), fig. 5.5.

A comunicação entre PAS e os processador(es) dos níveis inferiores, pode ser implementada, por exemplo, via memória "dual port" [21].

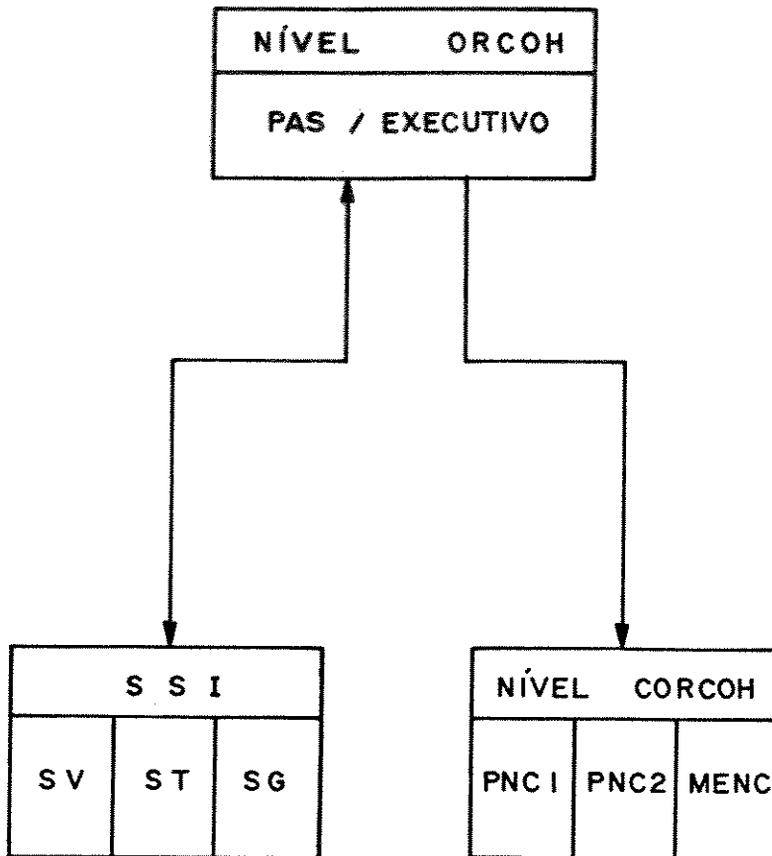
As informações fornecidas pelo sistema sensorial através do nível ORCOH, atuam no sistema de controle da trajetória no nível CORCOH.

Estas informações definem hierarquicamente as decisões a serem tomadas, levando o sistema à trajetória desejada, fig. 5.6.

É realizado também no nível ORCOH o cálculo das transformações e a comunicação entre PAS e PNC1, através de um "software" modular.

De uma forma básica, há que se considerar a indicação dos pontos finais da trajetória a ser seguida pelo robô, tanto na fase de aprendizagem ou preparação quanto durante a operação em modo totalmente automático.

Durante a fase de aprendizagem (preparação) a informação geométrica referente às posições atual e final é calculada e fornecidas ao processador PNC2 no nível CORCOH.



PAS = PROCESSADOR DE ACOPLAMENTO DOS SENSORES
 SSI = SISTEMAS DE SENSORES INTELIGENTES
 SV = SENSOR DE VISÃO
 ST = SENSOR DE TATO
 SG = SENSOR DE GARRA
 PNC1=PROCESSADOR DO NÍVEL COORDENADOR 1
 PNC2 = PROCESSADOR DO NÍVEL COORDENADOR 2
 MENC = MEMÓRIA NÍVEL COORDENADOR

Fig 5.5

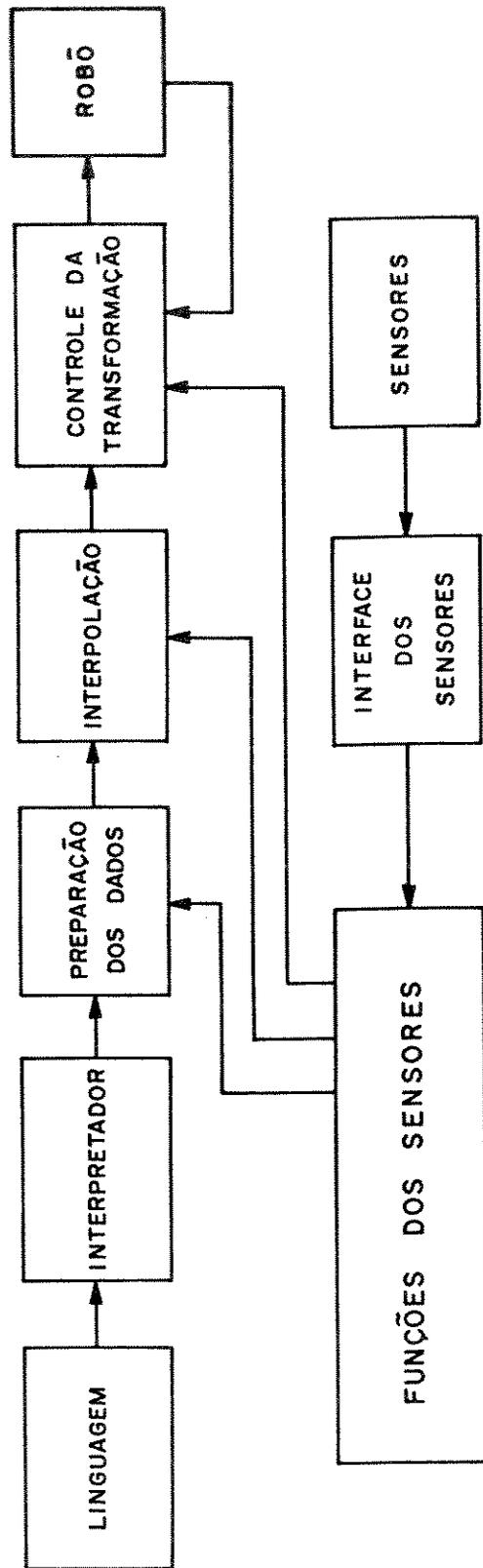


Fig 5.6

Caso as informações fornecidas pelos sensores sejam disponíveis no tempo cíclico da interpolação para o cálculo da trajetória (interpolador em PNC2 - nível CORCOH), há necessidade de serem realizadas transformações de coordenadas.

As transformações são feitas em até 3 etapas:

- . Estando o sistema de câmera montado em cima da garra, as coordenadas de câmera são transformadas para as coordenadas do plano de trabalho definido na fase de calibração ;
- . As novas coordenadas são então transformadas para as do sistema da garra;
- . A transformação final deste sistema para o sistema base do robô (sistema mundo) não é sempre necessária.

Os pontos de coordenadas transformados são então armazenados na memória do tipo "dual port", como sugerida anteriormente.

