

JOSÉ ANTÓNIO TENREIRO MACHADO

**GESTÃO DOS RECURSOS ESTRUTURAIS  
NO  
CONTROLO DE ROBOTS MANIPULADORES**

PORTO 1988

Faculdade de Engenharia da Universidade do Porto  
Departamento de Engenharia Electrotécnica e de Computadores

GESTÃO DOS RECURSOS ESTRUTURAIS  
NO  
CONTROLO DE ROBOTS MANIPULADORES

José António Tenreiro Machado

Tese Submetida para  
Provas de Doutoramento em  
Engenharia Electrotécnica  
e de Computadores  
Porto, 1988

à Alexandra e ao José Pedro

## Agradecimentos

Ao meu orientador científico, Prof. J. L. Martins de Carvalho, desejo expressar o meu agradecimento sincero, pelo apoio e disponibilidade manifestados no desenvolvimento do trabalho aqui apresentado.

Encontro-me também grato ao Prof. J. S. Silva Matos e ao Eng. António M. C. Costa, pelos seus contributos em alguns dos tópicos referidos nesta tese.

Desejo ainda agradecer ao Prof. J. C. D. Marques dos Santos pelo seu empenho no desenvolvimento na FEUP de diversos estudos na área da robótica o que, em certa medida, constitui uma motivação adicional para o presente estudo.

Aproveito também esta oportunidade, para manifestar o meu agradecimento à NATO, INIC, INESC, Fundação Luso-Americana e Fundação Calouste Gulbenkian, pelos apoios financeiros concedidos tendo em vista a minha participação em diversas conferências.

## ÍNDICE

1. INTRODUÇÃO	1
Referências	10
2. DINÂMICA E CONTROLO DE ROBOTS MANIPULADORES	11
2.1. Introdução	12
2.2. Estudo dinâmico de um robot manipulador	13
2.3. Compensação do robot manipulador	21
2.3.1. Compensação através de dispositivos mecânicos	22
2.3.2. Compensação por programação	25
2.4. Discussão e conclusões	30
Referências	32
3. CONTROLO EFICIENTE DE ROBOTS MANIPULADORES	35
3.1. Introdução	36
3.2. Enquadramento do controlador convencional	37
3.3. Melhoramento da arquitectura do controlador	43
3.4. Implementação da nova arquitectura de controlo	47
3.5. Discussão e conclusões	53
Referências	56
4. CONTROLO DE ROBOTS MANIPULADORES ATRAVÉS DE UM SISTEMA DE ESTRUTURA VARIANTE	60
4.1. Introdução	61
4.2. Algoritmo de controlo	63
4.3. Resultados e simulações	71
4.4. Discussão e conclusões	87
Apêndice 1	90
Apêndice 2	93
Referências	96

5. ASPECTOS COMPUTACIONAIS NO CONTROLO DE ROBOTS MANIPULADORES	99
5.1. Introdução	100
5.2. Optimização da gestão dos recursos computacionais: O compilador dedicado	106
5.2.1. Algumas considerações relativas à síntese do compilador dedicado	106
5.2.2. Computação híbrida usando a soma aritmética e a álgebra de Boole	113
5.2.3. Computação híbrida usando as soma e multiplicação aritméticas, e a álgebra de Boole	117
5.2.4. Procedimentos para a optimização do método híbrido	118
5.3. Implementação do compilador dedicado no robot 2R	131
5.4. Arquitecturas computacionais paralelas sugeridas pelo compilador dedicado	144
5.5. Discussão e conclusões	146
Apêndice 1	150
Apêndice 2	153
Apêndice 3	156
Referências	157
6. CONCLUSÃO	162
Referências	179

CAPÍTULO 1

INTRODUÇÃO

## 1. INTRODUÇÃO

Nas últimas décadas tem-se assistido ao alargamento do espectro científico particularmente em zonas de forte componente tecnológica. A robótica é uma das áreas que tem conhecido essa evolução. Se bem que os conceitos associados à robótica existam desde a Grécia clássica, foi somente no século vinte que estes se converteram em realidade. Já Aristóteles escrevia "Se cada instrumento pudesse realizar o seu próprio trabalho, obedecendo ou antecipando-se à vontade dos outros... se a lançadeira pudesse tecer, e a palheta tocar a lira, sem a mão para as guiar então os chefes não necessitariam de servos nem os donos de escravos". A própria palavra robot somente em 1921 foi inventada, quando o dramaturgo checo Karel Kapec a introduziu na sua peça "Os robots universais de Rossum", com base na palavra eslava robota que significa trabalho pesado. No entanto, e na opinião do autor, deve-se aos escritores de ficção científica o verdadeiro estímulo para a concretização desta "utopia". Tem-se verificado que muitas criações da ficção científica rapidamente se tornam realidade; tal foi o caso das viagens espaciais, dos computadores e ... da robótica. Com efeito, este estilo literário parece influenciar sobremaneira a imaginação dos engenheiros e cientistas. Assim, em 1960, com base na experiência existente nos campos dos telemanipuladores e máquinas ferramenta de controlo numérico, George C. Devol construiu o primeiro robot industrial. A partir desta data verificou-se uma afirmação deste ramo do conhecimento como uma ciência autónoma. De carácter multidisciplinar, fazendo uso de elementos ligados às engenharias mecânica, eléctrica, electrónica, de computadores e outras, tem mostrado a sua importância em

áreas tão dispares como na exploração espacial, na exploração sub-aquática, na biónica ou na indústria. É neste último campo que se tem investido maior esforço, talvez devido à sua forte componente sócio-económica. A indústria automóvel é disso um bom exemplo: robots manipuladores para pintura, soldadura por pontos e soldadura por arco, foram rapidamente admitidos como parte integrante da força laboral dessas empresas.

O alargamento da sua utilização a outras áreas industriais tem-se, no entanto, verificado ser mais difícil. Efectivamente, ao contrário das utilizações referidas, as novas aplicações são mais exigentes, e têm mostrado que mesmo em casos aparentemente simples, os robots manipuladores são ainda uma fraca tentativa de cópia do braço humano.

A investigação científica, tem procurado dar resposta a estas necessidades. A evolução tecnológica em geral, influencia positivamente cada caso em particular; todavia, existe uma componente própria a cada ramo científico, e a robótica não é excepção. Assim, investigação no campo de novas estruturas mecânicas, nomeadamente robots de accionamento directo e robots flexíveis, novas arquitecturas de controladores jogando com múltipla informação sensorial (força/binário, visão, etc.), algoritmos inteligentes de planeamento de trajectória, linguagens eficientes de descrição das tarefas e movimentos, e outras, são campos bem conhecidos como áreas de possível desenvolvimento. Este conhecimento aliado à necessidade duma rentabilidade no investimento a realizar, levou ao aparecimento do que, em certa medida, pode ser chamado como "moda científica". Sem pretender ser dogmático, é todavia opinião do autor, que não é necessariamente no mesmo sentido da "moda" o melhor, ou único, caminho a

percorrer. Em particular numa ciência de carácter tão inovador, onde os pilares foram tão recentemente construídos, há certamente muitas paredes para erigir até o edifício estar concluído. Por outras palavras, pode-se dizer que o conhecimento científico no campo da robótica constitui uma "rede de conhecimento", existindo áreas inexploradas quer na, claramente definida, periferia da rede, quer no espaço em aberto no interior de cada malha da rede.

É com base nesta filosofia que deve ser encarado o trabalho apresentado seguidamente. Comum a todos os capítulos está o conceito base de "gestão eficiente dos recursos estruturais". Assim, é desenvolvida uma utilização criteriosa, aplicada a vários sub-sistemas, de modo a torná-los industrialmente realizáveis, sem contudo ficar fortemente dependente de uma dada evolução tecnológica em particular.

Todos os sistemas são estudados com base num robot 2R, isto é, com dois graus de liberdade rotacionais. Esta estratégia tem sido seguida por muitos investigadores, pois permite uma fácil implementação e interpretação, sem contudo perder generalidade, já que, para os fins em vista, se verificam todos os fenómenos que aparecem em robots manipuladores com maior número de graus de liberdade.

Também foi tentado dar um carácter autónomo a cada capítulo, tendo em atenção o aspecto algo incómodo de ter que se folhear sistematicamente o trabalho, a fim de se reencontrar uma dada expressão ou resultado.

O capítulo 2 forma a base de todo o trabalho. A dinâmica de robots é complexa e não-linear, o que tem sido um motivo para a literatura científica a apresentar de forma algo complexa e de

difícil abordagem. Este tema é caracterizado de uma forma "desmistificadora", o que leva ao natural aparecimento de pistas de investigação.

No capítulo 3, é desenvolvida uma dessas pistas. Certos fenômenos que aparecem na dinâmica dos robots manipuladores são mais "importantes" que outros, quando examinados sob o ponto de vista de controlo. Daí se desenvolver um controlador não-linear que toma em consideração o peso relativo dessas influências.

Uma filosofia de implementação diferente leva ao controlador apresentado no capítulo 4. Aí tira-se partido da dualidade cálculo em tempo real/instrumentação e sensores, sempre existente em sistemas deste tipo. O algoritmo de controlo desenvolvido neste capítulo, beneficia de grandes simplificações à custa de um aumento do número de sensores necessários. Por outras palavras, verifica-se um deslocamento de exigências de potência de cálculo em tempo real, para necessidades sensoriais. Além disso uma estrutura simplificada do algoritmo, leva a objectivos e capacidades claramente definidas, o que é aliciante devido à dinâmica algo complexa do sistema a controlar.

Nem só a gestão eficiente do controlador é capaz de proporcionar melhores resultados. No capítulo 5, pretende-se realizar uma melhor administração dos recursos estruturais existentes no ambiente robot mais computador. A partir da verificação das capacidades e limitações quer do computador quer do sistemas periféricos (electrónicos, mecânicos e outros), é desenvolvida uma nova filosofia de cálculo com implicações não só nos computadores sequenciais convencionais, mas também em arquitecturas computacionais paralelas.

Finalmente, no capítulo 6, são apresentadas conclusões, bem

como algumas pistas de possível desenvolvimento futuro.

As facetas correspondentes à gestão eficiente dos recursos estruturais, referidas nestes capítulos, podem ainda ser interpretadas à luz de certos conceitos recentemente emergentes na comunidade científica. De facto mostra-se em franca evolução a área designada por "controlo inteligente", e que não é mais do que a fusão dos conceitos associados ao controlo clássico, à inteligência artificial e à investigação operacional (Fig. 1.1). A comunidade científica espera com esta estratégia obter um todo maior e melhor que a soma das partes. Assim, particularmente em zonas de conhecimento híbrido, com a utilização simultânea de técnicas de escolas diferentes, é de esperar o aparecimento de soluções mais eficazes.

Este trabalho pretende ser um vector de ligação entre diferentes perspectivas tendo por base a teoria de sistemas clássica. Os capítulos 2 e 3 mostram que no caso do controlo de robots manipuladores através do controlador não-linear conhecido por "método do binário calculado" (computed torque method), é possível evitar as complexas técnicas de análise de convergência de sistemas não-lineares e adoptar "inteligentemente" (i.e. como alternativa mais eficiente) métodos simples e bem conhecidos. Esta perspectiva garante não somente uma simplificação e clarificação do estudo, mas também a obtenção de informação adicional, como, por exemplo, a margem de fase, que dificilmente se consegue obter com os métodos de análise convencionais. Aliás, também o capítulo 4 se debruça sobre uma problemática muito semelhante; os sistemas de controlo de estrutura variante são conhecidos pela sua robustez bem como por dispensarem a necessidade de modelos

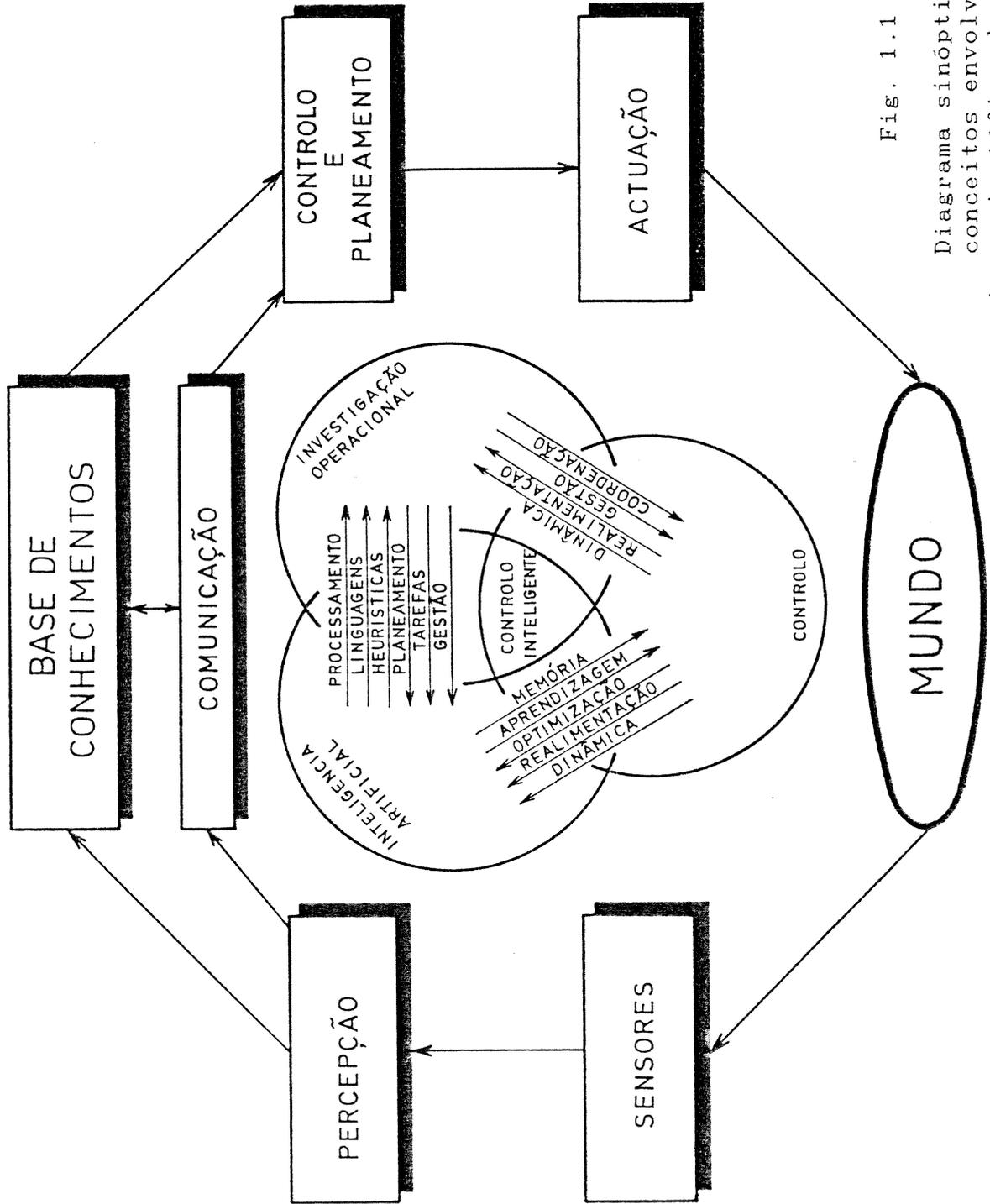


Fig. 1.1

Diagrama sinóptico dos conceitos envolvidos na área científica designada por "controlo inteligente"

rigorosos da planta a controlar. Desde que seja garantida uma condição - suficiente mas não necessária - do tipo  $s\dot{s} < 0$  (s será definido no capítulo 4) fica assegurada a convergência do sistema; todavia, não é acessível nenhuma informação quanto à "qualidade" dessa estabilidade. Neste capítulo é encontrado um conjunto de condições que permite projectar o controlador segundo esta ordem de ideias, contudo o autor prescinde dessa exactidão - suficiente mas não necessária - em troca de uma análise algo heurística baseada nos conceitos clássicos de teoria dos sistemas. Mais uma vez se obtém um método, que para além de assegurar a estabilidade, fornece informação quanto ao desempenho do sistema. Esta incursão, algo mais evidente que nos capítulos anteriores, num campo próximo da inteligência artificial serve também de motivação ao capítulo 5. A ligação entre as duas áreas passa não somente pelo uso de heurísticas, mas alternativamente pela representação do conhecimento através de "novas" álgebras. É bem conhecida a tendência dos cientistas para a adopção de modelos do mundo físico baseados nos conceitos do cálculo diferencial e integral. Acontece pois com naturalidade o aparecimento de dúvidas quanto à necessidade e benefícios decorrentes deste monopólio, tanto mais evidente quanto o ser humano não usa nos seus processos biológico-motores-mentais este tipo de álgebra. O capítulo 5 situa-se nesta perspectiva já que é conseguida uma descrição da dinâmica dos robots manipuladores através de uma álgebra diferente - a algebra de Boole. Os benefícios resultantes mostram mais uma vez que a "qualidade" da nova representação é, para certas aplicações, bem superior à convencional.

Para concluir esta introdução poder-se-á ainda dizer que este trabalho suscita muitas outras pistas de investigação. Um

debate de algumas dessas ideias bem como algumas respostas já obtidas, são matéria de discussão no capítulo 6.

## REFERÊNCIAS

- [1.1] Isaac Asimov  
Robot Completo  
Publicações Europa-América
- [1.2] Sebastião J. Formosinho  
Nos Bastidores da Ciência: Resistência dos Cientistas à  
Inovação Científica  
Gradiva, 1988.
- [1.3] George N. Saridis  
Toward the Realization of Intelligent Controls  
Proc. IEEE, vol. 76, n. 8, pg. 1115-1133, Aug., 1979.
- [1.4] George N. Saridis  
Intelligent Robotic Control  
IEEE Trans. on Automat., Contr., vol. AC-28, pg. 547-  
557, May, 1983.
- [1.5] C. S. George Lee  
Robot Arm, Kinematic, Dynamics and Control  
IEEE Computer, pg. 62-80, Dec., 1982.
- [1.6] J. Y. S. Luh  
An Anatomy of Industrial Robots and Their Controls  
IEEE Trans. Automat., Contr., vol. AC-28, pg. 133-153,  
Feb., 1983.
- [1.7] J. Y. S. Luh  
Industrial Robots - A Tutorial  
IEEE Trans. Syst., Man, Cybern., vol. SMC-13,  
pg. 298-316, May/June, 1983.
- [1.8] Gerardo Beni and Susan Hackood  
Recent Advances in Robotics  
John Wiley & Sons, 1985.
- [1.9] Wesley E. Snyder  
Industrial Robots: Computer Interfacing and Control  
Prentice-Hall Inc., 1985.
- [1.10] M. Vidyasagar  
System Theory and Robotics  
IEEE Control Syst. Magazine, vol. 7, pg. 16-17, April,  
1987.
- [1.11] M. Kokar, A. Meystel  
Panel Discussion on Machine Learning in Intelligent  
Control  
IEEE Intelligent Control Gazette, pg. 6-7, Summer,  
1988.