Todas estas informações e mais a de posição, orientação, velocidade, aceleração e tipo de interpolação, recebidas através do sistema sensorial são enviadas para o programa de controle do robô, residente nos processadores PNC1 e PNC2, nível CORCOH.

Neste nível está residente também a linguagem de programação de integração que permite a realização de tarefas complexas, o sistema de programação aplicativa (SPA).

Esta permite ao usuário trabalhar com vários tipos de dados, como por exemplo:

- . pontos em coordenadas de junta;
- . pontos em coordenadas base, posição e orientação;
- . real
- . inteiro
- . caracter
- . binário
- . "array"

5.2.2 - O NÍVEL COORDENADOR (CORCOH)

O nível coordenador tem como atividade fundamental a consecução da trajetória especificada, fig. 5.7.

Para isto recebe do nível superior a ele (ORCOH) informações, dados, comandos e solicitações em uma forma operacionalizável, tal como:

- . coordenadas de pontos da trajetória via sensores;
- . determinação do tipo de interpolação a ser seguida pelo interpolador;
- . sinais de controle dos eixos quando de sua realização via sensores;
- . para utilização no programa de controle do robô, da dos de:
 - posição
 - orientação
 - velocidade
 - aceleração;
- . instruções para sincronização com objeto externo móvel, como p.ex, correia transportadora;
- . quando da etapa de aprendizagem, informações geométricas referentes às posições atual e final.

Para a realização deste nível, um conjunto de itens são nele residentes, realizados por um grupo de processadores e memórias, tais como:

- . sistema operacional (S.O.)
- . sistema de teste
- . sistema de diagnóstico e de status
- . editor
- . interpolador
- . unidade de ponto flutuante e controle de "timing"
- . controlador de interrupção
- . memória(s)

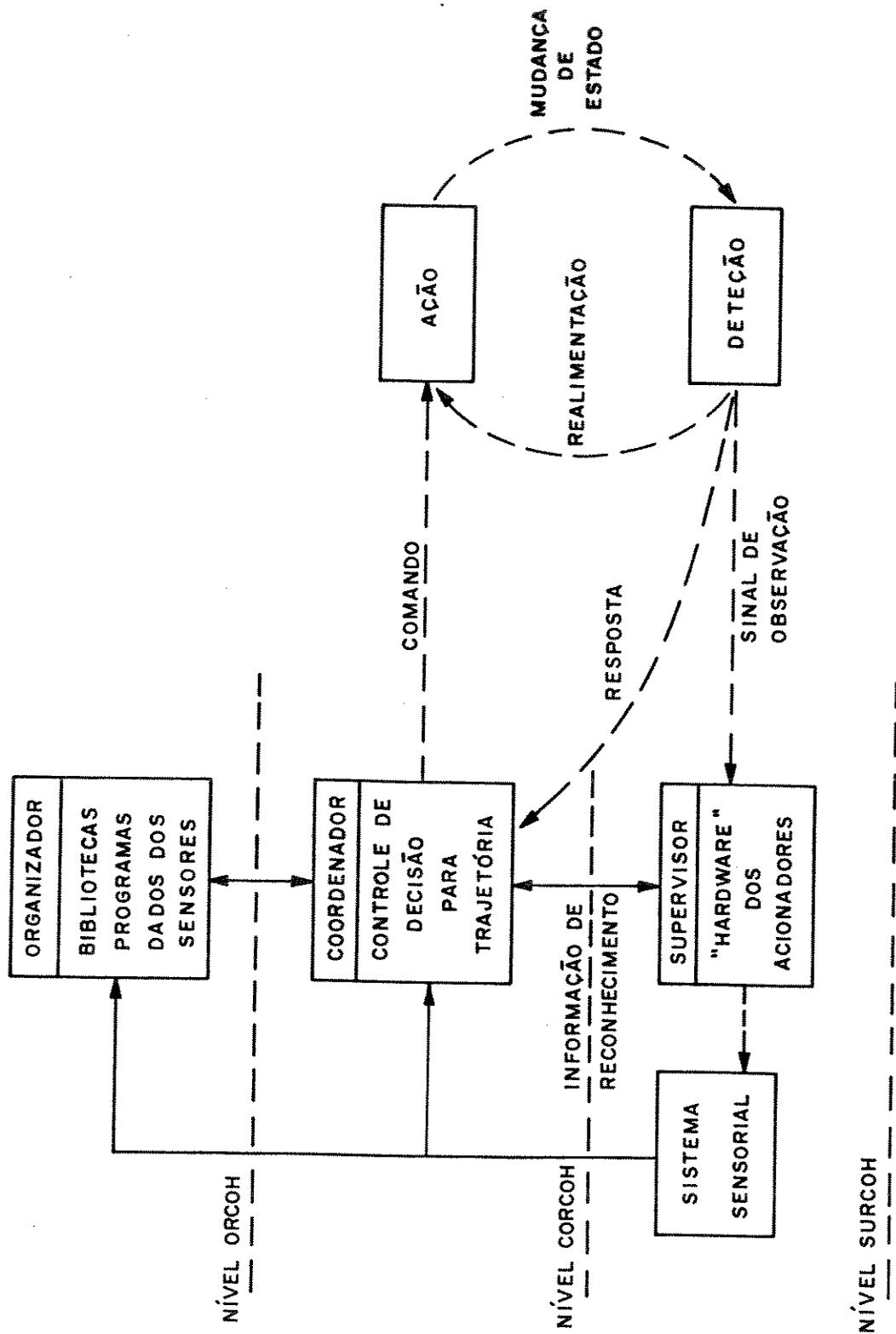


Fig 5.7

. linguagem do programa de controle do robô.

O nível coordenador é também responsável pela movimentação coordenada dos seis eixos do robô [26].

As matrizes de movimento do espaço do robô (corpo, torso e braço), estão no domínio XYZ e as do espaço local (punho, garra no domínio xyz (figs 5.7 e 5.8).

As matrizes nos domínio de XYZ e xyz são calculadas neste nível, refletindo a posição desejada e orientação da garra.

A informação é então reestruturada por algoritmo de transformação em seis componentes, definindo características de cada eixo individualmente ($R, z, \theta, \alpha, \beta$ e γ), fig 5.9.

A seguir as informações são enviadas ao nível próximo na hierarquia inteligente.

Estão também residentes neste nível as equações diferenciais que descrevem a trajetória nominal.

Os dados de posição, orientação, velocidade e aceleração, fig. 5.7, são realimentados, a partir do conjunto manipulador/garra, sendo os mesmos enviados ao nível coordenador e operacionalizados no nível supervisor.

O nível coordenador, de posse das informações referentes ao ambiente (estados) dadas pelo nível organizador, e da realimentação dos dados, compatibiliza-os, informando ao nível supervisor os valores aceitos, fig 5.6.

Com relação aos aspectos da implementação seguem as seguintes considerações: neste nível, conforme fig. 5.5 estão presentes dois processadores, denominados PNCl e PNC2, executando várias tarefas, sendo para o PNCl:

- . controlar o sistema operacional.
- . coordenar neste nível a comunicação, através de "soft" especial com o processador de acoplamento de sensores, residente no nível ORCOH.
- . no sistema operacional executar:
 - . teste
 - . diagnóstico
 - . modelos de status
 - . editor

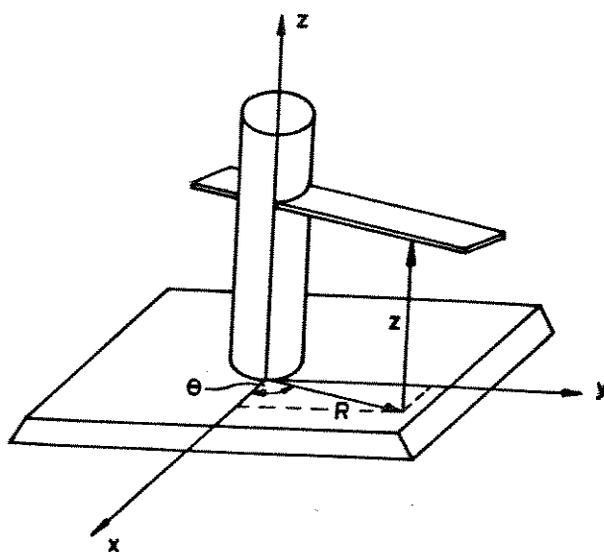


Fig 5.8

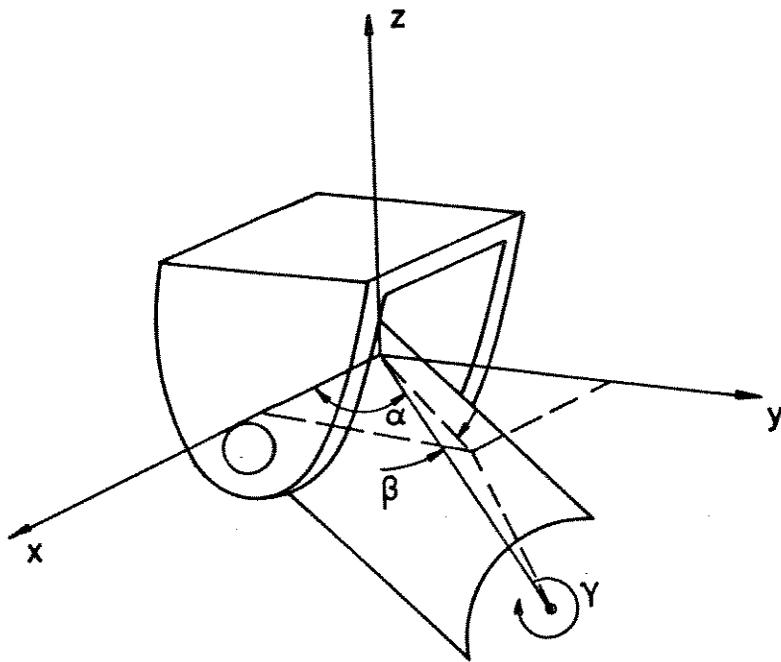


Fig 5.9

Para o PNC2, temos as tarefas:

- . calcular as interpolações necessárias
- . calcular as transformações de coordenadas XYZ e xyz e
- . controlar os eixos

Além disso dispõe de uma memória MENC (com outra opcional), fig. 5.5, que contém o sistema operacional e programa do usuário, unidade de ponto flutuante, unidade controladora de timing, controlador de interrupção.

A comunicação entre PNC1 e PNC2 é realizada, por exemplo, com memória do tipo "dual port".

Através do microprocessador PNC2, o nível CORCOH gera a posição/tempo solicitada para cada eixo [26].

O parâmetro de posição representa a posição final desejada, enquanto que o parâmetro de tempo representa o desejado tempo de convergência.

Estes dois parâmetros permitem que haja coordenação de todos os eixos simultaneamente.

Um algoritmo recebe estes parâmetros de forma a permitir que contornos complexos sejam realizados

O vetor de parâmetros posição/tempo é enviado ao nível SURCOH para cada controlador dos eixos com realimentação (CER) atualizando o movimento.

Cada CER aceita o vetor de parâmetros de posição /tempo a partir do nível CORCOH e atualiza o movimento.

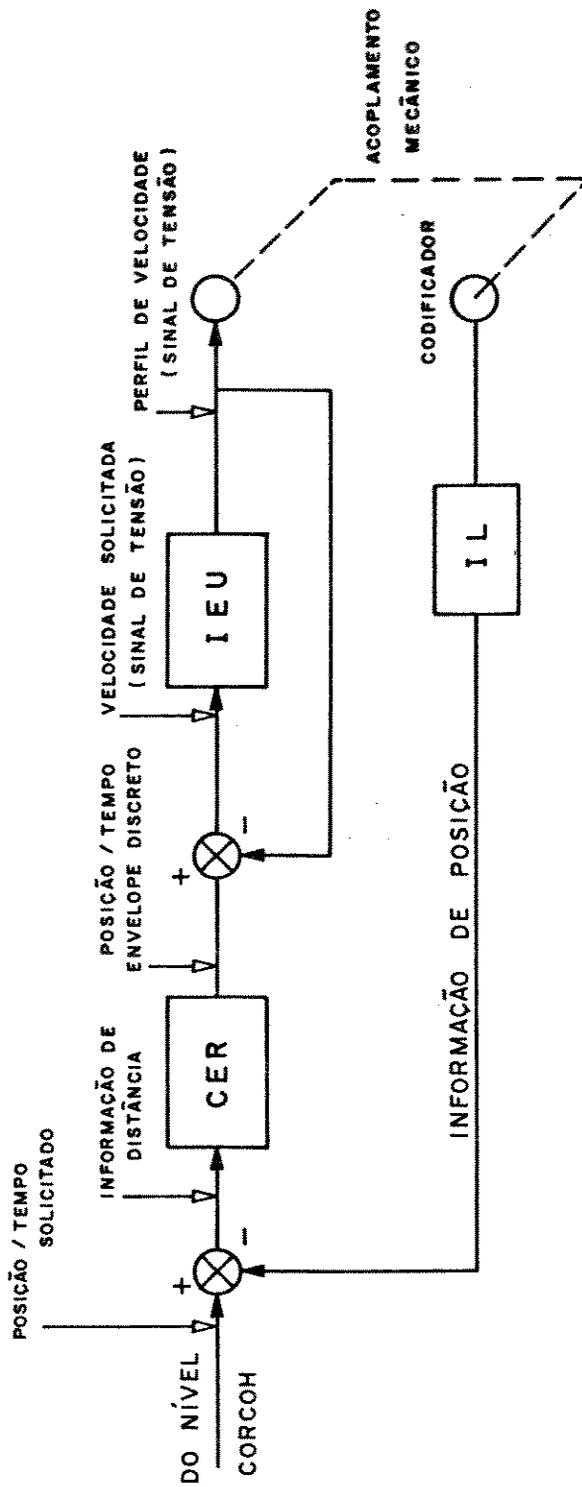
Na fig. 5.10 está esclarecido o fluxo de dados entre os componentes do sistema, a partir das informações do nível CORCOH.