CAPÍTULO 2

DINÂMICA E CONTROLO  
DE  
ROBOTS MANIPULADORES

## 2. DINÂMICA E CONTROLO DE ROBOTS MANIPULADORES

### 2.1. INTRODUÇÃO

A teoria de sistemas clássica é bem conhecida e permite fácil e rapidamente obter conclusões sobre a estabilidade, tipo de resposta temporal e outras características de sistemas lineares. No caso dos robots manipuladores, esta teoria é inadequada, dada a presença de uma dinâmica fortemente não-linear. Devido a este motivo a investigação tem conduzido ao aparecimento de uma nova geração de controladores não-lineares [2.1-2.4]. Simultaneamente, tem também sido desenvolvido trabalho com vista a obter um relacionamento entre esta dinâmica não-linear e as tradicionais técnicas de estudo de sistemas lineares [2.5-2.14]. Apesar disso, a complexidade do problema condicionou a obtenção de resultados sistemáticos, não permitindo a desejada clarificação ou, o mesmo será dizer, a obtenção de um melhor conhecimento quanto à estabilidade, sensibilidade relativamente a modelizações imperfeitas e outros fenómenos envolvidos no funcionamento desses controladores.

Este capítulo pretende interligar os dois campos, ou seja, facilitar o uso das ferramentas matemáticas clássicas, adaptando-as ao caso em estudo.

No parágrafo 2.2 a introdução de linearizações em redor de um dado ponto de funcionamento permite a obtenção de conclusões sobre:

- Modificação da função de transferência com as variações das coordenadas nos eixos do robot manipulador
- Estabilidade dessas funções de transferência
- Acoplamento entre as diversas saídas

Baseados nessas conclusões, no parágrafo 2.3 são determinadas quais as compensações necessárias, e conseqüentemente um algoritmo relativamente ao qual algumas estruturas usadas industrialmente e certos sistemas propostos na literatura científica, são casos particulares de implementação.

## 2.2. ESTUDO DINÂMICO DE UM ROBOT MANIPULADOR

As equações dinâmicas de um robot manipulador podem obter-se, facilmente, a partir do Lagrangeano [2.15-2.17]

$$T = d/dt(\partial L/\partial \dot{q}) - \partial L/\partial q \quad (2.1a)$$

$$L = K - P \quad (2.1b)$$

onde K é a energia cinética

P é a energia potencial

T é o binário/força generalizado

q é a coordenada generalizada

t é o valor da variável tempo

Na sua forma matricial, para um robot com n graus de liberdade, essas equações são da forma [2.18-2.19]

$$T = J(q)\ddot{q} + C(q, \dot{q}) + G(q) \quad (2.2)$$

onde J(q) é a matriz (simétrica) inercial, de dimensão n x n

C(q,  $\dot{q}$ ) é o vector, n dimensional, dos binários/forças Coriolis/centrípetos

G(q) é o vector, n dimensional, dos binários/forças gravitacionais.

Para o caso do robot 2R (Fig. 2.1), obtêm-se as seguintes expressões:

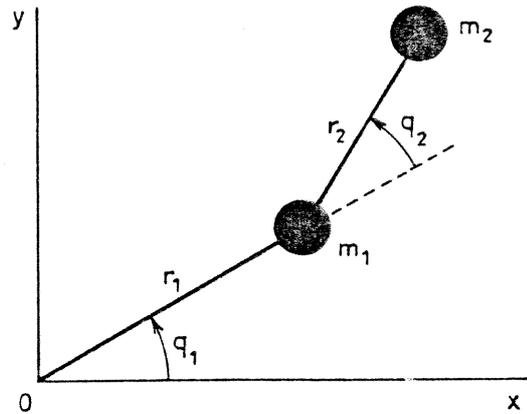


Fig. 2.1 Robot 2R

$$J(q) = \begin{bmatrix} m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1 & m_2 (r_2^2 + r_1 r_2 C_2) \\ m_2 (r_2^2 + r_1 r_2 C_2) & m_2 r_2^2 + J_2 \end{bmatrix} \quad (2.3a)$$

$$C(q, \dot{q}) = \begin{bmatrix} -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 - m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ m_2 r_1 r_2 S_2 \dot{q}_1^2 \end{bmatrix} \quad (2.3b)$$

$$G(q) = \begin{bmatrix} g[m_1 r_1 C_1 + m_2 (r_2 C_{12} + r_1 C_1)] \\ g m_2 r_2 C_{12} \end{bmatrix} \quad (2.3c)$$

onde  $C_1 = \cos(q_1)$ ,  $C_2 = \cos(q_2)$ ,  $C_{12} = \cos(q_1 + q_2)$  e  $S_2 = \sin(q_2)$ .

Tendo em vista uma linearização destas equações [2.20-2.22] torna-se necessário considerar pequenos desvios em redor de um ponto de equilíbrio  $(q_{10}, q_{20}, T_{10}, T_{20})$ , ou seja:

$$\begin{bmatrix} T_1 \\ T_2 \end{bmatrix} = \begin{bmatrix} T_{10} \\ T_{20} \end{bmatrix} + \begin{bmatrix} \delta T_1 \\ \delta T_2 \end{bmatrix} \quad (2.4a)$$

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} q_{10} \\ q_{20} \end{bmatrix} + \begin{bmatrix} \delta q_1 \\ \delta q_2 \end{bmatrix} \quad (2.4b)$$

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \delta \dot{q}_1 \\ \delta \dot{q}_2 \end{bmatrix} \quad (2.4c)$$

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \delta \ddot{q}_1 \\ \delta \ddot{q}_2 \end{bmatrix} \quad (2.4d)$$

Substituindo estas equações em (2.3) e considerando somente os termos constantes e de primeira ordem, obtem-se:

$$\begin{bmatrix} T_{10} \\ T_{20} \end{bmatrix} = g \begin{bmatrix} m_1 r_1 C_{10} + m_2 (r_1 C_{10} + r_2 C_{120}) \\ m_2 r_2 C_{120} \end{bmatrix} \quad (2.5a)$$

$$\begin{bmatrix} \delta T_1 \\ \delta T_2 \end{bmatrix} = \begin{bmatrix} (m_1 + m_2) r_1^2 + m_2 r_2^2 + 2 r_1 r_2 m_2 C_{20} + J_1 & m_2 r_2^2 + r_1 r_2 m_2 C_{20} \\ m_2 r_2^2 + r_1 r_2 m_2 C_{20} & m_2 r_2^2 + J_2 \end{bmatrix} \begin{bmatrix} \delta \ddot{q}_1 \\ \delta \ddot{q}_2 \end{bmatrix} - g \begin{bmatrix} m_1 r_1 S_{10} + m_2 (r_1 S_{10} + r_2 S_{120}) & m_2 r_2 S_{120} \\ m_2 r_2 S_{120} & m_2 r_2 S_{120} \end{bmatrix} \begin{bmatrix} \delta q_1 \\ \delta q_2 \end{bmatrix} \quad (2.5b)$$

onde  $C_{20} = \cos(q_{20})$ ,  $S_{10} = \sin(q_{10})$

$C_{120} = \cos(q_{10} + q_{20})$ ,  $S_{120} = \sin(q_{10} + q_{20})$ .

Uma descrição no plano  $s$ , obtem-se facilmente aplicando a Transformada de Laplace a (2.5b), resultando:

$$T(s) = G(s)Q(s) \quad (2.6a)$$

$$G_{11}(s) = [(m_1 + m_2) r_1^2 + m_2 r_2^2 + 2 r_1 r_2 m_2 C_{20} + J_1] s^2 - g [m_1 r_1 S_{10} + m_2 (r_1 S_{10} + r_2 S_{120})] \quad (2.6b)$$

$$G_{12}(s) = G_{21}(s) = (m_2 r_2^2 + r_1 r_2 m_2 C_{20}) s^2 - g m_2 r_2 S_{120} \quad (2.6c)$$

$$G_{22}(s) = (m_2 r_2^2 + J_2) s^2 - g m_2 r_2 S_{120} \quad (2.6d)$$

onde  $T_i(s) = L\{T_i\}$ ,  $Q_i(s) = L\{q_i\}$  e  $i=1,2$ .

A equação matricial (2.6), corresponde à chamada descrição do sistema inverso. Para obter a descrição directa, é necessário inverter-la, vindo:

$$Q(s) = G(s)^{-1}T(s) = [N(s)/d(s)]T(s) \quad (2.7a)$$

$$N_{11}(s) = s^2 [(m_1 + m_2)r_1^2 + m_2r_2^2 + 2r_1r_2m_2C_{20} + J_1] - g[m_1r_1S_{10} + m_2(r_1S_{10} + r_2S_{120})] \quad (2.7b)$$

$$N_{12}(s) = N_{21}(s) = -s^2(m_2r_2^2 + r_1r_2m_2C_{20}) + gm_2r_2S_{120} \quad (2.7c)$$

$$N_{22}(s) = s^2(m_2r_2^2 + J_2) - gm_2r_2S_{120} \quad (2.7d)$$

$$d(s) = N_{11}(s)N_{22}(s) - N_{12}(s)^2 \quad (2.7e)$$

Verifica-se que os termos Coriolis/centrípetos têm apenas uma influência de segunda ordem, em termos de estabilidade. Por outro lado, somente os termos inerciais e os gravitacionais são responsáveis pela localização dos pólos e zeros da função de transferência. Além disso, verifica-se que:

a) Existe acoplamento entre as saídas, devido a não serem nulos  $N_{ij}(s) = N_{ji}(s)$  ( $i \neq j$ )

b) As variações dos pólos e zeros, resultam da dependência dos termos inerciais e dos gravitacionais, com  $q_{10}$  e  $q_{20}$ . Para manipuladores situados no espaço, com ausência de gravidade, ou manipuladores com estruturas horizontais, onde os efeitos gravitacionais são suportados pelos rolamentos dos eixos, tem-se um sistema com todos os pólos na origem do plano  $s$ , e ganho variável devido aos termos inerciais dependentes da configuração

c) A equação característica  $d(s)$  não contém potências ímpares, pelo que as suas raízes ocorrem aos pares, ou no eixo imaginário ou no eixo real, tendo grandeza idêntica e sinais opostos

d) A cada elo da estrutura mecânica do robot posicionado

para baixo, corresponde um par de pólos conjugado imaginário puro; a cada elo posicionado para cima corresponde um par de pólos reais, simétricos relativamente à origem do plano  $s$  (Fig. 2.2). Não é pois surpreendente quando alguns fabricantes de robots manipuladores [2.23-2.25] e investigadores [2.26], chamam a atenção para o desempenho algo superior de estruturas de robots que funcionam com elos para baixo

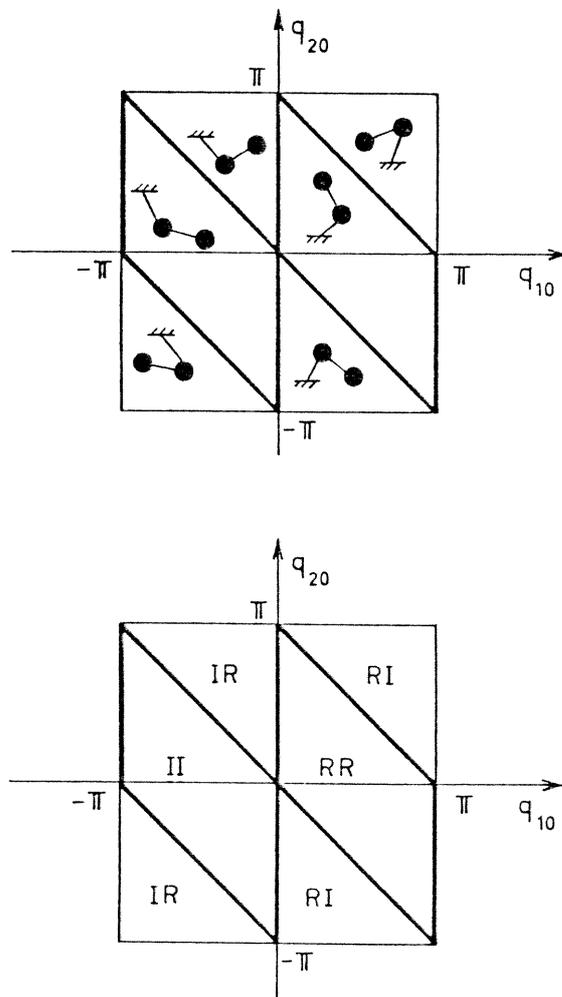


Fig. 2.2 Localização das raízes da equação característica  $d(s)$  para o robot 2R.  
 I Par de raízes complexas conjugadas  
 R Par de raízes reais e simétricas  
 — raiz dupla, com valor zero.

e) O valor absoluto dos pólos aumenta quando os elos se aproximam da vertical e decrescem até se anularem quando estes se aproximam da horizontal

f) Manipuladores com  $n$  graus de liberdade têm funções de transferência com  $2n$  pólos e  $2(n-1)$  zeros, excepto quando  $k$  elos estão horizontais. Nesta situação o sistema tem  $2(n-k)$  pólos e  $2(n-k-1)$  zeros quando  $n > k$ , ou somente dois pólos nulos quando  $n = k$ .

Na figura 2.3 está representado o lugar geométrico dos pólos e zeros de  $G_{11}(s)$ , quando ambos os elos do robot 2R estão alinhados (i.e.  $q_{20} = 0$ ). Para as outras funções de transferência obtêm-se gráficos semelhantes.

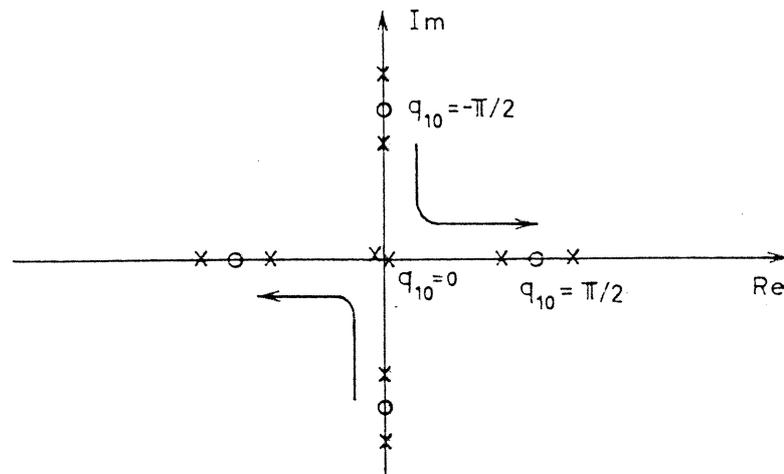


Fig. 2.3 Variação no plano  $s$ , das raízes da equação característica  $d(s)$ , com  $q_{20} = 0$  e para valores de  $q_{10}$  compreendidos entre  $-\pi/2$  e  $\pi/2$ .

Nos sistemas reais, a existência de atrito viscoso, não considerado nos modelos (2.2) e (2.3), melhora ligeiramente a estabilidade, já que "puxa" os pólos para o semiplano esquerdo.

Estas considerações possibilitam a derivação de algumas

conclusões sobre os requisitos necessários no controlo de manipuladores.

Como é óbvio, uma realimentação de posição e velocidade baseada numa só variável, é incapaz de estabilizar o sistema quando este tem elos posicionados para cima, já que as respectivas funções de transferência possuem pólos e zeros no semiplano direito. No entanto é prática industrial o uso de estruturas com um controlador PID em cada eixo do robot (Fig. 2.4).

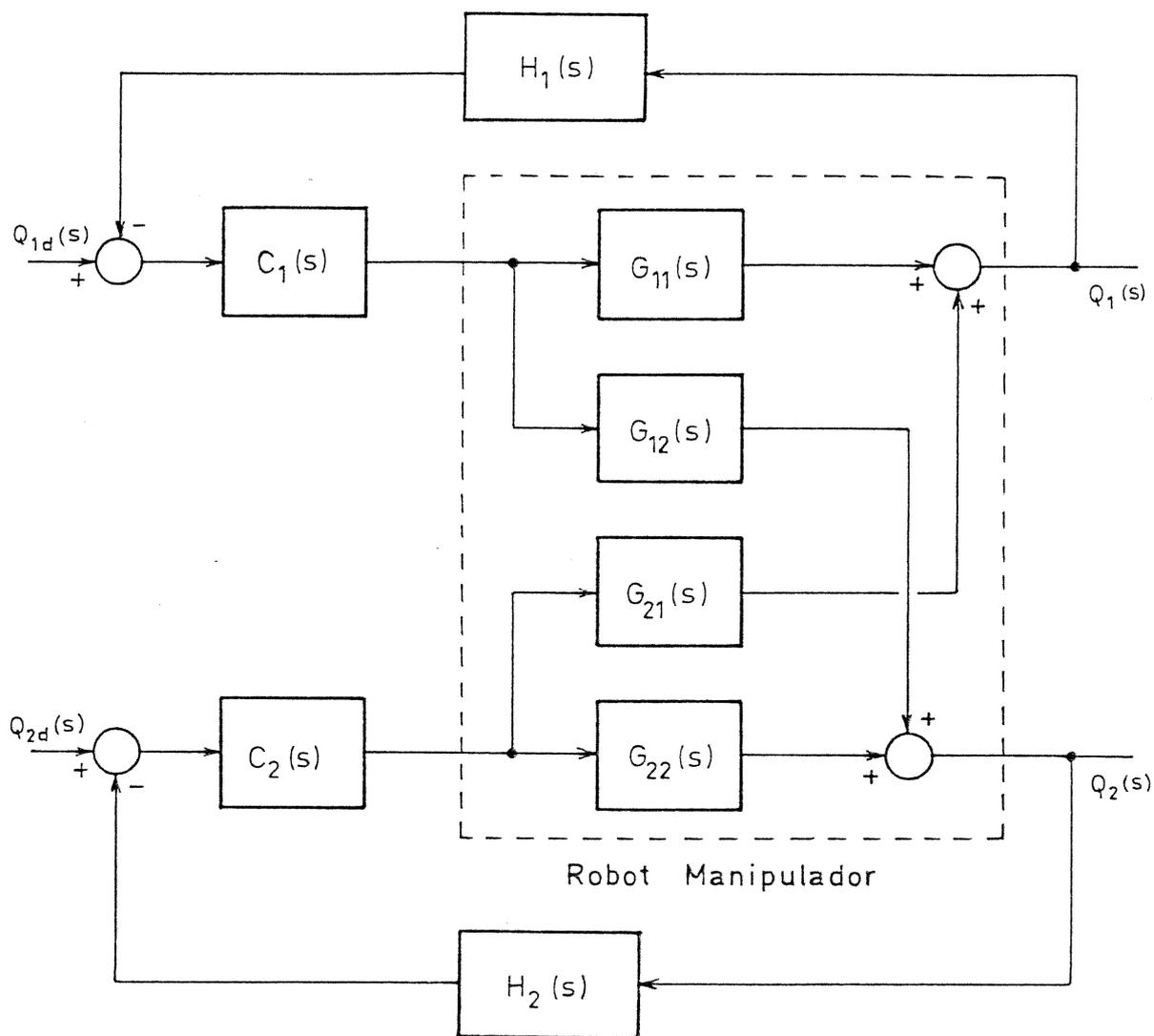
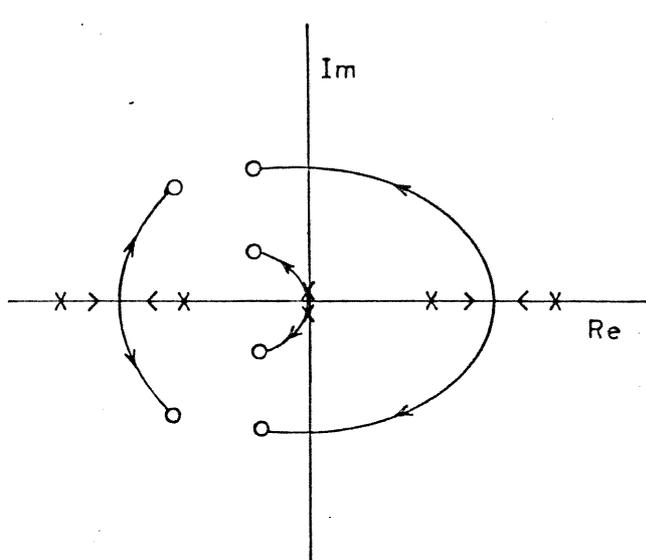
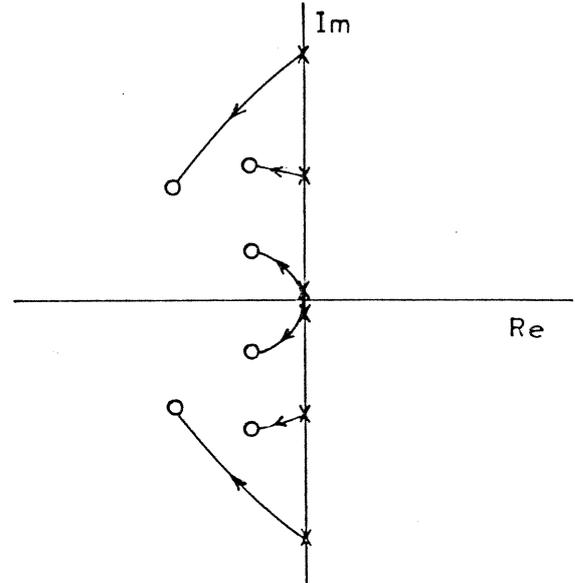


Fig. 2.4 Estrutura de controlo PID, distribuída por cada eixo do robot, e usual a nível industrial.  
 $H_i(s) = h_{pi} + h_{vi}s$ ;  $C_i(s) = c_{ri}/s + c_{pi} + c_{di}s$ ;  $i=1,2$ .