O robô é modelado por dois vetores: um vetor de posição $P_R(R, Z, \theta)$ e um vetor de orientação $O_L(\alpha, \beta, \gamma)$.

O vetor de posição representa a locação do punho (a origem do espaço local) no espaço do robô.

O vetor de orientação descreve a condição de mão no espaço local.

O vetor de posição P_R está no espaço tridimensional XYZ e pode ser escrito como um vetor representando a posição individual dos eixos envolvidos, como R, θ e Z, $P_R = [X, Y, Z] = [R, \theta, Z]$ sendo o seu comprimento



CER = CONTROLADOR DOS EIXOS COM REALIMENTAÇÃO
 IEU = INTERFACE DE EIXO UNITÁRIO
 IL = INTERFACE LÓGICA

Fig 5.10

$$|P_R| = (R^2 + Z^2)^{1/2} = (X^2 + Y^2 + Z^2)^{1/2}$$

e as relações, fig. 5.8 e 5.9,

$$X = R \cos \theta$$

$$Y = R \sin \theta$$

$$Z = Z$$

com transformadas inversas

$$\theta = \text{tg}^{-1} (Y/X)$$

$$R = (X^2 + Y^2)^{1/2}$$

$$Z = Z$$

para o vetor O_L similarmente:

$$O_L = [x, y, z] = [\alpha, \beta, \gamma]$$

o comprimento de O_L não se altera, uma vez que o comprimento do punho, l , é constante. Assim

$$x = l \cos \beta \cos \alpha$$

$$y = l \cos \beta \sin \alpha$$

$$z = l \sin \beta$$

e as transformadas inversas

$$\alpha = \text{tg}^{-1} (y/x)$$

$$\beta = \text{sen}^{-1} (z/l) = \text{tg}^{-1} (z/(x^2 + y^2)^{1/2})$$

$$\gamma = \gamma$$

onde α representa a rotação do vetor de orientação em torno da sua linha de centro, β é sua própria transformada, independente da orientação de O_L .

Se a posição do punho no espaço do robô é (X_1, Y_1, Z_1) correspondendo a (R_1, θ_1, Z_1) e a nova posição desejada (X_2, Y_2, Z_2) correspondendo a (R_2, θ_2, Z_2) é solicitada a diferença para cada eixo será:

$$R = R_2 - R_1$$

$$Z = Z_2 - Z_1$$

$$\Delta\theta = \theta_2 - \theta_1$$

Desde que a velocidade máxima para cada eixo seja conhecida (V_{RMAX} , V_{ZMAX} e $V_{\theta MAX}$), o tempo máximo possível para eixo

individual alcançar seu destino é, abandonando acelerações como a aproximações de 1.^a ordem,

$$t_{RMAX} = \Delta R / V_{RMAX}$$

$$t_{ZMAX} = \Delta Z / V_{ZMAX}$$

$$t_{MAX} = \Delta \theta / V_{\theta MAX}$$

onde

$$t_{MAX} = \text{MAX} (t_{RMAX}, t_{ZMAX}, t_{MAX})$$

Se cada CER é instruído a alcançar seu destino respectivo em tempo t_{MAX} , haverá convergência simultânea.

Desta maneira, ficará computado o parâmetro de tempo de convergência (PTC) e o parâmetro final posição/ ângulo (PFP) para cada eixo.

Enquanto os CER estarão supervisionando o movimento de seus eixos respectivos, no nível CORCOH através de PNC2 estão sendo computados os próximos PTC e PFP solicitados.

5.2.3 - O NÍVEL SUPERVISOR (SURCOH)

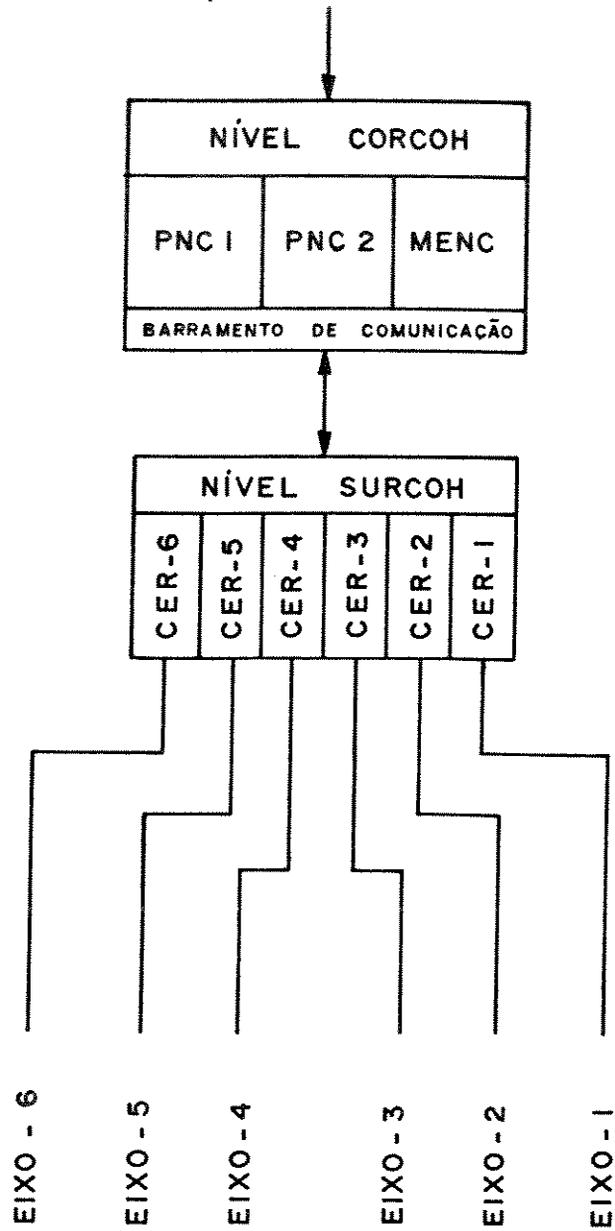
O nível supervisor, fig. 5.11, é o nível hierarquicamente mais inferior.

A definição própria de estrutura hierárquica mostra que este nível é aquele que realiza sua tarefa de forma mais precisa isto é direta, usando de menor quantidade de "inteligência".

Outras definições tornam-se necessárias para que se possa dar um direcionamento à modelagem deste nível.

Assim há que se considerar as seguintes proposições definidas a priori:

- . O controle neste nível é apresentado, como um sistema amostrado;
- . a malha externa de controle, amostrado, é realizado no microprocessador;
- . a malha externa abriga outra interna, analógica, fechada na unidade de acionamento.



CER = CONTROLADOR DOS EIXOS
COM REALIMENTAÇÃO

Fig 5.11

Partindo do acima exposto podemos considerar o nível como composto de dois sistemas principais:

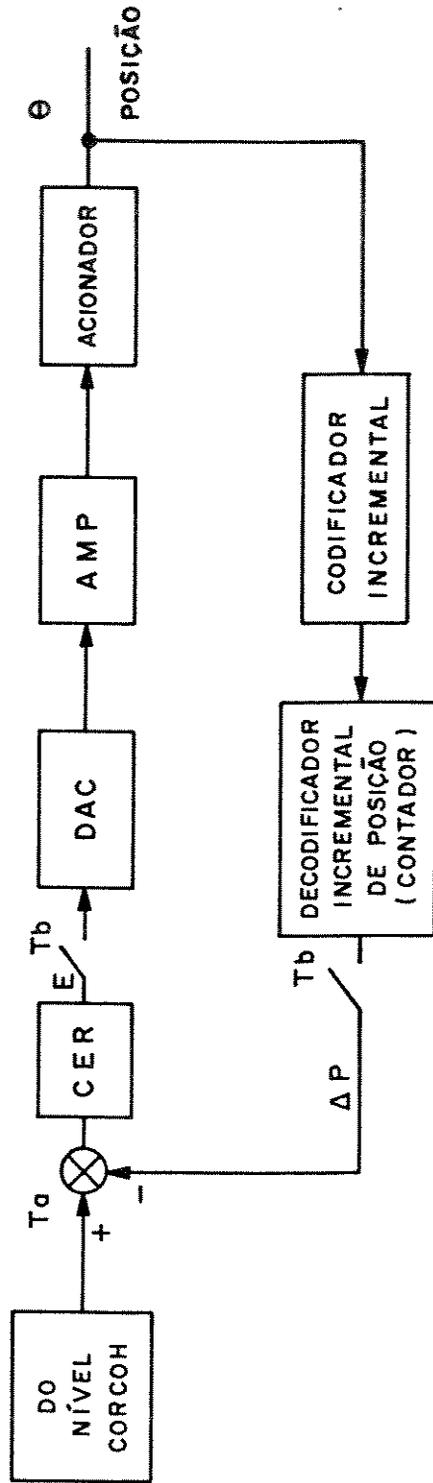
- . sistema microprocessado (micro-processador controlador)
- . sistema de controle do servomecanismo para cada junta.

O sistema microprocessador de controle dos eixos com realimentação (CER) é responsável pela, execução das seguintes tarefas:

- . recebe do nível coordenador os sinais determinantes da posição para a qual deve mover-se cada motor;
- . monitora a realimentação de posição para determinar o erro de posição;
- . determina a instrução a ser enviada ao motor baseada na instrução atual recebida do nível coordenador e no erro de posição detectado;
- . transmite o comando apropriado ao motor;
- . o comando é enviado primeiro ao conversor D/A onde é transformado da forma digital para analógica; aplicado ao amplificador que dirige o motor.
- . a conversão de coordenadas realizadas no nível CORCOH, torna possível a emissão de comandos incrementais de posição. O nível SURCOH recebe em cada CER cada T_a milissegundos, p.ex, $T_a = 30\text{ms}$, esses comandos;
- . Cada CER envia ao nível CORCOH informação definindo como completa a posição de cada eixo, habilitando assim movimento para próxima posição especificada.

O sistema de controle do servomecanismo por sua vez compõe-se, conforme fig. 5.12, para uma das seis juntas:

- . conversor analógico-digital (DAC)
- . amplificador (AMP)
- . acionador
- . codificador incremental ("encoder")