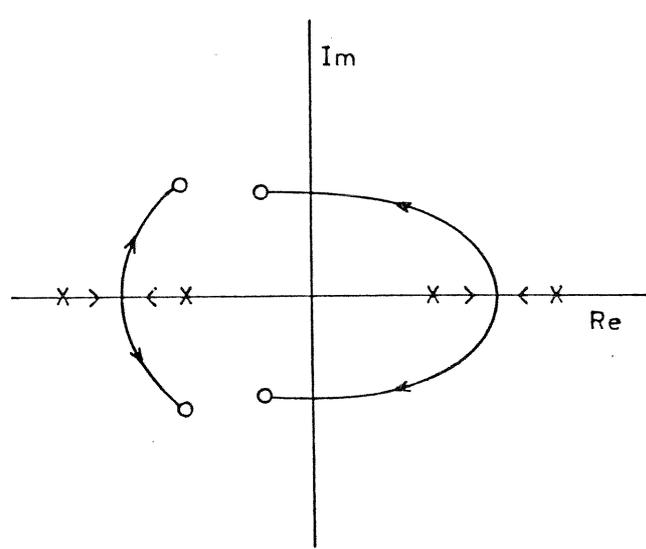
$q_{10} = \pi/2, q_{20} = 0$

Controlo PID



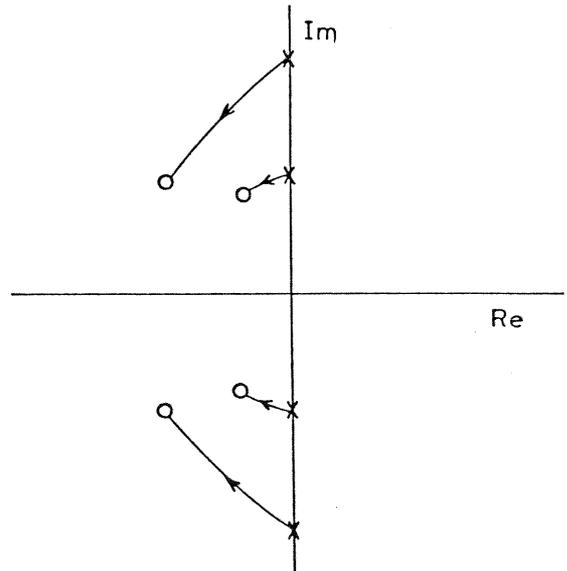
$q_{10} = -\pi/2, q_{20} = 0$

Controlo PID



$q_{10} = \pi/2, q_{20} = 0$

Controlo PD



$q_{10} = -\pi/2, q_{20} = 0$

Controlo PD

Fig. 2.5 Possíveis lugares de raízes, para a estrutura de controlo representada na figura 2.4.

A partir do modelo linear anteriormente obtido, este sistema pode ser descrito pela expressão:

$$\begin{bmatrix} Q_1(s) \\ Q_2(s) \end{bmatrix} = 1/d'(s) \begin{bmatrix} C_1(N_{11}+C_2H_2) & C_2N_{21} \\ C_1N_{12} & C_2(N_{22}+C_1H_1) \end{bmatrix} \begin{bmatrix} Q_{1a}(s) \\ Q_{2a}(s) \end{bmatrix} \quad (2.8a)$$

$$d'(s) = d(s) + C_1N_{11}H_1 + C_2N_{22}H_2 + C_1C_2H_1H_2 \quad (2.8b)$$

Para um dado conjunto de valores dos parâmetros do controlador, o sistema pode agora ser estabilizado (Fig. 2.5), e inclusive, como já foi investigado [2.27], apresentar uma resposta algo semelhante à de um sistema linear. Contudo, somos confrontados com a selecção de 5n parâmetros interdependentes, sem dispormos de um critério sistemático para a sua selecção! Além disso, um conjunto de valores desses parâmetros, encontrado satisfactorio para uma dada região do espaço operacional, pode mostrar-se inadequado noutra zona, tal como se verifica, por vezes, com os actuais controladores lineares implementados a nível industrial, resultando o aparecimento de vibrações em algumas regiões de funcionamento.

Torna-se pois necessária a implementação de uma estrutura de controlo, que permita um ajuste simples e sistemático dos parâmetros do controlador, bem como um desacoplamento das saídas para qualquer região de funcionamento.

### 2.3. COMPENSAÇÃO DO ROBOT MANIPULADOR

Devido à variação, com  $q_{10}$  e  $q_{20}$ , dos pólos e zeros das funções de transferência de um robot manipulador, torna-se necessário encontrar uma compensação apropriada.

Esta compensação deve ser adaptativa, isto é, deve ajustar-

-se às variações da dinâmica do sistema. Além disso, pode conceber-se uma estrutura inerente ao próprio sistema mecânico, situação que passamos a denominar de "compensação através de dispositivos mecânicos", ou uma compensação baseada num algoritmo residente no computador de controlo, que por oposição chamaremos de "compensação por programação".

Começaremos por estudar a primeira situação, já que corresponde a uma prática frequente a nível industrial, e com base nas conclusões daí decorrentes é seguidamente introduzida a estratégia alternativa.

### 2.3.1. Compensação através de dispositivos mecânicos

Com vista à obtenção de um melhor desempenho dinâmico, são actualmente usadas estruturas mecânicas quer à base de contrapesos quer em paralelogramo [2.28]. Devido à sua simplicidade, é em seguida considerada a primeira situação.

O uso de contrapesos no nosso protótipo, dá origem à estrutura representada na figura 2.6. Usando mais uma vez o

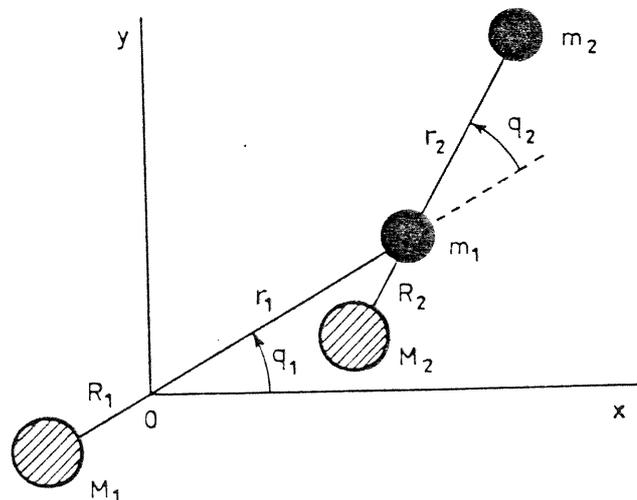


Fig. 2.6 Robot 2R com contrapesos.

Lagrangeano (2.1), as equações dinâmicas podem ser apresentadas matricialmente na forma:

$$J^*(q) = \begin{bmatrix} m_1 r_1^2 + M_1 R_1^2 + (m_2 + M_2) r_1^2 + m_2 r_2^2 & m_2 r_2^2 + M_2 R_2^2 + (m_2 r_2 - M_2 R_2) r_1 C_2 \\ + M_2 R_2^2 + 2(m_2 r_2 - M_2 R_2) r_1 C_2 + J_1^* & \\ m_2 r_2^2 + M_2 R_2^2 + (m_2 r_2 - M_2 R_2) r_1 C_2 & m_2 r_2^2 + M_2 R_2^2 + J_2^* \end{bmatrix} \quad (2.9a)$$

$$C^*(q, \dot{q}) = \begin{bmatrix} -2(m_2 r_2 - M_2 R_2) r_1 S_2 \dot{q}_1 \dot{q}_2 - (m_2 r_2 - M_2 R_2) r_1 S_2 \dot{q}_2^2 \\ (m_2 r_2 - M_2 R_2) r_1 S_2 \dot{q}_1^2 \end{bmatrix} \quad (2.9b)$$

$$G^*(q) = \begin{bmatrix} g[(m_1 r_1 - M_1 R_1) C_1 + (m_2 r_2 - M_2 R_2) C_{12} + (m_2 + M_2) r_1 C_1] \\ g(m_2 r_2 - M_2 R_2) C_{12} \end{bmatrix} \quad (2.9c)$$

Se se verificarem as condições

$$m_1 r_1 = M_1 R_1, m_2 r_2 = M_2 R_2 \quad (2.10)$$

então as expressões apresentadas anteriormente vêm consideravelmente simplificadas

$$J^*(q) = \begin{bmatrix} m_1 r_1^2 + M_1 R_1^2 + (m_2 + M_2) r_1^2 + m_2 r_2^2 + M_2 R_2^2 + J_1^* & m_2 r_2^2 + M_2 R_2^2 \\ m_2 r_2^2 + M_2 R_2^2 & m_2 r_2^2 + M_2 R_2^2 + J_2^* \end{bmatrix} \quad (2.11a)$$

$$C^*(q, \dot{q}) = [0, 0]^T \quad (2.11b)$$

$$G^*(q) = [g r_1 (m_2 + M_2) C_1, 0]^T \quad (2.11c)$$

Comparando as equações (2.3) e (2.11), verificamos que:

- Os termos Coriolis/centrípetos foram completamente eliminados
- Houve um cancelamento parcial dos termos gravitacionais
- Os termos inerciais dependentes da configuração, desapareceram, todavia surgiram novos termos inerciais constantes.

Repetindo o esquema de linearização e conversão para uma descrição na variável complexa  $s$ , o robot 2R com contrapesos, em termos de estabilidade é modelizado pela equação:

$$\begin{bmatrix} T_1^*(s) \\ T_2^*(s) \end{bmatrix} = \left( \begin{bmatrix} m_1 r_1^2 + (m_2 + M_2) r_1^2 + m_2 r_2^2 + & m_2 r_2^2 + M_2 R_2^2 \\ + M_1 R_1^2 + M_2 R_2^2 + J_1^* & \\ m_2 r_2^2 + M_2 R_2^2 & m_2 r_2^2 + M_2 R_2^2 + J_2^* \end{bmatrix} s^2 - \right. \\ \left. -g \begin{bmatrix} r_1 (m_2 + M_2) C_{10} \\ 0 \end{bmatrix} \right) \begin{bmatrix} Q_1(s) \\ Q_2(s) \end{bmatrix} \quad (2.12)$$

Facilmente se conclui sobre uma melhoria da estabilidade; contudo, ainda ocorrem situações onde a função de transferência apresenta pólos no semiplano direito. Este facto deve-se ao cancelamento incompleto dos termos gravitacionais da dinâmica do manipulador. Além disso, os novos termos inerciais que afectam a expressão (2.11), fazem diminuir a largura de banda e, consequentemente, tornam mais lento este sistema.

Uma compensação mais eficiente deve apresentar as seguintes propriedades:

- a) Eliminar completamente os termos gravitacionais
- b) Fazer desaparecer todos os termos inerciais que dependam da configuração
- c) Se possível, fazer decrescer os termos inerciais constantes, de modo a aumentar a largura de banda

É evidente que a metodologia de compensação estudada, baseada numa estrutura física, tem fortes limitações. Estas restrições, são contudo evitáveis, optando-se por uma implementação por programação, conforme se apresenta no ponto 2.3.2.

### 2.3.2. Compensação por programação

O algoritmo de compensação pode ser derivado a partir do caso estudado anteriormente. Sob o ponto de vista formal, a compensação através de dispositivos mecânicos corresponde às equações:

$$T=T^*+T_c \quad (2.13a)$$

$$T=J(q)\ddot{q}+C(q,\dot{q})+G(q) \quad (2.13b)$$

$$T^*=J^*(q)\ddot{q}+C^*(q,\dot{q})+G^*(q) \quad (2.13c)$$

$$T_c=J_c(q)\ddot{q}+C_c(q,\dot{q})+G_c(q) \quad (2.13d)$$

O binário  $T$ , anterior ao procedimento de compensação, é decomposto na soma de dois termos:

- O binário  $T^*$  requerido aos actuadores, após a compensação
- O binário  $T_c$ , correspondente à compensação por contrapesos

O respectivo diagrama de blocos (Fig. 2.7), mostra claramente que para o robot manipulador ter a capacidade de seguir instantâneamente uma dada trajectória  $\ddot{q}(t)$  (i.e. com largura de banda infinita), e sem exigir nenhum binário extra  $T^*$  (i.e.  $T^*=0$ ), então é necessário que:

$$J_c(q)=J(q) \quad (2.14a)$$

$$C_c(q,\dot{q})=C(q,\dot{q}) \quad (2.14b)$$

$$G_c(q)=G(q) \quad (2.14c)$$

Estas condições são complexas e elucidativas relativamente às dificuldades em conseguir uma compensação perfeita através de dispositivos mecânicos. A estratégia alternativa - compensação por programação - não tem esse tipo de limitações, pelo que está naturalmente vocacionada para uma implementação num controlador

digital.

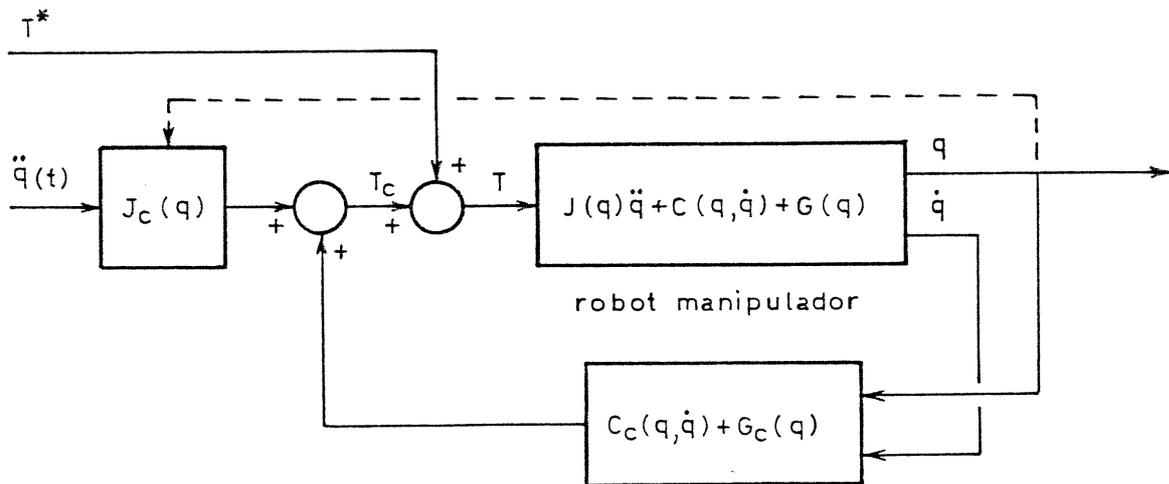


Fig. 2.7 Diagrama de blocos correspondente ao robot com o sistema de compensação por contrapesos.

Deve desde já notar-se que as condições anteriores têm significados diversos quando interpretados à luz do estudo efectuado:

- As condições (2.14a) e (2.14b) providenciam uma compensação dos termos de primeira ordem. A compensação dos termos inerciais (2.14a), permite a obtenção de funções de transferência com um ganho constante (i.e. independente da posição) às "altas frequências"; a condição complementar (2.14c) (compensação dos termos gravitacionais), faz com que as funções de transferência tenham todos os pólos na origem do plano  $s$ .

- A condição (2.14b) realiza uma compensação dos termos de segunda ordem. Apesar de não terem influência em termos de estabilidade, a sua não observância corresponde à existência de



de "desenho directo" (direct design), será providenciar a realimentação de posição e velocidade, bem como um ganho na malha directa, resultando a acção de controlo  $u(t)$ :

$$T^*=0 \quad (2.15a)$$

$$\ddot{q}(t)=u(t) \quad (2.15b)$$

$$u(t)=\Lambda q_d(t)-[K_1\dot{q}(t)+K_2q(t)] \quad (2.15c)$$

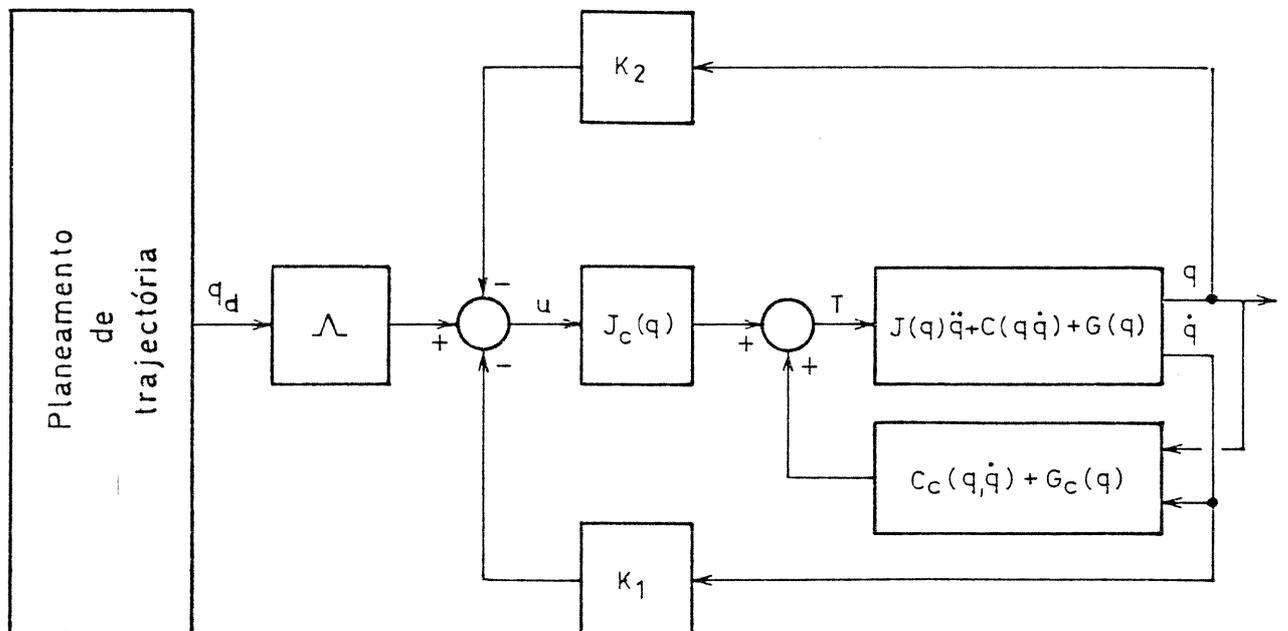


Fig. 2.9 Estrutura de controlo não-linear do tipo "desenho directo" (direct design).