CER = CONTROLADOR DE EIXO COM REALIMENTAÇÃO
AMP = AMPLIFICADOR

Fig 5.12

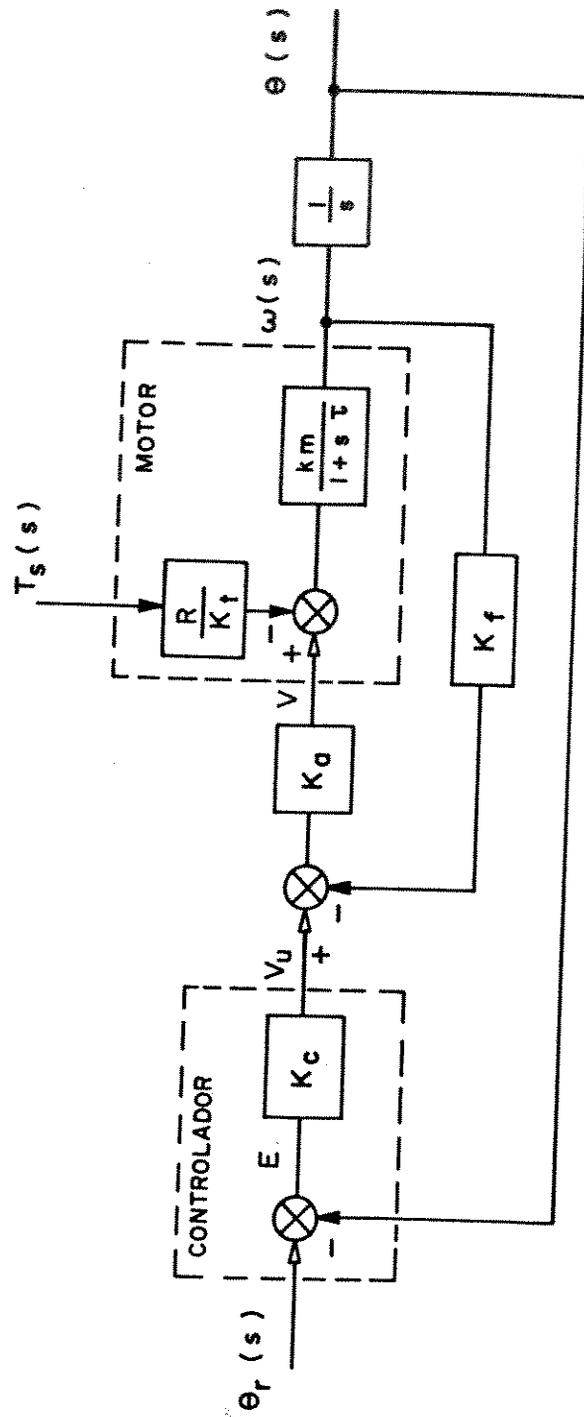


Fig 5.13

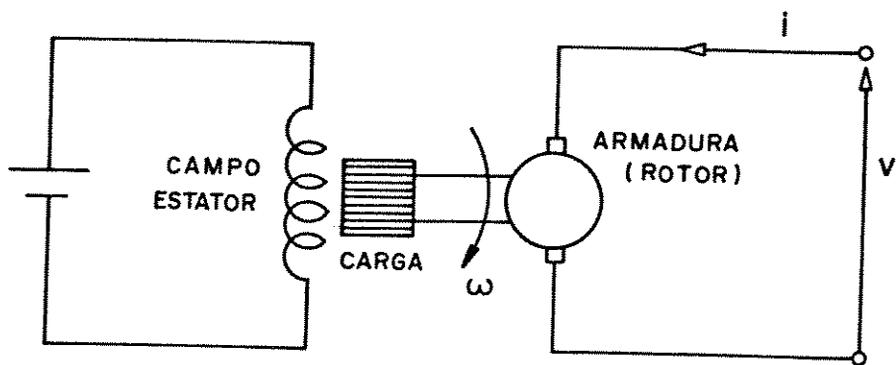


Fig 5.14

. decodificador de posição ("decoder")

e executa as seguintes tarefas:

- . converte o sinal digital proveniente do microprocessador para analógico;
- . o sinal é amplificado e transmitido ao motor;
- . a posição do motor é realimentada ao microprocessador controlador pelo sensor de posição;
- . o sensor incremental ("encoder") gera N pulsos por rotação do motor (frequentemente, dois canais por quadratura);
- . cada ciclo do sensor ("encoder") pode ser dividido em quatro intervalos, desta forma a resolução efetiva é $4N$ quadraturas/rotação;
- . a saída do sensor incremental é aplicada ao decodificador de posição (contador) que conta os pulsos;
- . o decodificador de posição (contador) age como "buffer" entre o sensor incremental ("encoder") e o microprocessador.

O microprocessador "amostra" o sumário do contador à intervalos fixos de tempo T_b .

O valor transferido do contador do computador, ΔP , fig. 5.12, é igual ao deslocamento incremental em BRU_S ("Basic Resolution Unit").

O programa de controle compara a referência informada pelo nível CORCOH com o sumário do contador, determinando o erro de posição.

Aplicando-se (conforme definido anteriormente) o controle de velocidade do conjunto braço/garra do robô, o erro de posição é realimentado cada T_b milisegundos, por exemplo, $T_b = 1 \text{ ms}$, ao conversor analógico digital (DAC).

Este por sua vez fornece uma tensão proporcional à requerida pela velocidade no eixo.

MALHA DE CONTROLE COM AMPLIFICADOR DE TENSÃO

Partindo da definição *à priori* de realização de controle de velocidade no conjunto braço/garra do robô pela monitoração de tensão nos motores das juntas, usamos o amplificador (AMP), de tensão, fig. 5.12, como elemento fundamental.

No diagrama de bloco, fig. 5.13, mostra como a saída do amplificador de tensão, que provê uma saída proporcional à entrada, é capaz de suprir a corrente necessária ao motor.

A saída da malha é determinada em termos de velocidade ou de posição da junta do robô.

O torque T_S , é devido principalmente ao acoplamento de inércia e a atuação da força de gravidade como perturbações ao motor.

A malha de controle da figura 5.13, inclui uma malha interna consistindo de:

- . amplificador de tensão com ganho K_a .
- . motor DC.
- . tacômetro, como realimentador de velocidade, com ganho K_a .

A malha é considerada como um conjunto integrado e é contido no bloco denominado acionador, da fig. 5.12.

A tensão de entrada no motor é

$$V(S) = K_a [V_u(S) - K_f(S)] \quad \text{eq.5.1}$$

como o torque em uma junta do robô pode ser escrito genericamente

$$T = J(\theta_i) \frac{d\omega}{dt} + B(\omega_i)\omega + T_S(\dot{\omega}_i, \omega_i, \theta_i), \text{eq.5.2}$$

onde θ_i , ω_i e $\dot{\omega}_i$ são respectivamente posição, velocidade e aceleração das juntas, e T_S contém as componentes de inércia de acoplamento e os de força centrífuga além de torques estáticos devidos à gravidade e fricção [34],

$$T = K_t I \quad \text{eq.5.3}$$

o torque relaciona-se com a corrente de armadura (motor DC)

$$E = K_V \omega \quad \text{eq. 5.4}$$

a tensão relaciona-se com a velocidade angular de rotação, e a tensão de entrada (V), fornecida pela armadura, fig. 5.14, torna,

$$V - IR = K_V \omega \quad \text{eq. 5.5}$$

onde

R = resistência do circuito de armadura,

IR = queda de tensão através da resistência, com indutância da armadura considerada desprezível,

eliminando I e T a partir das eqs. 5.3, 5.5 e 5.2,

$$\frac{JR}{K_t} \frac{d\omega}{dt} + K_V \omega + \frac{RB\omega}{K_t} = V - \frac{R}{K_t} T_S \quad \text{eq.5.6}$$

Na eq. 5.6, feitas as considerações,

$$K_V \omega \gg IR \frac{B}{IK_t}$$

onde $\frac{B}{IK_t}$ é a relação entre torque de consolis e o torque total,

na prática menor que 10%, IR muito menor que $E = K_V \omega$ podemos escrever

$$\tau \frac{d\omega}{dt} + \omega = K_m V - \frac{KmR}{K_t} T_S \quad \text{eq.5.7}$$

onde $K_m = \text{ganho do motor} = \frac{1}{K_V}$

$\tau = \text{cte mecânica do acionamento e}$

$$\tau = \frac{R}{K_t K_V}$$

e que depende do momento de inércia,

A transformada de Laplace de eq. 5.6, mostra

$$(S) = \frac{K_m V(S) - [RK_m/K_t] T_S(S)}{1 + S \tau} \quad \text{eq. 5.8}$$

Combinando eq. 5.8 com eq. 5.1, temos

$$\omega(S) = \frac{\alpha K_a K_m V_\mu(S) - (R K_m / K_t) \alpha T_S(S)}{1 + S \tau} \quad \text{eq. 5.9}$$

onde α = fator de atenuação =
$$\frac{1}{1 + K_a K_f K_m}$$

Comparando 5.8 com 5.9, vemos que o efeito da realimentação via tacômetro é de forma a:

- . reduzir a constante de tempo
- . reduzir o efeito do torque da carga
- . reduzir não linearidades do amplificador de tensão
- . facilitar o ajuste do ganho final, pelo ajuste do ganho K_f

5.2.4 - SISTEMA DE VISÃO E TATO

O sistema de visão realiza a primeira aproximação e orientação à peça de trabalho.

Durante o processo de pega (através da garra) o sensor de tato confirma ou modifica esta primeira aproximação.

A modificação é dada pela matriz de correção ao sistema de controle do robô, consistindo de valores de correção para localização da pega e orientação do objeto a ser pego.

O sensor de tato dá informações acerca da identidade do objeto eliminando ambiguidades que não podem ser resolvidas apenas pelo sistema de visão.

O sistema de visão, utiliza um computador hospedeiro [21], que manuseia todas as interfaces com dispositivos externos (impressoras, discos), etc.

Este computador poderia ser do tipo compatível com PC AT, com processadores 80286 e 80287, com "floppy drive" de alta densidade e disco rígido 20MB.

O computador está conectado à unidade de processamento de imagem através de interface paralela a qual conecta o barramento de I/O paralelo na placa de visão, capacitando rápida trans

ferência de dados.

A unidade de processamento de imagem poderá ter, por exemplo, três processadores dedicados para controlar e endereçar memória de imagem e para executar análise de imagem e algoritmos de processamento.

A memória da imagem poderia usar resolução de 1024 x 1024 pixels em 8 bits.

A saída da imagem é monocromática ou pseudo colorida dependendo da configuração do sistema.

O soft do programa de controle do robô, projetado para permitir ao operador "ensinar" ao sistema até 200 faces componentes as quais podem ser vistas por quaisquer das câmeras.

Adicionalmente estes componentes podem ser agrupados em qualquer uma das diferentes bibliotecas.

O reconhecimento é conseguido pelo uso de técnicas de reconhecimento de padrões, baseadas em um conjunto de medições disponíveis feitas na forma externa e interna do objeto.

5.3 - CONCLUSÃO

O desenvolvimento de sistemas robóticos inteligentes é um campo de pesquisa relativamente novo. Ele está baseado no aperfeiçoamento dos seguintes tópicos:

- . desenvolvimento de sistemas de aquisição de dados reconhecimento e decisão;
- . desenvolvimento de algoritmos e metodologias para controle de trajetórias utilizando o(s) resultado(s) anterior(es);
- . desenvolvimento de sistemas de controle das junta(s) , utilizando controle adaptativo.

Neste capítulo procurou-se utilizar princípios de controle hierárquico onde cada nível está ligado as suas tarefas características e tem sua "inteligência" específica.

O modelo hierárquico foi definido em três níveis.

No primeiro nível um conjunto de processadores interpreta os comandos externos recebidos, realiza reconhecimento de ima

gem e definição de tarefas baseadas neste reconhecimento.

No segundo, um outro conjunto de processadores, executa o conjunto de tarefas pertinentes ao controle de trajetória, recebendo comandos do nível superior e realimentação do nível inferior.

No terceiro nível, executa-se o controle direto da junta, com um processador por junta, com realimentação dos sensores diretamente locados nelas.

CAPÍTULO 6 - CONCLUSÃO

CONCLUSÃO

Neste trabalho procurou-se estabelecer uma sistemática como uma contribuição ao desenvolvimento da pesquisa em sistemas robóticos inteligentes.

Desta forma voltamo-nos para a análise do comportamento cinemático, dinâmico e de uma estrutura de controle hierarquizada.

No estudo do comportamento cinemático, o desenvolvimento da técnica de controle dos subsistemas de posicionamento e orientação, permite a redução das dificuldades computacionais.

Esta redução é conseguida quando comparada à solução proposta em [4] por R.P. Paul, B. Shimano e G.E. Mayer em que as variáveis de controle das juntas são obtidas de modo sequencial.

O método sequencial obriga ao isolamento das variáveis sucessivas a partir da mão/ferramenta pela pre-multiplicação das sucessivas matrizes "A" inversas.

Em consequência torna-se possível neste método a maior probabilidade de ocorrência de erro devido à frequência com que são realizadas as premultiplicações além do natural gasto de tempo e memória computacional desencorajando a pesquisa desta proposição.

Em contrapartida, o método proposto neste estudo, deriva dos de [1] e [6] divide o sistema manipulador/mão-ferramenta em dois subsistemas menores, conseguindo-se com isso:

- . a partição da solução global em soluções menores, mais simples e menos sujeitas a erro;
- . a combinação das soluções parciais em uma única solução apenas no final da execução;
- . a redução do tédio de formação da solução e do esforço computacional necessário.

No estudo do comportamento dinâmico, através da modelagem de Lagrange - Euler, procurou-se definir uma ferramenta de investigação em processamento simbólico, que permita avaliação da aplicação de outros modelos de robô propostos.

Este estudo está deste modo alicerçado em três pontos básicos:

- . a formulação da dinâmica do braço do robô, através da metodologia de Lagrange-Euler;
- . a utilização dentro do programa do modelo dinâmico simbólico da ferramenta denominada combinadores, segundo a trabalho publicado em [11] , [25] e [31];
- . a implementação deste programa em linguagem Prolog.

Com relação à metodologia de Lagrange-Euler, neste trabalho, segundo as diretrizes propostas em [8] os seguintes pontos podem ser considerados de forma positiva:

- . a derivação do modelo dinâmico de um manipulador baseado em Lagrange-Euler (LE) é simples e sistemática;
- . quando aplicada de formulação de LE em conjunto com a representação matricial de Denavit-Hartenberg (DH) em [1], resulta em um algoritmo compacto.

A utilização dentro do modelo dinâmico simbólico da ferramenta de combinadores descritas por Burge em [31], e tornada popular após a descrição de linguagem funcional por Backus em [10] baseada neles, tem as seguintes vantagens:

- . facilidade na implementação,
- . evita o acesso à estrutura interna das matrizes,
- . evita a indexação das matrizes, tratando as operações com matrizes como números ou objetos.

Por sua vez a implementação do programa em Prolog apresenta vantagens e algumas desvantagens.

Aquelas podem ser consideradas como sendo:

- . fácil e clara implementação,
- . rapidez na programação,
- . linguagem interativa simbólica permitindo grande número de recursos

Por outro lado, algumas desvantagens existem em especial:

- . um "debug" mais trabalhoso quando comparado à linguagens convencionais

- . limitação na capacidade de memória, na base de dados, em especial na versão Arity 3.2, de até 64K bytes por página, para um total de 256 páginas
- . limitação, em especial na versão Arity 3.2, de 255 caracteres (átomos).
- . as limitações na capacidade de memória, tornam a complementação de programas de grande porte, difícil.

A proposta de uma estrutura de controle hierarquizada, é um dos capítulos mais recentes na pesquisa sobre controle inteligente de robôs.

Neste trabalho procurou-se realizar a estrutura em três níveis, o mais elevado deles com maior "inteligência", emitindo comandos e processando as informações sensoriais.

Neste nível, os aspectos relevantes referem-se principalmente à capacidade de reconhecimento de situações complexas definição de padrões e tomada de decisão.

Os outros níveis, o de controle de trajetória e o de controle direto das juntas procuram implementar os modelos disponíveis na área.