Para o caso de matrizes diagonais constantes (i.e.  $\Lambda = \text{diag}(\lambda_i)$ ,  $K_1 = \text{diag}(K_{1i})$  e  $K_2 = \text{diag}(K_{2i})$  com  $i=1, \dots, n$ ), então obtêm-se  $n$  funções de transferência, desacopladas, da forma:

$$Q_i(s)/Q_{d_i}(s) = \lambda_i / (s^2 + K_{1i}s + K_{2i}) \quad , \quad i=1, \dots, n \quad (2.16)$$

onde os valores de  $\lambda_1$ ,  $K_{11}$  e  $K_{21}$  são escolhidos de modo a que o sistema permaneça estável e a satisfazer o desempenho desejado (naturalmente que para se ter ganho unitário às baixas frequências deverá ser  $\lambda_1 = K_{21}$ ).

É contudo possível obter uma estrutura ainda mais eficiente. Nos modernos sistemas de controlo de robots, normalmente existe um bloco de planeamento de trajectória, e como tal estão acessíveis, com precisão, os valores desejados para a posição, velocidade e aceleração (i.e.  $q_d(t)$ ,  $\dot{q}_d(t)$  e  $\ddot{q}_d(t)$ ).

Podemos tirar partido deste factor com vista a melhorar o controlador anterior, através do conhecido "método do binário calculado" (computed torque method) [2.1, 2.30], representado na Fig. 2.10 e descrito pelas expressões:

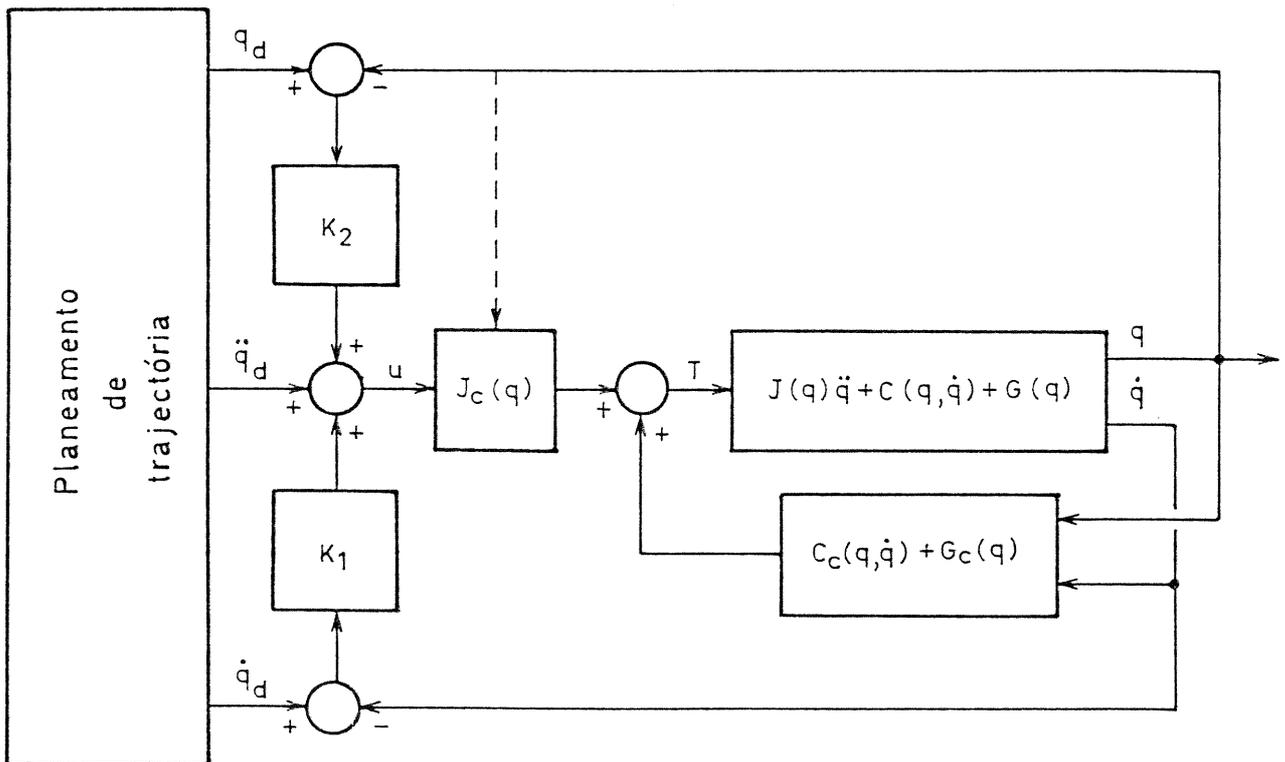


Fig. 2.10 Estrutura de controlo não-linear baseada no método do binário calculado.

$$T^*=0 \quad (2.17a)$$

$$\ddot{q}(t)=u(t) \quad (2.17b)$$

$$u(t)=\ddot{q}_a(t)+K_1[\dot{q}_a(t)-\dot{q}(t)]+K_2[q_a(t)-q(t)] \quad (2.17c)$$

Agora as funções de transferência vêm:

$$Q_i(s)/Q_{di}(s)=(s^2+K_{1i}s+K_{2i})/(s^2+K_{1i}s+K_{2i})=1, \quad i=1, \dots, n \quad (2.18)$$

ou seja, este método está bem adaptado para aplicações onde seja necessário um seguimento preciso e instantâneo da trajectória de referência.

#### 2.4. DISCUSSÃO E CONCLUSÕES

Neste capítulo foi estudada a dinâmica de robots manipuladores. Apesar da sua complexidade procurou-se obter uma descrição, simples e clara, no plano complexo.

A partir das conclusões obtidas, resultaram naturais todos os passos com vista à obtenção de um controlador eficiente.

O controlador obtido foi já proposto anteriormente por outros investigadores; contudo agora dispomos da experiência e intuição possíveis com uma descrição na variável  $s$ . Este facto deve ser realçado, e dele será tirado proveito no próximo capítulo, pois uma das maiores limitações nos controladores não-lineares sugeridos pelos diversos investigadores consiste na falta de ferramentas que permitam analisar os fenómenos dinâmicos envolvidos. No capítulo 3, as limitações do controlador não-linear pelo método do binário calculado, são examinadas e conseqüentemente é desenvolvido um novo e mais eficiente controlador.

Também foram estudados dois tipos de implementação das

compensações necessárias, tendo-se verificado que a realização por programação permitia uma muito maior liberdade. Deve ser notado, que esta estratégia tem contudo um inconveniente; ao invés da compensação através de dispositivos mecânicos, ambos os binários  $T^*$  e  $T_c$  têm que ser providenciados pelos actuadores. Daí podem resultar maiores exigências e eventual saturação dos motores de accionamento.

Ambas os sistemas de controlo apresentados isto é, as estruturas descritas pelas expressões (2.15) e (2.17), pressupõem  $T^*=0$  e  $\ddot{q}(t)=u(t)$ . Isto significa que o cálculo (correcto) de  $T^*$  corresponderia a saber quais os erros de modelização o que, não sendo possível, inviabiliza o uso deste sinal. Nesta ordem de ideias a acção de controlo deverá basear-se no outro sinal de entrada à nossa disposição, ou seja fazer  $\ddot{q}(t)=u(t)$ , o que na realidade foi, implicitamente, assumido. Contudo, parece ser interessante pôr em causa este pressuposto e considerar uma estrutura de controlo que tire proveito de ambas as entradas. Assim, o cálculo de  $T^*$ , ainda que aproximado, constituiria uma acção de antecipação (feedforward), permanecendo  $\ddot{q}(t)$  com as funções de "tratamento" deste sinal. No próximo capítulo, verificaremos como atingir esta estrutura, se bem que a partir de considerações, à partida, algo diferentes!

## REFERÊNCIAS

- [2.1] J. Y. S. Luh, M. W. Walker, R. P. C. Paul  
Resolved-Acceleration Control of Mechanical Manipulators  
IEEE Trans. Automat. Contr. vol. 25, pp. 468-474, June, 1980.
- [2.2] E. Freund  
Fast Nonlinear Control with Arbitrary Pole-Placement for Industrial Robots and Manipulators  
The Int. J. Robotics Research, vol. 1, pp. 65-78, Spring, 1982.
- [2.3] M. Sahba, D. Q. Mayne  
Computer-Aided Design of Nonlinear Controllers for Torque Controlled Robot Arms  
IEE Proc., vol. 131, Pt. D, pg. 8-14, Jan., 1984
- [2.4] Philippe Coiffet  
Robot Technology: Modelling and Control  
Kogan Page, London, 1983.
- [2.5] H. W. Stone, C. P. Neuman  
Dynamic Modelling of a Three-Degrees-of-Freedom Robotic Manipulator  
IEEE Trans. Syst., Man, Cybern., vol. SMC-14, pg. 643-654, July/Aug., 1984.
- [2.6] S. J. Williams, K. Glover  
Robot Dynamic Control Instabilities  
Proc. American Control Conf., pg. 390-392, 1985.
- [2.7] S. J. Williams, G. A. Bundell  
Detailed Dynamic Simulation and Interactive Verification of an Industrial Robot  
Proc. IEE Conf. Control '85, pg. 28-33, 1985.
- [2.8] S. J. Williams  
Frequency Response Multivariable Control of Robotic Manipulators  
IEE Proc., vol. 132, Pt. D, pg. 144-150, July, 1985.
- [2.9] M. C. Good, L. M. Sweet, K. L. Strobel  
Dynamic Models for Control Systems Design of Integrated Robot and Drive System  
ASME J. Dynamic Syst., Meas., Contr., vol. 107, pg. 53-59, March, 1985.
- [2.10] T. M. Stepien, L. M. Sweet, M. C. Good, M. Tomizuka  
Control of Tool/Workpiece Contact Force with Application to Robotic Deburring  
IEEE Int. Conf. Robotics and Automation, pg. 670-679, St. Louis, Missouri, USA, 1985.

- [2.11] Larry M. Sweet, Malcolm G. Good  
Redefinition of the Robot Motion-Control Problem  
IEEE Control System Magazine, pg. 18-24, Aug., 1985.
- [2.12] Thomas M. Stepien, Larry M. Sweet, Malcolm C. Good,  
Masayoshi Tomizuka  
Control of Tool/Workpiece Contact Force with  
Application to Robotic Deburring  
IEEE J. Robotics and Automation, vol. RA-3, pg. 7-18,  
Feb., 1987.
- [2.13] H. Seraji, M. Jamshidi, Y. K. Tim, M. Shahinpoor  
Linear Multivariable Control of Two-Link Robots  
J. of Robotic Systems, vol. 3, pg. 349-365, Winter,  
1986.
- [2.14] H. Seraji  
An Approach to Multivariable Control of Manipulators  
ASME J. Dynamic Syst., Meas., Contr., vol. 109,  
pg. 146-154, June, 1987.
- [2.15] Murray R. Spiegel  
Theoretical Mechanics with an Introduction of  
Lagrange's Equations and Hamiltonian Theory  
McGraw-Hill Inc., 1967.
- [2.16] Emil Jan Konopinski  
Classical Descriptions of Motion  
W. E. Freeman and Company, San Francisco, 1969.
- [2.17] Michel Geradin  
Elements de Robotique: Aspects Mechaniques  
Universite de Liège, 1985.
- [2.18] R. P. Paul  
Robot Manipulators: Mathematics, Programming and  
Control  
The MIT Press, 1981.
- [2.19] M. Brady et al.  
Robot Motion: Planning and Control  
The MIT Press, 1982.
- [2.20] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Dynamics and Control of Robot Manipulators  
20<sup>o</sup> DINAME - Segundo Simposio Sobre Sistemas Dinâmicos  
da Mecânica. Problemas Dinâmicos em Máquinas e  
Estruturas, Campos de Jordão, São Paulo, Brasil, 1988.
- [2.21] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Robot Manipulators Systems: Analysis and Control  
3<sup>rd</sup> Int. Symp. on Systems Analysis and Simulation,

- [2.22] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Robot Manipulators Systems: Analysis and Control  
Int. J. of Systems Analysis-Modelling-Simulation  
1989.
- [2.23] Ostman I., Holmqvist V., Allared C.-A  
Pendulum Robot  
Asea Journal, vol. 58, pg. 12-17, 3-4, 1986.
- [2.24] MultiCraft a.s  
The "MultiCraft 560" Industrial Robot.
- [2.25] B. Dwolatzky, G. S. Thornton  
The GEC Tetrabot - A Serial-Parallel Topology Robot:  
Control Design Aspects  
IEE Control '88, Oxford, UK, 1988.
- [2.26] H. Asada, T. Kanade, I. Takeyama  
Control of a Direct-Drive Arm  
ASME J. Dynamic Syst., Meas., Contr., vol. 105,  
pg. 136-142, Sept., 1983.
- [2.27] D. F. Golla, S. C. Garg, P. C. Hughes  
Linear State Feedback Control of Manipulators  
Mech. Machine Theory, vol. 16, pg. 93-103, 1981.
- [2.28] Michel Geradin  
Mechanical Aspects of Robotics  
Seminário, IST, Lisboa, 1985.
- [2.29] V. D. Tourassis, C. P. Neuman  
Robust Nonlinear Feedback Control for Robotic  
Manipulators  
IEE Proc., vol. 132, Pt. D, pg. 134-143, July, 1985.
- [2.30] Charles P. Neuman, Vassilios D. Tourassis  
Robust Discrete Nonlinear Feedback Control for Robotic  
Manipulators  
J. Robotic Systems, vol. 4, pg. 115-143, Feb., 1987.

CAPÍTULO 3

CONTROLO EFICIENTE  
DE  
ROBOTS MANIPULADORES

### 3. CONTROLO EFICIENTE DE ROBOTS MANIPULADORES

#### 3.1. INTRODUÇÃO

No capítulo anterior foi estudada a dinâmica de robots manipuladores. A luz das conclusões obtidas foram extrapoladas algumas arquitecturas de controladores, das quais o conhecido método do binário calculado é aquele que melhor se adapta às normais exigências de funcionamento. Este controlador foi, no entanto, desenvolvido a partir de considerações ideais, nomeadamente estimação perfeita do modelo do robot manipulador e uso de um controlador digital com uma capacidade de cálculo e período de amostragem em tempo nulo. Evidentemente estas condições não se verificam na prática e, conseqüentemente, permanece o problema de saber até que ponto o sistema é robusto relativamente a desvios entre o modelo matemático de que se dispõe e a realidade existente, ou ainda, quais os melhoramentos possíveis e apropriados que urge introduzir.

Devido à natureza não-linear do problema, têm sido raros os esforços desenvolvidos neste sentido. Assim, é frequente na literatura científica o aparecimento de novos tipos de controladores, sem todavia existirem trabalhos que estabeleçam claramente quais as vantagens e inconvenientes dos já existentes.

Neste capítulo pretendem-se aprofundar certos aspectos do controlador referido. Algum trabalho nesta área foi já desenvolvido por Neuman [3.1-3.2], que demonstrou a possibilidade de se melhorar o desempenho global a partir de modificações na estrutura do sistema de controlo.

O estudo apresentado seguidamente pretende realizar um tratamento mais sistemático, e como tal proporcionar a aplicação

das ferramentas matemáticas usuais em engenharia de controlo. Nesta ordem de ideias, no parágrafo 3.2 essas ferramentas são aplicadas ao controlador pelo método de binário calculado, resultando assim evidentes as exigências postas a um novo e mais eficiente controlador. Consequentemente em 3.3 a correspondente arquitectura de controlo é desenvolvida, e em 3.4, o seu desempenho comparado com o método convencional. Por último, no parágrafo 3.5, os resultados obtidos são discutidos e sumariados.

### 3.2. ENQUADRAMENTO DO CONTROLADOR CONVENCIONAL

Conforme foi desenvolvido no capítulo 2, para um robot manipulador genericamente descrito pelo conjunto de equações diferenciais não lineares

$$T=J(q)\ddot{q}+C(q,\dot{q})+G(q) \quad (3.1)$$

o controlador pelo método do binário calculado (computed torque method) (Fig. 3.1), consistindo no algoritmo

$$\text{Commando em aceleração} \quad u(t)=\ddot{q}_a+K_1(\dot{q}_a-\dot{q})+K_2(q_a-q) \quad (3.2a)$$

$$\text{Binário calculado} \quad T(t)=J_c(q)u(t)+C_c(q,\dot{q})+G_c(q) \quad (3.2b)$$

$$\text{Erro} \quad e(t)=q_a(t)-q(t) \quad (3.2c)$$

$$\text{Equação de erro} \quad \ddot{e}+K_1\dot{e}+K_2e=W(t) \quad (3.2d)$$

$$\text{Vector perturbação} \quad W(t)=J_c(q)^{-1}\{[J(q)-J_c(q)]\ddot{q}+ \\ +[C(q,\dot{q})-C_c(q,\dot{q})]+[G(q)-G_c(q)]\} \quad (3.2d)$$

permite, a priori, melhor desempenho que os usuais controladores PID lineares. Em particular, se as condições

$$J_c(q)=J(q) \quad (3.3a)$$

$$C_c(q,\dot{q})=C(q,\dot{q}) \quad (3.3b)$$

$$G_c(q) = G(q) \quad (3.3c)$$

forem verificadas, então o vector  $W(t)$  é nulo, e o sistema comporta-se como sendo linear. Para além disso, é possível um desacoplamento entre saídas desde que  $K_1$  e  $K_2$  sejam matrizes diagonais.

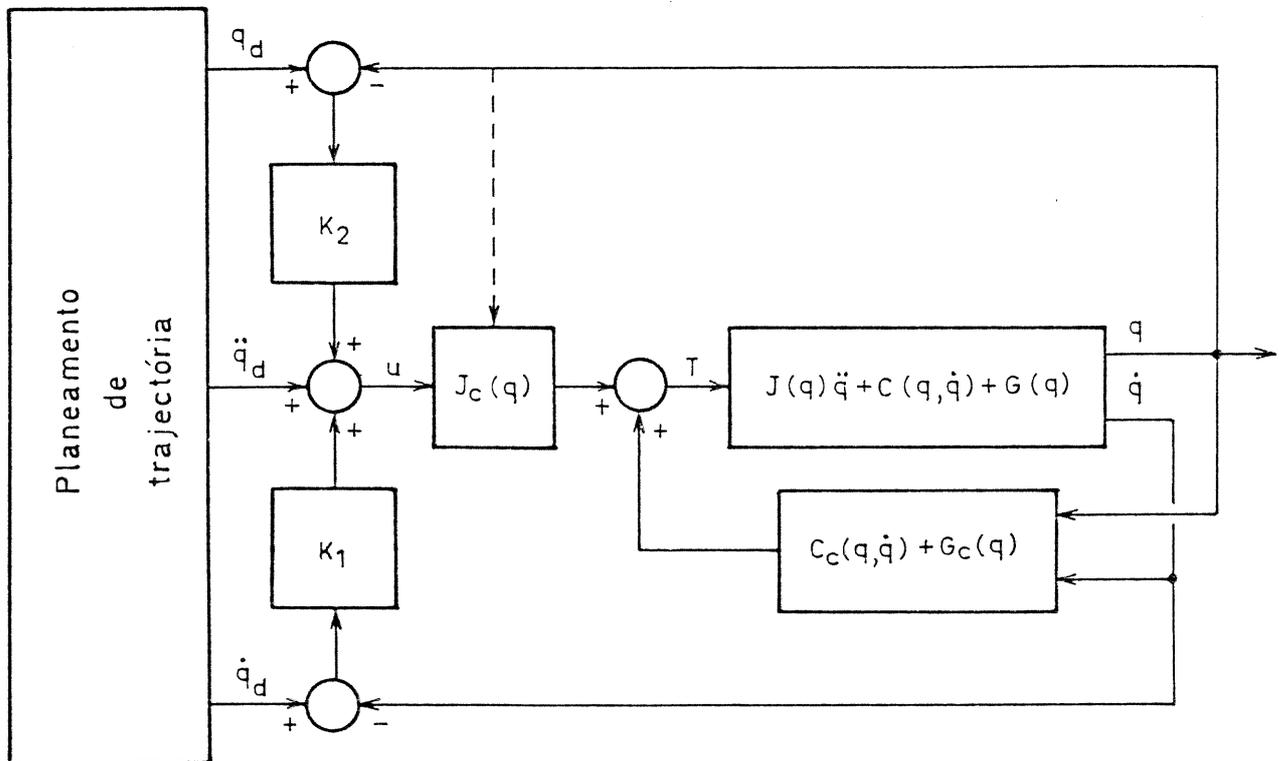


Fig. 3.1 Estrutura de controlo não-linear baseada no método do binário calculado.

Verificamos que este controlador está fortemente baseado no modelo (3.1), pelo que resultam naturais e imediatas dúvidas sobre o seu desempenho quando não são observadas as condições ideais (3.3). A descrição através da variável  $s$  conseguida no capítulo anterior possibilita essa resposta.

Para um modelo exacto o sistema pode ser descrito pela

função de transferência

$$Q_i(s)/Q_{d_i}(s) = (s^2 + K_{v_i}s + K_{p_i}) / (s^2 + K_{v_i}s + K_{p_i}) = 1, \quad i=1, \dots, n \quad (3.4)$$

No entanto, os factores não-ideais já referidos, motivam uma investigação mais profunda quanto à sua exactidão. Rearranjando as equações (3.2d) e (3.2e), obtem-se [3.4-3.7]:

$$J_c(q)(\ddot{e} + K_1\dot{e} + K_2e) = \delta T \quad (3.5)$$

onde

$$\delta T = [J(q)\ddot{q} + C(q, \dot{q}) + G(q)] - [J_c(q)\ddot{q} + C_c(q, \dot{q}) + G_c(q)] \quad (3.6)$$

é o erro de modelização. A equação (3.5) permite concluir que:

- O segundo membro da equação pode ser interpretado como o binário devido ao erro de modelização e o primeiro membro como o binário correctivo exigido ao controlador.

- O binário correctivo exigido ao controlador é providenciado pela malha  $J_c(q)$ ,  $K_1$ ,  $K_2$ .

- Quanto maior for a inexactidão do modelo usado, tanto maiores serão as exigências em termos de binário correctivo exigido ao controlador.

- Em sistemas de controlo discreto, com frequência de amostragem finita, não é possível uma correcção contínua; o carácter descontínuo do binário atribuído pelo controlador, pode implicar uma degradação do desempenho do sistema.

Estas conclusões apontam claramente para a dependência das propriedades dinâmicas do sistema relativamente à malha  $J_c(q)$ ,  $K_1$ ,  $K_2$ ; todavia, é também necessário um bom desempenho em regime permanente. Neste caso temos:

$$\dot{q}_a = 0 ; \ddot{q}_a = 0 ; \dot{q}_a = 0 ; \ddot{q}_a = 0 \quad (3.7)$$

resultando a equação

$$e_{RP} = [K_2 J_c(q)]^{-1} [G(q) - G_c(q)] \quad (3.8)$$

onde  $e_{RP}$  é o vector (n-dimensional) dos erros de posicionamento em regime permanente. Tal como seria de esperar,  $e_{RP}$  é determinado pelo erro de modelização dos termos gravitacionais e é inversamente proporcional ao ganho  $K_2$ .

Das expressões (3.5), (3.6) e (3.8) verificamos que os blocos constitutivos do controlador desempenham tarefas diferentes:

- Em termos dinâmicos (i.e. regime transitório e estabilidade) a malha de realimentação  $J_c(q)$ ,  $K_1$ ,  $K_2$  é de importância fundamental.

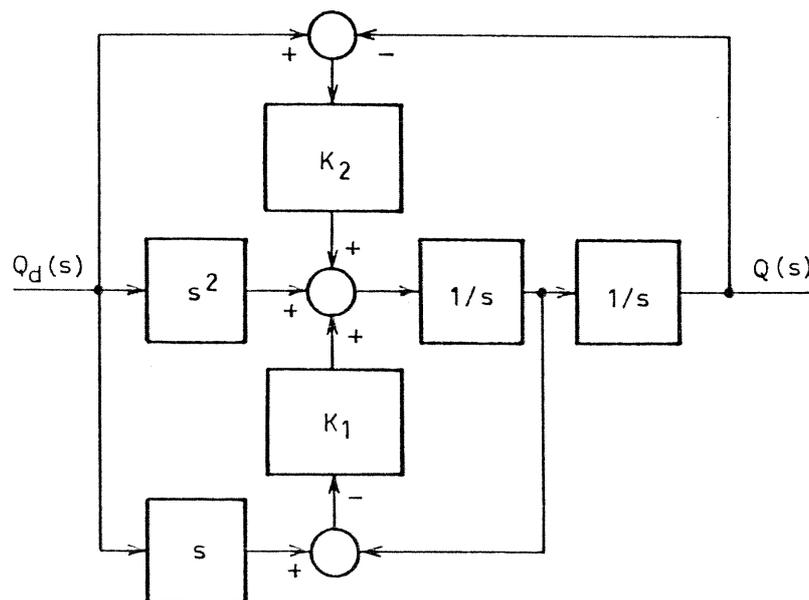


Fig. 3.2 Modelo linear para um controlador ideal baseado no método do binário calculado.

- Em regime permanente, o modelo gravitacional ( $G_c(q)$ ) e a matriz constante de ganho proporcional ( $K_z$ ) são os blocos principais.

Apesar de exactas estas expressões são complexas e pouco apropriadas como ferramenta de trabalho quer na análise quer na síntese do controlador. O uso consistente dos métodos tradicionais de teoria dos sistemas, implica algumas modificações no diagrama de blocos representativo do sistema ideal (Fig. 3.2). Assim, uma descrição mais exacta na variável  $s$ , deve entrar em linha de conta com:

- Um bloco de atraso ( $e^{-sT_1}$ ), correspondente ao tempo de cálculo ( $T_1$ ) do algoritmo.

- Um segundo atraso ( $e^{-sT_2}$ ), que descreve aproximadamente o bloco de amostragem e retenção (sample and hold). Este bloco é frequentemente modelizado pela expressão

$$H(s) = (1 - e^{-sh}) / (sh) \quad , \quad h\text{-período de amostragem} \quad (3.9)$$

Contudo, um cálculo elementar e o desenvolvimento em série de Taylor mostram que para "baixas frequências"  $H(s)$  pode ser descrito pela expressão

$$H(s) \approx e^{-sh/2} \quad ; \quad T_2 = h/2 \quad (3.10)$$

- Um ganho  $K \gg 1$ , presente na malha directa modelizando as imprecisões dos termos diagonais da matriz  $J_c(q)$ .

- As restantes inexactidões do modelo, seja nos termos inerciais não diagonais seja nos termos Coriolis/centrípetos e gravitacionais, são representados por uma perturbação  $P(s)$  que influencia a malha directa.

- Com vista a serem concebidas estruturas de controlo mais

genéricas, as matrizes constantes  $K_1$  e  $K_2$  são substituídas por funções de transferência  $G_1(s)$  e  $G_2(s)$ , respectivamente.

O diagrama (desacoplado) de blocos resultante está representado na Fig. 3.3, vindo

$$Q(s)/Q_d(s) = Ke^{-sT_d} \{1 + (1 - e^{-sT_d}) [sG_1(s) + G_2(s)] / D(s)\} \quad (3.10a)$$

$$Q(s)/P(s) = Ke^{-sT_d} / D(s) \quad (3.10b)$$

$$D(s) = s^2 + Ke^{-sT_d} [sG_1(s) + G_2(s)] \quad (3.10c)$$

$$T_d = T_1 + T_2 \quad (3.10d)$$

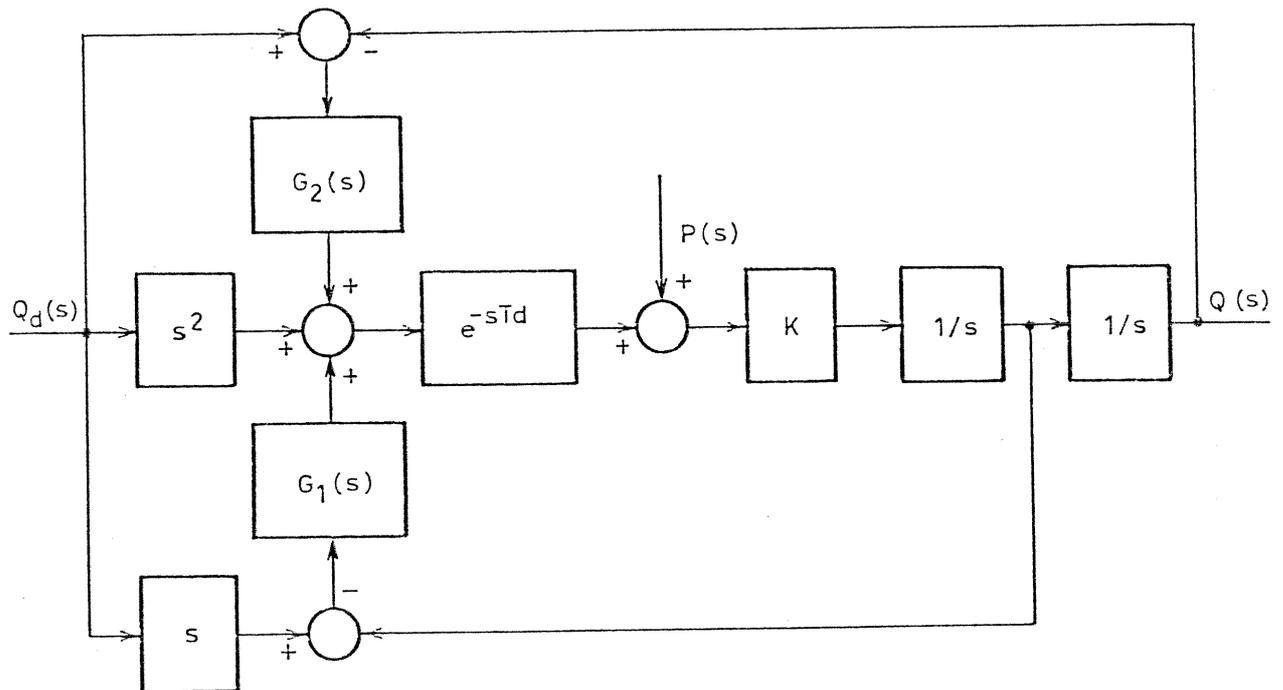


Fig. 3.3 Modelo linear para um controlador não-ideal baseado no método do binário calculado.

Este modelo, ainda que aproximado, possibilita conclusões semelhantes às expressões exactas (3.5), (3.6) e (3.8), nomeadamente:

- O tempo de atraso  $T_d = T_1 + T_2$  deve ser o menor possível
- O ganho  $K$  deve ser unitário
- Um controlador mais robusto implica a necessidade de algoritmos mais sofisticados para os blocos  $G_1(s)$  e  $G_2(s)$ .

### 3.3. MELHORAMENTO DA ARQUITECTURA DO CONTROLADOR

A partir das equações (3.10), e usando as habituais ferramentas de análise de sistemas, podemos agora introduzir novas e mais coerentes modificações na estrutura base do controlador. Deste modo o uso de técnicas padrão tais como:

- Uso de acção integral (i.e.  $G_2(s) = K_2 + K'_2/s$ ), para eliminar o erro em regime permanente  $e_{RP}$  (3.8).
- Acção derivativa (i.e.  $G_1(s) = K_1 + K'_1 s$ ) para melhorar a estabilidade.
- Análise da estabilidade a partir de (3.10c) através do diagrama de Nyquist.

Para além destes melhoramentos convencionais, pode conseguir-se uma redução do tempo de atraso  $T_d$  através do uso de vários e diferentes períodos de amostragem [3.8], consoante as malhas e blocos envolvidos. Assim:

- A malha  $J_c(q)$ ,  $G_1(s)$ ,  $G_2(s)$ , com fortes implicações na estabilidade e resposta transitória, é calculada a uma alta frequência de amostragem.

- A malha  $C_c(q, \dot{q}) + G_c(q)$ , com uma influência reduzida nesse desempenho, pode ser calculada a baixas frequências de amostragem.

Podemos pois afirmar que o "novo" controlador particiona o respectivo tempo de funcionamento de acordo com as necessidades manifestadas pelo sistema. Este aspecto é de grande importância, já que corresponde a um uso mais racional e eficiente dos recursos existentes. Uma frequência de amostragem uniforme em todos os blocos, representa um "desperdício" de potência de cálculo na malha  $C_c(q, \dot{q}) + G_c(q)$ , que será mais útil na malha  $J_c(q)$ ,  $G_1(s)$ ,  $G_2(s)$ .

Outra observação igualmente importante reside na carga computacional colocada por cada bloco. Constata-se que os blocos  $G_1(s)$  e  $G_2(s)$  são facilmente calculados,  $J_c(q)$  e  $G_c(q)$  representam uma carga média, e que  $C_c(q, \dot{q})$  é altamente complexo. Assim, na malha que requer uma maior frequência de amostragem,  $J_c(q)$  constitui o bloco que apresenta um maior peso computacional. Metodologias de cálculo, cada vez mais eficientes [3.9-3.13] (ver também Cap. 5), têm vindo a ser desenvolvidas e podem, evidentemente, ser aplicadas no caso de  $J_c(q)$ ; todavia, um sistema computacional alternativo surge naturalmente quando se considera o uso de memória. Este expediente para aliviar o peso computacional foi já estudado tendo em vista a dinâmica global do robot manipulador [3.14-3.23]; não obstante, a sua adaptação ao caso dos termos inerciais é agora muito menos exigente. Esta estratégia pode assim ser vista como uma acção suplementar a nível de gestão dos recursos existentes. Efectivamente, algumas centenas de Kbytes estão perfeitamente ao alcance dos microcomputadores actuais, tornando-se pois imperativo tirar partido dessas capacidades, nas arquitecturas de controlo em desenvolvimento. Um microcomputador dispõe não só de uma dada capacidade de processa-

mento mas também da possibilidade de endereçar certa quantidade de memória. Muitas vezes só a primeira característica é considerada, explicitamente, para a avaliação das necessidades de processamento de um dado algoritmo de controlo, o que constitui um lapso que tem vindo a ser eliminado em certos centros de desenvolvimento [3.25].

Além dos aspectos referidos deve ainda notar-se que as necessidades em memória, impostas pela tabulação dos termos inerciais, podem ser optimizadas, atendendo a que:

- Em robots dispendo de juntas rotativas, esta tabela pode ser reduzida a metade já que os termos inerciais são idênticos para configurações simétricas de posicionamento.

- O número de coeficientes de  $J_c(q)$  tabelados, pode ser reduzido de  $n^2$  para  $n(n+1)/2$ , devido a tratar-se de uma matriz simétrica.

- A dimensão da matriz em memória é somente de ordem  $n-1$  (e não de ordem  $n$ ), pois os termos inerciais não dependem da configuração do primeiro elo da estrutura do robot manipulador.

- O uso de métodos de compressão de informação, tais como o modelo de memória associativa CMAC [3.14, 3.15, 3.20, 3.24], pode ser também considerado.

- E, obviamente, podem usar-se interpolações.

Finalmente, uma terceira medida de gestão dos recursos estruturais pode ser implementada através de uma acção predictiva [3.26-3.30]. Como é de esperar um bom acompanhamento da trajectória planeada, então os termos Coriolis/centrípetos e os termos gravitacionais, podem ser calculados previamente, estimando-se as futuros valores das posições e velocidades como sendo semelhantes aos planeados, ficando o cálculo (aproximado) dos valores inter-

Planeamento de trajectória

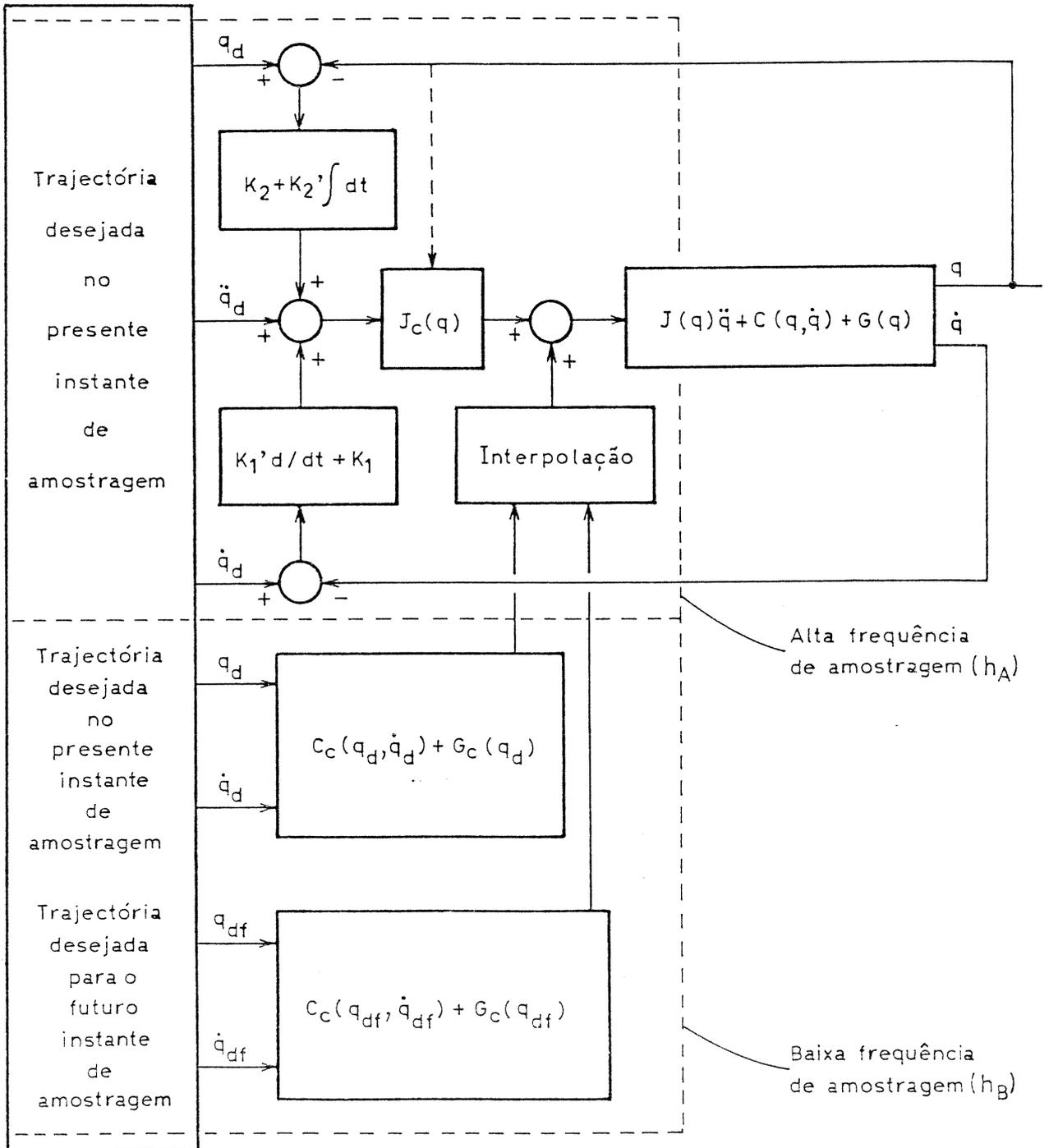


Fig. 3.4 O novo controlador não-linear.

médios, entre amostragens, reduzido a simples interpolações.