Com proposição futura assinalaria a intenção de aperfeiçoamento do modelo dinâmico gerado em Prolog, através de sua otimização em conjunção com uma versão mais poderosa de seu interpretador, bem como a necessidade de aplicação de recursos gráficos.

Por outro lado, também o estudo referente ao controle hierárquico terá perspectivas de aperfeiçoamento em todos os níveis, em especial naquele diretamente ligado com o processamento dos sensores, cujas características de hardware e software, estão em continua evolução.

REFERÊNCIAS BIBLIOGRÁFICAS

- 1 J. Denavit, R.S. Hartenberg. "A Kinematic Notation for Lower-pair Mechanism Based on Matrices" Journal of Applied Mechanism, pp. 215-221, june 1955.
- 2 Richard P. Paul, "Robot Manipulators, Programming and Control", The MIT Press, Cambridge, MA.
- 3 P.G. Ránky, C.Y. Ho., "Robot Modelling-Control and Applications with Software", IFS (Publications) Ltda., UK.
- 4 Paul, R., Shimano B., Meyer G., "Kinematic Control Equation for a Simple Manipulator", IEEE Trans. System, Man and Cybernetics SMC-11, june 449-455, 1981.
- 5 Lee, C.S.G., "Robot Kinetics, Dynamics and Control", IEEE Computer, December: 62-80, 1982.
- 6 Ho, C.Y. and Copland, K.W. "Solution of Kinematic Equation for Robot Manipulators". Digital Systems for Industrial Automation, 1(4): 335-352, 1982.
- 7 Lee, CSG, "Robot Arm Dynamics", Tutorial on Robotics, Lee, C.S.G., Gonzalez, R.C., Fu, K.S., editors, IEEE Computer Society Press, 1984.
- 8 A.K. Bejczy, "Robot Arm Dynamic and Control", JPL Technical Memorandum 33-669, Pasadena, CA.
- 9 T. Mizutani - "ATGS - Autonomous Trajectory Generating Servomechanism" - Pub. Interna - CTI-IA/DAM, 1985.
- 10 Backus, John, "Can Programming be Liberated from the Von Neumann's Style ? A Funtional Style and Its Algebra of Programs", Communications of the ACM, vol. 21., n° 8, pp. 613-641, August 1978.
- 11 Backus, John. "The Algebra of Functional Programs: Functional Level, Reasoning, Linear Equation and Extended Definitions" Lecture Notes on Computer Science #107, Formalization of Programming Concepts, New York, Springer Verlag, 1981.

- 12 C.S.G. Lee, "On the Robotic Manipulation Control". Advances in Automation and Robotics, vol. 1, pp. 65-115.
- 13 K. Hasegawa, T. Mitzutani, "On the Autonomous Trajectory Generating Servomechanism for Manipulation Control", Systems Science, vol. 9, n° 3, Wroclaw, Poland, 1983.
- 14 G.N. Saridis, "Intelligent Robotic Control", IEEE Transaction on Automatic Control, vol. AC-28, n° 5, may 1983.
- 15 A.K. Bejczy, M.I. Vuvković, "An Iterative Manipulator Control System" Proceedings of the Second International Symposium on Mini & Microcomputers in Control, Fort Landerdale, Florida, December, 10-11, 1979.
- 16 Kang G. Shim, Stuart B. Malin, "A Structured Framework for the Control of Industrial Manipulators", IEEE Transactions on Systems, Man, and Cybernetics , vol. SMC 15 n° 1, Jan/Feb., 1985.
- 17 K.S. Fu, "Learning Control Systems - review and outlook", IEEE Transactions on Automatic Control, vol. AC-15, Apr. 1970, pp. 210,221.
- 18 M.A.L. Thathachar, P.S. Sastry, "A New Approach to the Design of Reinforcement Schemes for Learning Automata" IEEE Transactions on Systems, Man, and Cybernetics , vol. SMC-15, n° 1, Jan/Feb., 1985.
- 19 W.G. Keckler, R.E. Larson, "Control of a robot in a partially unknown environment " Automatica, vol. 6, May, 1970, pp. 469, 479.
- 20 R.B. Banerjii, "Theory of Problem-Solving; An aproach to Artificial Intelligence".New York: Elsevier, 1969.
- 21 W.Gerke, H. Bolle, "Integration of Vision and Tactile Sensors in a Robot Control System", IFAC, 10th World Congres on Automatic Control - Preprints - International Federation of Automatic Control, Munick, FRG, July 27-31, 1987.

- 22 Charles P. Neuman, John J. Murray, "Computacional Robot Dynamics Foundations and Applications", Technical Report, Departament of Electrical and Computer Engineering, Carnegie-Mellon University, 1984.
- 23 L. Vecchio, S. Nicosia, F. Nicolò, D. Lentini, "Automatic Generation of Dynamical Models of Manipulators", International Symposium on Industrial Robots, 5:7. March, 1980, Milan
- 24 S.R. Dillon, H. Hemani, "Automated Equation Generation and its Application to Problems in Control", The Ohio State University report n° GK-25292.
- 25 Siqueira Neto, JL; Alves J.B.M.; Pereira A.E.C., "Symbolic Computation Applied To Robot Dynamic Modelling" 16th ISIR-International Symposium on Industrial Robot", Brussels, Belgium, 29-9/02-10, 1986.
- 26 Kang G. Shin, Stuart B. Malin, "A Hierarchically Distributed Robot Control System", Proceedings Compsac, october, 1980, pg. 814-820.
- 27 G. Hirzinger, "Sensory Feedback in Robotics - State - of - The - Art in Research and Industry", IFAC, 10th World Congress on Automatic Control -Preprints - International Federations of Automatic Control, Munich, FRG, July, 27-31, 1987.
- 28 Y. Koren, "Robotics for Engineers", McGraw Hill Book Company, 1985.
- 29 Dillon, S.R.: "Computer Assisted Equation Generation in Linkage Dynamics", PhD dissertation Departament of Electrical Engineering, Ohio State University, Columbus, OH, August 1973.
- 30 Murray, John J. and Neumann, Charles P., "ARM: An Algebraic Robot Dynamic Modelling Program", Proceedings of First International IEEE Conference on Robotics, Atlanta, USA, pp. 103-114, March, 1984.

- 31 Burge, Willian H., "Recursive Programming Techniques", Addison-Wesley, 1975.
- 32 Darlington, J.; R.M. Burstall, "Some Transformations for Developing Recursive Programs, Journal of the ACM, vol.24, nº 1, pg. 444, 1977.
- 33 Vukobratovic, M. e Kircanski, N.; "A Method for Computer-Aided Construction of Analytical Models of Robotic Manipulators", Proceedings of the First International IEEE Conference on Robotics, Paul, R.P. ed., Atlanta, GA, Marcj, 13-15, 1984, pp. 519-528.
- 34 Alves, J.B.M., "Controle de Robô", Cartgraf, Campinas, SP, 1988.