A figura 3.4 mostra a estrutura do controlador decorrente das acções empreendidas. Note-se que este controlador tem algumas semelhanças com o esquema de compensação por antecipação (feedforward) da dinâmica do robot manipulador [3.31]. Esta observação não é surpreendente, já que um estudo recente [3.32] mostra as fortes relações entre o método de compensação por antecipação e o método do binário calculado.

### 3.4. IMPLEMENTAÇÃO DA NOVA ARQUITECTURA DE CONTROLO

Passemos agora à implementação do controlador desenvolvido e comparação com o método convencional, para o robot 2R com os parâmetros numéricos apresentados na Tabela 3.1.

Tabela 3.1 Valores numéricos dos parâmetros do robot 2R

$r_1=1$ m	$r_2=0.8$ m	$J_1=5$ Kgm <sup>2</sup>	$J_2=5$ Kgm <sup>2</sup>
$m_1=0.5$ Kg	$m_0=0.25$ Kg	$m_L=6$ Kg	$m_2=m_0+m_L$

Partindo das expressões do erro de acompanhamento de trajectória

$$\ddot{e}+K_1\dot{e}+K_2e=0 \quad (3.11a)$$

$$(I+K_1')\ddot{e}+K_1\dot{e}+K_2e+K_2'\int e dt=0 \quad (3.11b)$$

para os casos convencional (3.11a) e em desenvolvimento (3.11b), podem ser calculados os parâmetros dos controladores (Tabela 3.2), de modo a obter-se uma resposta de segunda ordem com amortecimento crítico.

Tabela 3.2 Parâmetros do controlador

tipo de controlador	método do binário calculado	novos controladores
equação de projecto	$\ddot{e} + K_1 \dot{e} + K_2 e = 0$	$(1 + K_1') \ddot{e} + K_1 \dot{e} + K_2 e + K_2' \int e dt = 0$
localização dos pólos no domínio s	$s_{1,2} = -W_n$	$s_{1,2} = -W_n ; s_3 = -a W_n$
coeficientes diagonais	$K_1 = 2W_n ; K_2 = W_n^2$	$K_1 = (1 + K_1')(2W_n + a)$ $K_2 = (1 + K_1')(W_n + 2a)W_n$ $K_2' = (1 + K_1')aW_n^2$
valores numéricos	$W_n = 4$	$W_n = 4 ; a = 10 ; K_1' = 0.1$
período de amostragem (seg)	0.005	$h_A = 0.005, h_B = 0.16$
margem de fase	$72^\circ$	$60^\circ$

Conforme foi já referido anteriormente, o tempo de atraso  $T_a$  pode gerar um comportamento instável do controlador. Assim, recorreu-se ao diagrama de Nyquist para verificar as características dinâmicas de ambos os controladores. Com essa ferramenta de análise obtiveram-se margens de fase de  $72^\circ$  e  $60^\circ$  respectivamente, pelo que ambos os controladores podem ser considerados satisfatórios sob o ponto de vista de estabilidade. Por outro lado, foi também possível encontrar a relação entre  $T_a$  e  $W_n$  (valor do pólo duplo) como  $T_a \leq 0.651/W_n$  para o algoritmo convencional e  $T_a \leq 0.119/W_n$  para a nova arquitectura de controlo.

Tendo em vista uma comparação dos desempenhos dos dois sistemas de controlo em termos de seguimento de trajectória, foi considerada uma trajectória teste (Fig. 3.5) bem como uma medida

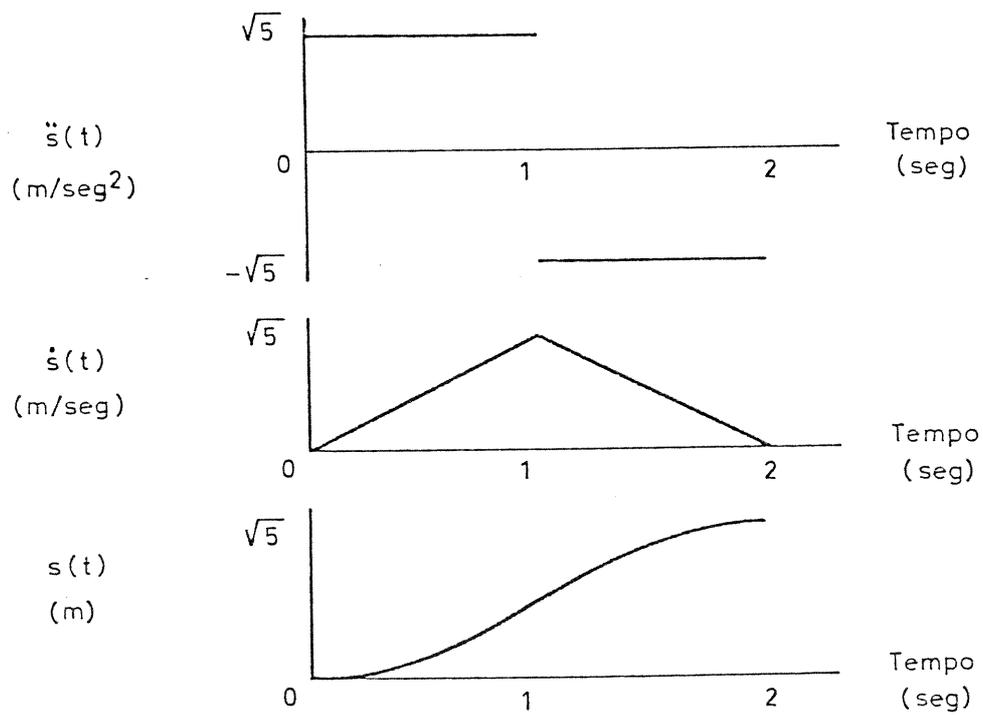
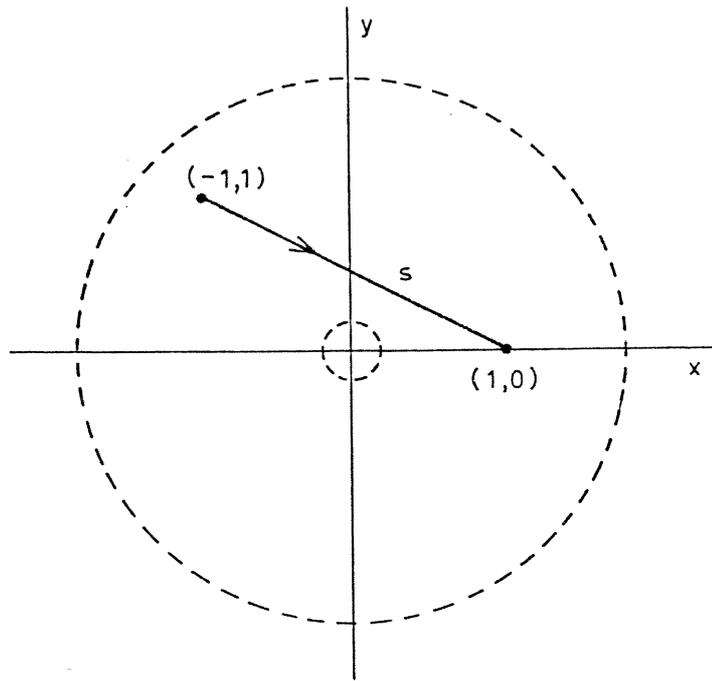


Fig. 3.5 Trajetória de teste.

do erro de acompanhamento de acordo com a expressão

$$e_p = [(x_d - x)^2 + (y_d - y)^2]^{1/2} \quad (3.12)$$

A figura 3.6 mostra a evolução deste índice para as situações de sobre-estimação ( $\hat{m}_L = 12$  Kg), estimação correcta ( $\hat{m}_L = m_L = 6$  Kg) e sub-estimação ( $\hat{m}_L = 0$  Kg) da carga real do robot manipulador ( $m_L = 6$  Kg). Tal como foi anteriormente observado por outros investigadores [3.1, 3.2, 3.33] a situação de sobre-estimação da carga acarreta menores erros que a situação oposta de sub-estimação.

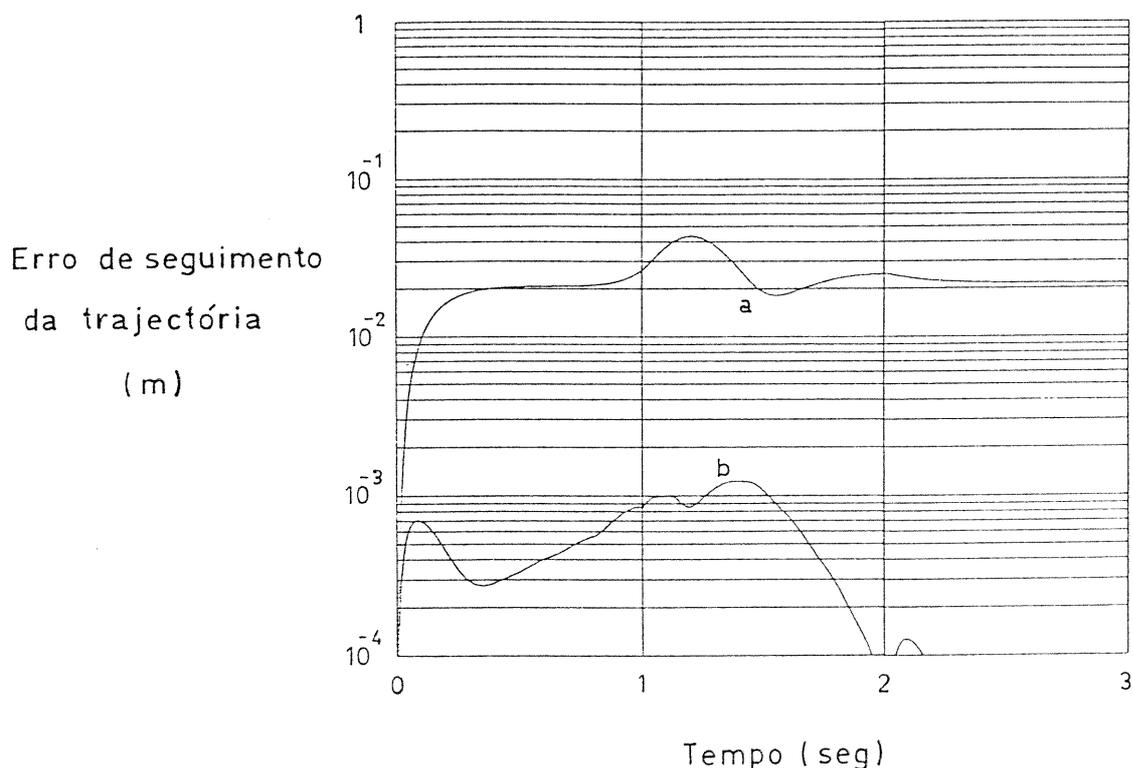
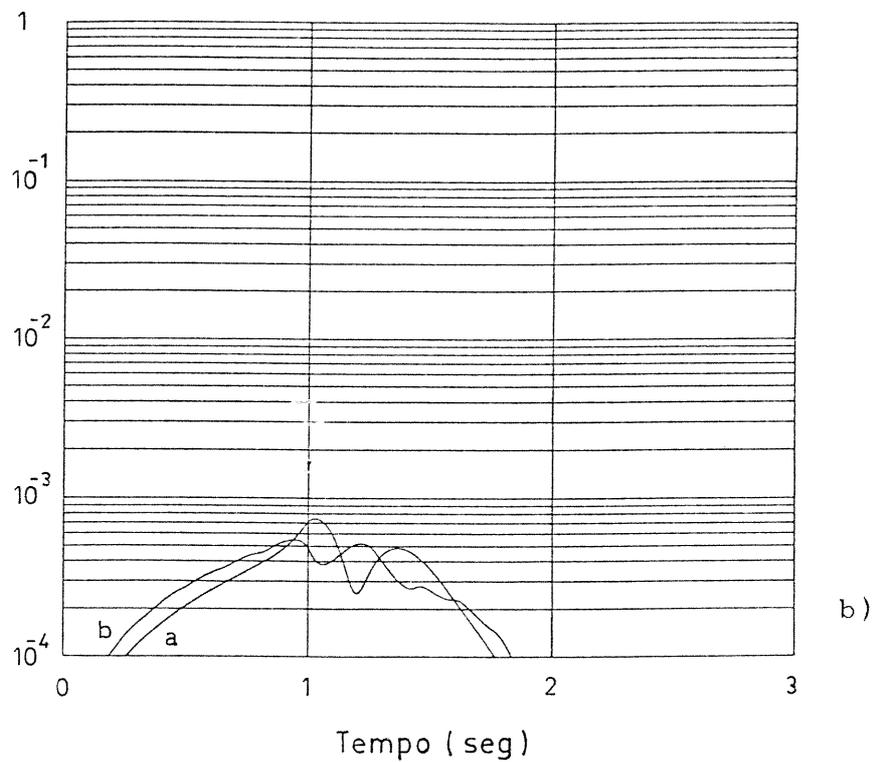


Fig. 3.6 Erro de seguimento da trajectória teste.  
 a - controlador convencional ( $h=0.005$  seg)  
 b - novo controlador  
 ( $h_A=0.005$  seg,  $h_B=0.16$  seg,  $T_1=0.1$   $h_A$ )  
 a) Para uma sobreestimação da massa de carga ( $\hat{m}_L = 12$  Kg)

Erro de seguimento  
da trajectória  
(m)



Erro de seguimento  
da trajectória  
(m)

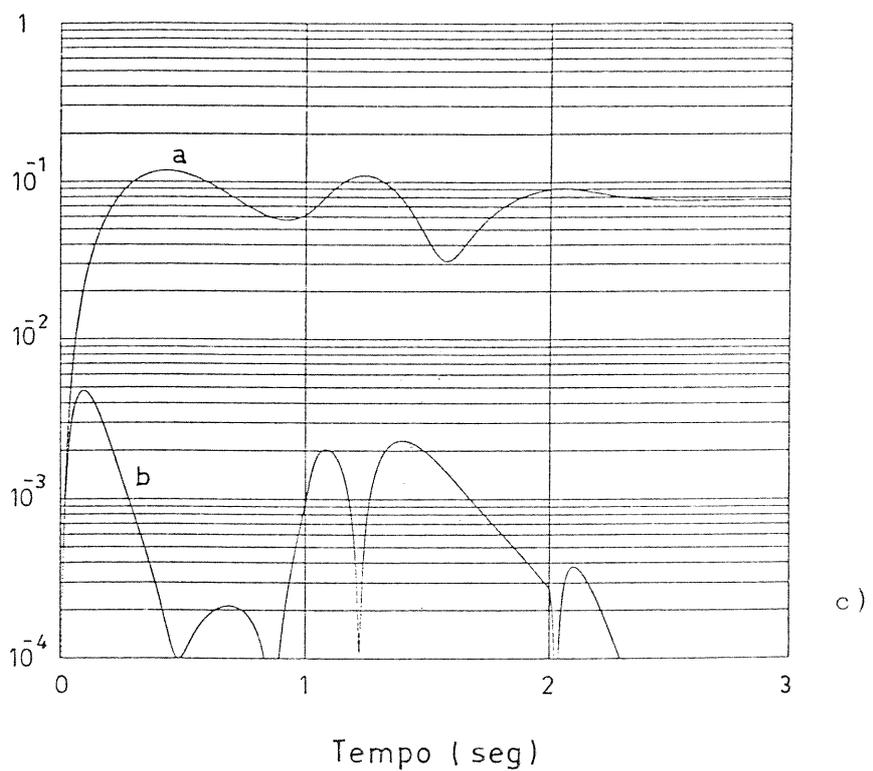


Fig. 3.6 Erro de seguimento da trajectória teste (continuação).  
 a - controlador convencional ( $h=0.005$  seg)  
 b - novo controlador  
 ( $h_A=0.005$  seg,  $h_B=0.16$  seg,  $T_I=0.1 h_A$ )  
 b) Para uma estimação exacta da massa de carga ( $\hat{m}_L=6$  Kg)  
 c) Para uma subestimação da massa de carga ( $\hat{m}_L=0$  Kg).

A nova arquitectura de controlo elimina também o erro em regime permanente e reduz o máximo erro de seguimento da trajectória teste. Para as simulações efectuadas o máximo erro de acompanhamento foi reduzido de um factor de 35.5, 1.4 e 24.7, respectivamente.

Deve ainda assinalar-se que a situação representada na figura 3.6b) confirma os pressupostos iniciais baseados nas equações (3.5) e (3.6). A estimação exacta do modelo do robot manipulador, favorece o funcionamento do controlador, mas não elimina totalmente os erros de acompanhamento de trajectória.

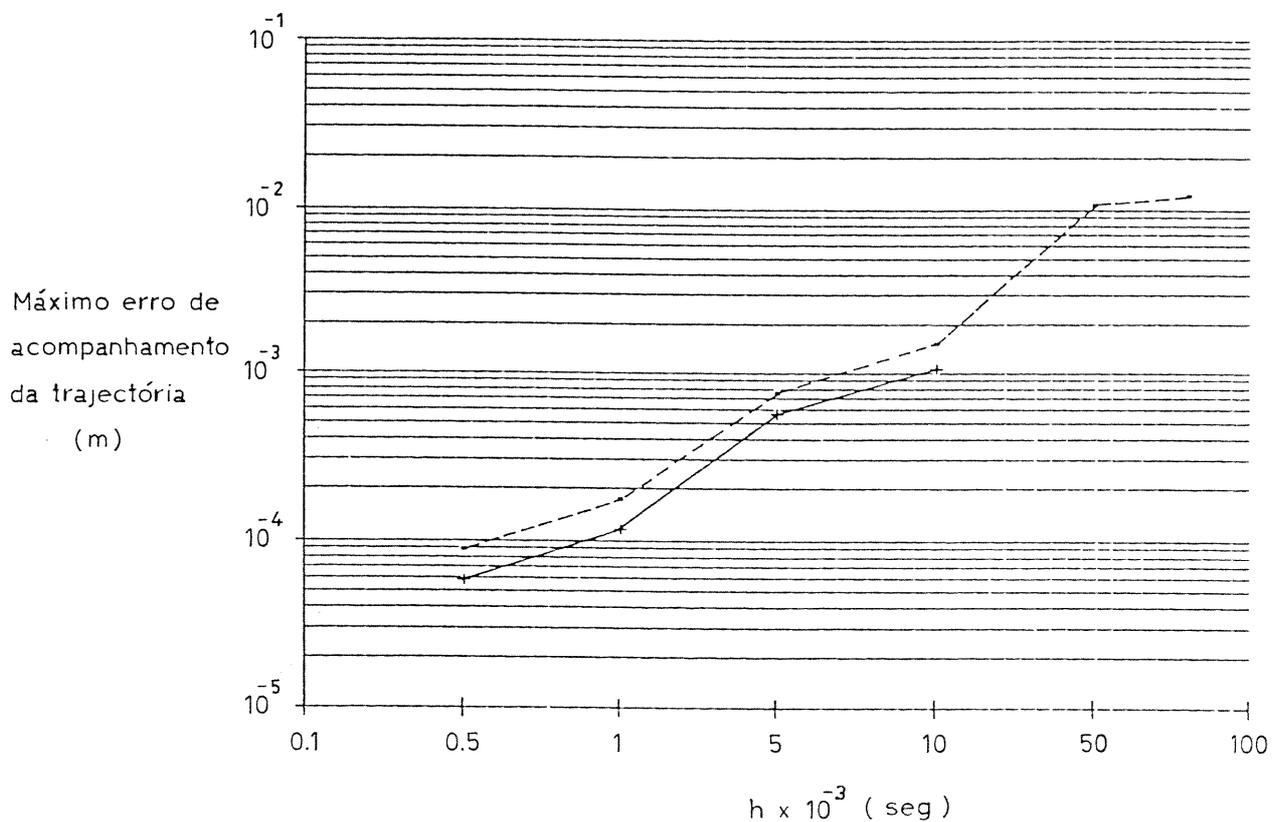


Fig. 3.7 Máximo erro de acompanhamento da trajectória teste para diferentes frequências de amostragem, com estimação exacta da massa de carga.  $T_1=0.1 h_A$   
 ----- controlador convencional  
 ————— novo controlador.

Nesta situação os melhoramentos introduzidos na malha  $J_c(q)$ ,  $G_1(s)$ ,  $G_2(s)$  são de pouca influência, sendo a maior fracção da degradação do desempenho devida ao carácter discreto, logo descontínuo, do computador de comando. Escolhendo como índice de desempenho do sistema, o máximo erro de seguimento da trajectória, os dados representados na Fig. 3.7 - melhoramento do índice com o aumento da frequência de amostragem - resultam então imediatos, provando mais uma vez a correcção dos argumentos iniciais.

### 3.5. DISCUSSÃO E CONCLUSÕES

O estudo efectuado no capítulo 2 possibilitou uma análise fácil e sistemática do algoritmo de controlo conhecido por método do binário calculado. Esta análise perspectivou o algoritmo como, um primeiro passo no sentido da obtenção de uma arquitectura de controlo mais eficiente.

Todo o estudo foi desenvolvido no domínio  $s$  (variável de Laplace). Aparentemente a utilização de um controlador discreto pareceria indicar como mais apropriada uma descrição pela variável  $z$ ; todavia, a estratégia adoptada, aliando a simplicidade com o carácter intuitivo das representações "contínuas", provou ser correcta e precisa. A dinâmica do tipo passa-baixo do sistema a controlar, conjugada com a adopção de uma frequência de amostragem "suficientemente elevada", torna lícito aproximar o bloco de amostragem e retenção por um atraso puro. Sendo assim, fica legitimada a utilização das ferramentas tradicionais no domínio  $s$ .

As observações decorrentes motivaram o desenvolvimento de uma arquitectura de controlo obedecendo a uma filosofia de gestão

eficiente dos recursos computacionais. Assim, foram planificadas e implementadas acções tais como:

- Amostragem a diferentes frequências, consoante a influência de cada bloco do sistema

- Uso da memória como meio alternativo para a obtenção de menores tempos de cálculo em tempo real

- Acção predictiva com vista a restringir os efeitos de uma amostragem lenta.

A arquitectura do controlador é não só praticável com base nos actuais microcomputadores como também compatível com a prática e experiência industrial de ajuste dos respectivos parâmetros.

Neste ponto deve notar-se a não consideração de outros fenómenos da dinâmica do robot, tais como folgas (backlash), atritos não-lineares ou flexibilidade dos dispositivos mecânicos que formam cada articulação do manipulador. A sua identificação e modelização é um problema de difícil solução não somente na robótica mas também para qualquer sistema em geral. Para um controlador conceptualmente baseado num modelo, como é o caso do sistema desenvolvido neste capítulo, podem questionar-se as possíveis performances de um sistema real que, eventualmente, inclua esse tipo de fenómenos. A imaturidade do conhecimento científico nestas áreas, aponta como plausível a concepção de controladores menos dependente do modelo matemático. No próximo capítulo esta estratégia é aplicada num controlador que não faz apelo a um modelo deste tipo. Assim, ambas as arquitecturas de controlo (Capítulos 3 e 4), devem ser consideradas como alternativas possíveis, disponíveis numa "biblioteca de controladores",

cuja escolha final dependa do desempenho exibido em cada caso particular de sistema e implementação.

## REFERÊNCIAS

- [3.1] V. D. Tourassis, C. P. Neuman  
Robust Nonlinear Feedback Control for Robotic Manipulators  
IEE Proc., vol. 132, Pt. D, pg. 134-143, July, 1985.
- [3.2] Charles P. Neuman, Vassilios D. Tourassis  
Robust Discrete Nonlinear Feedback Control for Robotic Manipulators  
J. Robotic Systems, vol. 4, pg. 115-143, Feb., 1987.
- [3.3] J. Y. S. Luh, M. W. Walker, R. P. C. Paul  
Resolved-Acceleration Control of Mechanical Manipulators  
IEEE Trans. Automat. Contr. vol. 25, pg. 468-474, June, 1980.
- [3.4] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Dynamics and Control of Robot Manipulators  
2º DINAME - Segundo Simpósio Sobre Sistemas Dinâmicos da Mecânica. Problemas Dinâmicos em Máquinas e Estruturas, Campos de Jordão, São Paulo, Brasil, 1988.
- [3.5] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Robot Manipulator Systems: Analysis and Control  
3rd Int. Symp. on Systems Analysis and Simulation, Berlim, GDR, 1988.
- [3.6] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
A Nonlinear Controller for Robot Manipulators  
Int. J. of Systems Analysis-Modelling-Simulation 1988.
- [3.7] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
Engineering Design of a Multirate Nonlinear Controller for Robot Manipulators  
J. Robotic Systems, Feb., 1989.
- [3.8] Douglas P. Glasson  
Development and Application of Multirate Digital Control  
IEEE Control Syst. Magazine, pg. 2-8, Nov., 1983.
- [3.9] J. Y. S. Luh, M. W. Walker, R. P. C. Paul  
On-Line Computational Scheme for Mechanical Manipulators  
ASME J. Dynamic Syst., Meas., Contr., vol. 102, pg. 69-76, June, 1980.
- [3.10] M. C. Leu, N. Hemati  
Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators  
ASME J. Dynamic Syst., Meas., Contr., vol. 108, pg. 172-179, Sept., 1986.

- [3.11] J. Koplik, M. C. Leu  
Computer Generation of Robot Dynamics Equations and the  
Related Issues  
J. Robotic Systems, vol. 3, pg. 301-319, Fall 1986.
- [3.12] Charles P. Neuman, John J. Murray  
Customized Computational Robot Dynamics  
J. Robotic Systems, vol. 4, pg. 503-526, Aug., 1987.
- [3.13] Charles P. Neuman, John J. Murray  
Symbolically Efficient Formulations for Computational  
Robot Dynamics  
J. Robotic Systems, vol. 4, pg. 743-769, Dec., 1987.
- [3.14] J. S. Albus  
A New Approach to Manipulator Control: The Cerebellar  
Model Articulation Controller (CMAC)  
ASME J. Dynamic Syst., Meas., Contr., vol. 97, pg. 220-  
227, Sept. 1975.
- [3.15] J. S. Albus  
Data Storage in the Cerebellar Model Articulation  
Controller (CMAC)  
ASME J. Dynamic Syst., Meas., Contr., vol. 97, pg. 228-  
233, Sept. 1975.
- [3.16] Marc H. Raibert  
Analytical Equations vs. Table Look-Up for  
Manipulation: A Unifying Concept"  
Proc. IEEE Conf. Decision Contr., New Orleans, pg. 576-  
579, Dec. 1977.
- [3.17] M. H. Raibert  
A Model for Sensorimotor Control and Learning  
Biological Cybernetics 29, pg. 29-36, 1978.
- [3.18] M. H. Raibert, B. K. P. Horn  
Manipulator Control Using the Configuration Space  
Method  
The Industrial Robot 5, n. 2, pg. 69-73, June 1978.
- [3.19] Suguru Arimoto, Sadao Kawamura, Fumio Miyazaki  
Bettering Operation of Robots by Learning  
J. Robotic Systems, vol. 1, pg. 123-140, Summer, 1984.
- [3.20] Gerhard Hirzinger  
Robot Systems Completely Based on Sensory Feedback  
IEEE Trans. Ind. Electron., vol. IE-33, pg. 105-109,  
May, 1986.
- [3.21] F. Miyazaki, S. Kawamura, M. Matsumori, S. Arimoto  
Learning Control Scheme for a Class of Robot Systems  
with Elasticity  
Proc. IEEE 25<sup>th</sup> Conf. Decision Contr., Athens, Greece,  
Dec., 1986.

- [3.22] W. Thomas Miller III  
Sensor-Based Control of Robotic Manipulators Using a  
General Learning Algorithm  
IEEE J. Robot., Automat. , vol RA-3, pg. 157-165, April  
1987.
- [3.23] W. T. Miller III, F. H. Glanz, L. G. Kraft III  
Application of a General Learning Algorithm to the  
Control of Robotic Manipulators  
The Int. J. Robotics Research, vol. 6, n. 2, pg. 84-98,  
Summer, 1987.
- [3.24] Friedrich Lange  
Der Assoziativspeicher ME7  
DFVLR, Institut für Dynamik der Flugsysteme,  
Oberpfaffenhofen, Juni 1986, DFVLR-FB 86-31.
- [3.25] Scott Y. Harmon, Alexander Meystel  
Conversa particular  
NATO ARW Highly Redundant Sensing in Robotic Systems,  
Italy, 1988.
- [3.26] Kenichi Yoshimoto, Kunihiro Wakatsuki  
Application of the Preview Tracking Control Algorithm  
to Servoing a Robot Manipulator  
Robotics Research, The First Int. Symp., pg. 883-897,  
1984.
- [3.27] Kenichi Yoshimoto, Hajime Sugiuchi  
Trajectory Control of Robot Manipulator based on the  
Preview Tracking Control Algorithm  
Robotics Research, The Second Int. Symp., pg. 171-178,  
1985.
- [3.28] Carlos D. Vinante, Carlos Bermudez, Francisco Tarre  
Predictive Compensation Implemented with a  
Microprocessor  
IEEE Control Syst. Magazine, vol. 6, pg. 40-43, Feb.,  
1986.
- [3.29] H. B. Kuntze, A. Jacobasch, J. Richalet, Ch. Arber  
On the Predictive Functional Control of an Elastic  
Industrial Robot  
Proc. IEEE 25<sup>th</sup> Conf. Decision Contr., Athens, Greece,  
Dec., 1986.
- [3.30] J. Richalet, S. Abu El Ata-Doss, C. Arber  
Predictive Functional Control: Application to Fast and  
Accurate Robots  
Proc. 10<sup>th</sup> IFAC World Congress on Automat. Contr.,  
Munich, R.F.A., 1987.
- [3.31] Chae H. An, Christopher G. Atkeson, John M. Hollerbach  
Experimental Determination of the Effect of Feedforward  
Control on Trajectory tracking Errors  
Proc. IEEE Int. Conf. Robotics and Automat., San  
Francisco, USA, April, 1986.

- [3.32] Pradeep K. Khosla, Takeo Kanade  
Experimental Evaluation of the Feedforward Compensation  
and Computed-Torque Control Schemes  
Proc. American Control Conference, Seattle, USA, June,  
1986.
- [3.33] J. R. Hewit, J. S. Burdess  
Fast Dynamic Decoupled Control for Robotics, Using  
Active Force Control  
Mech. Machine Theory, vol. 16, pg. 535-542, 1981.

## CAPÍTULO 4

# CONTROLO DE ROBOTS MANIPULADORES ATRAVÉS DE UM SISTEMA DE ESTRUTURA VARIANTE

## 4. CONTROLO DE ROBOTS MANIPULADORES ATRAVÉS DE UM SISTEMA DE ESTRUTURA VARIANTE

### 4.1. INTRODUÇÃO

O elevado peso computacional dos algoritmos de controlo de robots manipuladores propostos na literatura científica [4.1], tais como esquemas de controlo com compensação por antecipação (feedforward), controladores não-lineares, ou ainda algoritmos adaptativos, motivou o aparecimento de arquitecturas posicionadas em perspectivas diferentes.

Uma dessas estratégias consiste no desenvolvimento de sistemas com estrutura variante [4.2] (SEV) (ver Apêndice 1). Estudos teóricos [4.3-4.11] e implementações práticas [4.12-4.17], têm demonstrado as características destes sistemas, nomeadamente baixa carga computacional, robustez, boa capacidades de seguimento de trajectória e ausência da necessidade de um modelo matemático detalhado.

Nos controladores de estrutura variante propostos até ao momento, cada elo da estrutura mecânica do robot manipulador é induzido a realizar uma mímica de um sistema (desacoplado) de primeira ordem. As trajectórias resultantes são compostas de duas fases distintas. Uma primeira fase de aproximação, onde a trajectória do elo do robot se aproxima da lei linear ideal, e uma fase posterior (deslizamento) onde o sistema tenta seguir a referida característica. Esta estrutura apresenta, no entanto, dois problemas:

- Na fase de aproximação o sistema é sensível a variações dos parâmetros, não estando assim assegurada uma convergência da trajectória.

- Na fase de deslizamento, as trajectórias apresentam uma alta frequência de comutação, a qual pode excitar modos ressonantes da estrutura mecânica do robot manipulador.

Tem sido desenvolvida investigação no sentido de eliminar estas desvantagens, o que conduziu ao aparecimento de architecturas de controlo de estrutura variante com antecipação (feedforward) [4.3, 4.13], usando um modelo simplificado do robot [4.6], ou ainda algoritmos com uma estrutura de comutação (i.e. decisão) mais "suave" [4.4, 4.7, 4.8, 4.16, 4.17]. Além destas estratégias foi também apontada [4.13, 4.16, 4.17] como possível pista de investigação o uso de outro tipo de trajectórias que não as correspondentes a um sistema linear de primeira ordem.

O algoritmo proposto neste capítulo deriva destas contribuições. Assim, no parágrafo 4.2 é desenvolvido um novo controlador constituído por dois sub-sistemas, nomeadamente por um primeiro bloco correspondente a um modelo de referência, e um segundo onde é implementada a lei de comando. No que respeita à selecção do modelo de referência, foram considerados dois requisitos: escolha de uma ordem para o modelo o mais baixa possível mantendo-se, no entanto, compatível com as propriedades dinâmicas do sistema, e uso de um modelo linear, com vista a simplificar o correspondente tratamento matemático. Com base nestes argumentos, é então considerado como sendo o modelo mais apropriado, um sistema linear de segunda ordem. Por outro lado, no segundo bloco, é implementado um algoritmo, que não é mais do que uma adaptação de um esquema PI convencional, às características próprias dos sistemas de estrutura variante.

No parágrafo 4.3 é mostrado, através de simulações, o desempenho do sistema. Assim, verifica-se que a nova architectura

é facilmente ajustável com base num procedimento heurístico, que o acoplamento entre eixos é desprezável, e que as trajectórias das diversas variáveis do sistema não apresentam as indesejáveis altas frequências de comutação atrás referidas.

Finalmente, no parágrafo 4.4 a estratégia adoptada é discutida, e as conclusões são apresentadas.

#### 4.2. ALGORÍTMO DE CONTROLO

O novo algoritmo de estrutura variante proposto [4.18-4.20], é agora descrito. A arquitectura de controlo é desenvolvida considerando o problema do seguimento de posição, todavia não há qualquer perda de generalidade já que, como se verá, a extensão ao problema de seguimento de trajectória é imediata.

Nos sistemas de controlo de robots com estrutura variante propostos até ao momento, são impostos modelos lineares de primeira ordem

$$s_i = c_i q_i + \dot{q}_i = 0 \quad ; \quad i=1, \dots, n \quad (4.1)$$

onde  $n$  representa o número de graus de liberdade e  $q_i$ ,  $\dot{q}_i$  são as componentes de ordem  $i$  dos vectores  $n$  dimensionais das posições e velocidades, respectivamente. Baseados neste modelo, esses algoritmos implementam um conjunto de equações de decisão de forma a gerar uma acção de controlo  $u$ , que force o robot manipulador a apresentar um comportamento dinâmico semelhante ao modelo (4.1). Normalmente este vector de controlo é da forma

$$u = u[\text{sgn}(s)] \quad (4.2)$$

onde  $\text{sgn}()$  representa a função sinal. A estabilidade

(assimptótica) do sistema é garantida desde que a condição

$$s_i \dot{s}_i < 0 ; i=1, \dots, n \quad (4.3)$$

seja satisfeita.

Como é sabido os sistemas de primeira ordem podem apresentar trajectórias descontínuas no plano de fase. Todavia, cada elo da estrutura mecânica do robot apresenta uma certa inércia, o que implica a existência de continuidade nas trajectórias referidas. Assim, existe um conflito de exigências quando o robot manipulador se tenta comportar de um modo semelhante ao sistema de primeira ordem. Na realidade, como os actuadores não podem fornecer binários infinitos, devido a entrarem em saturação, estas trajectórias acabam por ser contínuas; contudo, a fase de aproximação demora algum tempo, o que pode acarretar problemas de convergência conforme anteriormente referido.

Estes argumentos mostram que o modelo mais simples para a referência, deverá ser, pelo menos, de ordem dois a fim de manter uma compatibilidade com a dinâmica do robot manipulador. Deste modo é necessário seleccionar um modelo de referência obedecendo a uma expressão da forma

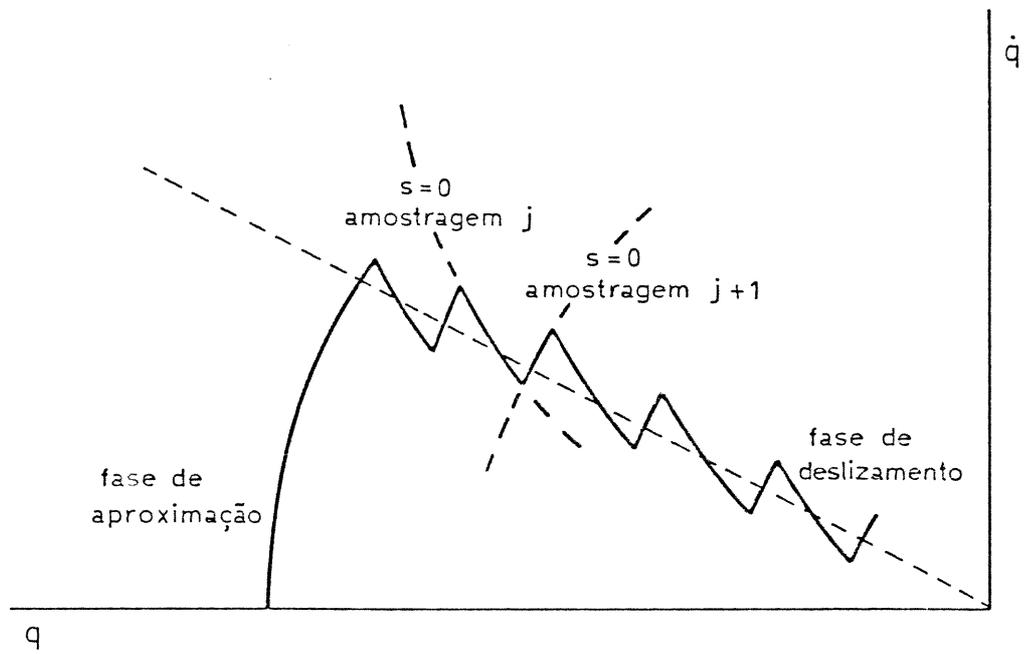
$$s_i = \ddot{q}_i + 2e_i w_{ni} \dot{q}_i + w_{ni}^2 q_i = 0 ; i=1, \dots, n \quad (4.4)$$

onde  $e_i$  é o coeficiente de amortecimento e  $w_{ni}$  a frequência natural não amortecida. Além da referida compatibilidade, este modelo apresenta ainda uma propriedade interessante: há um número infinito de trajectórias no plano de fase que satisfazem (4.4); em particular isto significa que existe sempre uma trajectória contendo um dado ponto  $(q, \dot{q})$  do plano de fase. Consequentemente são eliminados, simultâneamente, dois problemas: a indesejável

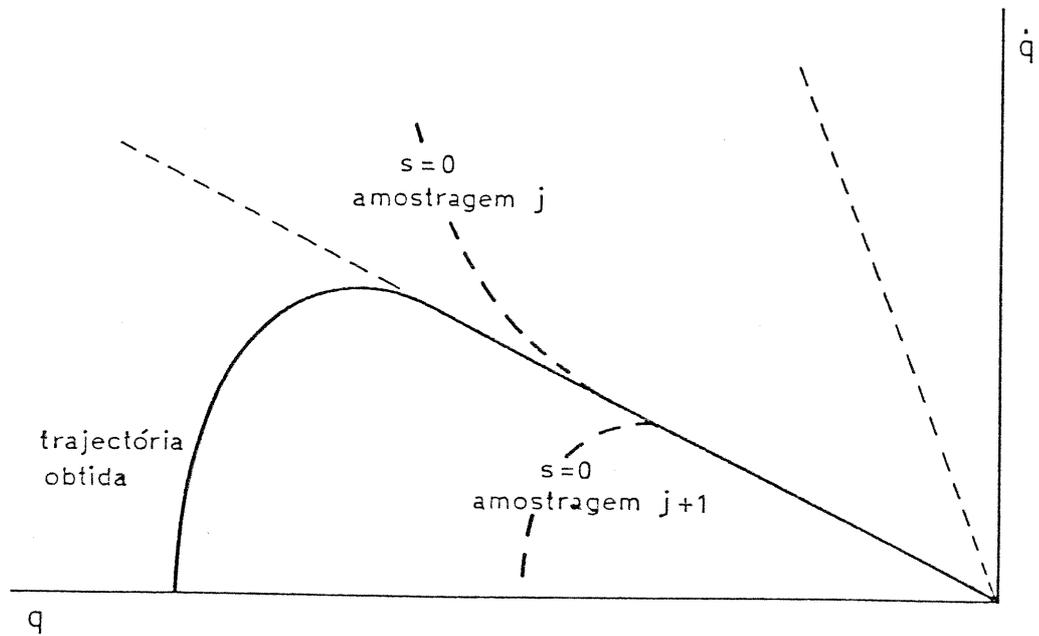
fase de aproximação é suprimida, e a amplitude da comutação existente na fase de deslizamento é fortemente atenuada. Efectivamente, quando ocorre uma perturbação, a trajectória afasta-se da ideal; se se usa um modelo de primeira ordem então o controlador reage providenciando trajectórias opostas e convergentes para a ideal. Um controlador digital introduz sempre um dado tempo de atraso computacional, pelo que aparece a já referida comutação. O uso de um modelo de segunda ordem atenua este efeito, pois neste caso o sistema não é forçado para a trajectória ideal (e única), mas em vez disso tenta seguir uma nova curva (com os mesmos valores de  $e_i$  e  $w_{n,i}$ ) que contem o presente ponto  $(q, \dot{q})$ . Deste modo o controlador segue a trajectória mais recente, quase paralela à anterior e que passa pelo ponto  $(q, \dot{q})$  (Fig. 4.1).

Teçamos agora algumas considerações prévias com vista a estabelecer uma estratégia de desenvolvimento do segundo bloco, isto é, um controlador com estrutura variante que conduza à eliminação da comutação na fase de deslizamento.

Young [4.3] fez notar que o peso do binário exigido ao controlador de estrutura variante poderia ser diminuído caso fosse introduzida uma acção de antecipação (feedforward); desta maneira a redução do binário de comutação implica uma redução semelhante na amplitude do sinal resultante. Outro ponto também focado, consiste na disparidade entre o binário produzido pelo controlador, normalmente calculado através de um processo rudimentar, e o binário realmente necessário, descrito por um complexo conjunto de equações diferenciais não-lineares. Morgan e Ozguner [4.6] experimentaram um algoritmo de controlo que refle-



a)



b)

Fig. 4.1 Trajectórias no plano de fase, resultantes de um sistema de controlo com estrutura variante usando  
 a) um modelo linear de primeira ordem  
 b) um modelo linear de segunda ordem.

ctia de modo simplificado a dinâmica do robot manipulador. O binário de controlo resultante, mais adaptado às necessidades reais do sistema, conduziu a trajectórias com uma menor amplitude de comutação. Finalmente, diversos investigadores [4.4, 4.7, 4.8, 4.16, 4.17], demonstraram que uma redução da amplitude do sinal indesejável podia também ser conseguida substituindo a acção do tipo "tudo ou nada" (on-off), dos controladores convencionais de estrutura variante, por uma lei de tipo contínuo.

Nesta ordem de ideias, e tendo em mente que para uma implementação num microcomputador, as acções de controlo devem ser simples, de modo a permitir altas frequências de amostragem, pode-se agora prosseguir o desenvolvimento do controlador de estrutura variante.

A dinâmica de robots manipuladores é descrita por um conjunto de equações diferenciais matriciais [4.21-4.22] da forma

$$T=J(q)\ddot{q}+C(q,\dot{q})+G(q) \quad (4.5)$$

onde  $J(q)$  é a matriz das inércias, com dimensão  $n \times n$ , e  $C(q,\dot{q})$  bem como  $G(q)$  representam os vectores,  $n$  dimensionais, dos binários Coriolis/centrípetos e gravitacionais, respectivamente. Dado que  $C(q,\dot{q})$  e  $G(q)$  são funções contínuas e que as variáveis  $q, \dot{q}$  são funções contínuas do tempo, então é de esperar que uma boa estimação do binário correspondente seja uma curva "suave". A estimação dessa componente ( $T_s$ ) pode ser conseguida mediante a equação

$$T_s(j)=T_s(j-1)+KT_{Ev}(j) \quad (4.6)$$

onde  $j-1$  e  $j$  representam instantes de amostragem consecutivos,  $K$  é um factor de ganho e  $KT_{Ev}$  é um binário de actualização da

estimação. Todavia, o binário total pode ter descontinuidades devido à componente  $J(q)\ddot{q}$ . Torna-se pois necessário a existência de uma forma de estimação das "componentes rápidas" isto é, da acção de ajuste de estimação bem como dos binários inerciais. Estas componentes são conseguidas a partir da estrutura variante

$$(T_{EV})_i = \begin{cases} D_i \operatorname{sgn}(s_i) & \text{se } \operatorname{abs}(s_i) > \delta_i \\ D_i s_i / \delta_i & \text{se } \operatorname{abs}(s_i) < \delta_i ; i=1,2, \dots, n \end{cases} \quad (4.7)$$

onde  $(T_{EV})_i$  designa a componente de ordem  $i$  do vector  $n$  dimensional  $T_{EV}$ , e  $\delta_i$  e  $D_i$  definem os parâmetros de uma característica do tipo proporcional/saturação, nomeadamente com  $D_i/\delta_i$  para o ganho da parte proporcional e  $D_i$  para a amplitude da saturação.

Observe-se que apesar de terem sido introduzidas com base na dinâmica dos robots manipuladores, as equações de comando (4.6) e (4.7) podem também ser interpretadas como correspondendo a um controlador PI convencional, tendo um bloco de saturação (com vista a introduzir uma acção moderadora) na parte proporcional!

Esta estrutura pode, inclusivé, ser comparada com um controlador PID, onde os efeitos do bloco diferencial são de certo modo conseguidos pelo bloco com saturação. De facto, a introdução de uma diferenciação na malha directa poderia acarretar elevados níveis de ruído já que incidiria, entre outras, sobre a componente de aceleração. Uma técnica usada para evitar este problema, consiste na sua substituição por um sistema que, por outros meios, introduza uma acção moderadora semelhante à conseguida pelo bloco de diferenciação. Assim, enquanto que a acção diferencial origina um melhor desempenho dinâmico, com a consequente

diminuição (indirecta) do sinal erro em regime transitório, o uso de um bloco de saturação tem incidência directa na limitação da sua amplitude. Deve, no entanto, notar-se que existem algumas diferenças, nomeadamente:

- A acção diferencial reflete-se numa melhoria do regime transitório, qualquer que seja a amplitude dos sinais em jogo.

- A acção de saturação sobrevem somente para níveis impróprios (i. e. inaceitáveis) do sinal erro.

Em conclusão, verifica-se uma certa semelhança quanto aos resultados conseguidos (i. e. obtenção de uma acção moderadora) a partir de blocos baseados em conceitos e especificações diferentes. A saturação pode ser encarada como um bloco de ganho variante que se adapta de forma "inteligente" à situação actual, nomeadamente reduzindo o seu valor na presença de uma dinâmica inaceitável e aumentando-o à medida que se aproxima da situação pretendida.

O binário total, no instante  $j$ , é então a soma de ambas as componentes, ou seja:

$$T(j) = T_s(j) + T_{Ev}(j) \quad (4.8)$$

O cálculo de  $s$  na expressão (4.7), através da equação (4.4), requer informação sobre a aceleração  $\ddot{q}$ . Normalmente apenas os valores da posição  $q$  e da velocidade  $\dot{q}$ , são conhecidos a partir de sensores; neste caso, tal como no método introduzido por Morgan [4.6],  $\ddot{q}$  pode ser calculada pela formula de diferenciação finita

$$\ddot{q}(j) = [\dot{q}(j) - \dot{q}(j-1)]f \quad (4.9)$$

onde  $f$  representa a frequência de amostragem do controlador. Deve

aqui ter-se em atenção, que a consideração da aceleração obtida quer mediante sensores [4.23-4.24] quer por cálculo a partir da velocidade [4.6], foi já estudada por diversos investigadores, sem verificarem níveis inaceitáveis de ruído. No presente caso, a expressão (4.9) foi calculada a uma elevada frequência de amostragem, não tendo sido observados quaisquer problemas, pelo que não foram consideradas medidas suplementares de filtragem deste sinal. Evidentemente que na prática uma filtragem será sempre necessária e aconselhável, quer se opte pela medição do sinal de aceleração, quer se opte pelo seu cálculo. No primeiro caso (medição de  $\ddot{q}(t)$ ) há a ter em conta os bem conhecidos problemas de variação do sinal a muito baixas frequências (drift) e a característica ressonante dos acelerómetros a altas frequências. Quanto à segunda possibilidade (cálculo de  $\ddot{q}(t)$ ) deve observar-se que a filtragem pode ser conseguida numa só etapa, substituindo a equação (4.9) por uma expressão de recorrência mais sofisticada e com as características apropriadas.

Finalmente, deve ainda notar-se que o algoritmo proposto é facilmente generalizável ao problema de seguimento de trajectória. Para esta situação a curva de deslizamento passa a ser da forma

$$s_i = \ddot{q}_i + 2e_i w_{ni} \dot{q}_i + w_{ni}^2 q_i - (\ddot{q}_{i,d} + 2e_i w_{ni} \dot{q}_{i,d} + w_{ni}^2 q_{i,d}) = 0 \quad (4.10)$$

onde  $q_{i,d}$ ,  $\dot{q}_{i,d}$  e  $\ddot{q}_{i,d}$  são os vectores,  $n$  dimensionais, das posições, velocidades e acelerações, de acordo com o bloco de planeamento de trajectória.

A arquitectura de controlo correspondente às expressões (4.4) e (4.6)-(4.10) está representada na Fig. 4.2, sendo  $K^* = Kf$ .

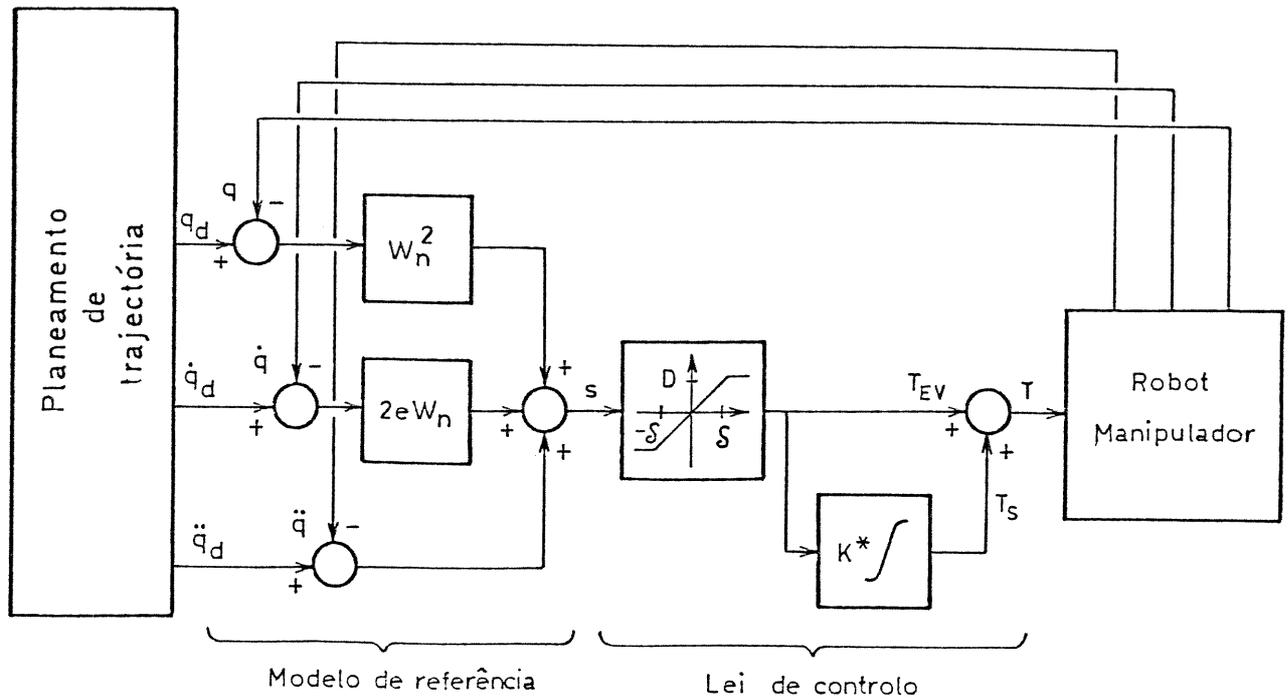


Fig. 4.2 Arquitectura do novo controlador de estrutura variante.

### 4.3. RESULTADOS E SIMULAÇÕES

Neste parágrafo são realizadas diversas simulações que ilustram o desempenho conseguido através do novo algoritmo.

Se bem que exista uma expressão analítica (ver Apêndice 2), a partir da qual se podem obter condições (suficientes mas não necessárias) do tipo

$$J^{-1}(q)D > \begin{cases} \beta(K_f)^{-1} & \text{se } \text{abs}(s) > \delta \\ \beta\delta(K_f)^{-1} & \text{se } \text{abs}(s) < \delta \end{cases}, \quad \beta \in \mathbb{R}^+ \quad (4.11)$$

e que garantem a convergência do sistema, entendeu-se que um ajuste experimental seria mais apropriado, particularmente tendo em mente a sua utilização num ambiente industrial. De facto, a expressão (4.11) fornece algumas indicações sobre o caminho

conducente à escolha dos parâmetros  $D$ ,  $\delta$  e  $K$ , todavia, é omissa relativamente aos parâmetros  $e$  e  $w_n$ . Além disso não é acessível qualquer informação sobre a "qualidade" da estabilidade obtida, problema também verificado com outros sistemas de controlo não lineares propostos na literatura científica e cujo estudo assenta em leis suficientes mas não necessárias. No caso presente esta insuficiência resulta da expressão (4.11) ter sido derivada a partir da lei geral (4.3) a qual, como já foi referido, sendo suficiente mas não necessária pode conduzir a realizações do tipo "força bruta" com poucas ou nenhuma indicações sobre o desempenho dinâmico do sistema. Assim, sabendo de antemão e com base na experiência adquirida qual a ordem de grandeza das variáveis envolvidas, considerou-se empiricamente um conjunto de valores iniciais para os parâmetros do controlador. Subsequentemente esses parâmetros foram ajustados heurísticamente e os resultados obtidos comparados com os ideais, até se chegar a desempenhos satisfatórios.

Nas simulações efectuadas é usado um robot 2R, descrito pelas expressões

$$J(q) = \begin{bmatrix} (m_1+m_2)r_1^2+m_2r_2^2+2m_2r_1r_2C_2+J_1 & m_2r_2^2+m_2r_1r_2C_2 \\ m_2r_2^2+m_2r_1r_2C_2 & m_2r_2^2+J_2 \end{bmatrix} \quad (4.11a)$$

$$C(q, \dot{q}) = \begin{bmatrix} -m_2r_1r_2S_2\dot{q}_2^2 - 2m_2r_1r_2S_2\dot{q}_1\dot{q}_2 \\ m_2r_1r_2S_2\dot{q}_1^2 \end{bmatrix} \quad (4.11b)$$

$$G(q) = \begin{bmatrix} g[m_1r_1C_1+m_2(r_1C_1+r_2C_{12})] \\ m_2gr_2C_{12} \end{bmatrix} \quad (4.11c)$$

sendo  $C_1 = \cos(q_1)$ ,  $C_{12} = \cos(q_1+q_2)$ ,  $C_2 = \cos(q_2)$  e  $S_2 = \sin(q_2)$ . Tal

como nos estudos desenvolvidos por Young [4.3], Morgan e Ozguner [4.6] e Machado e Carvalho [4.18-4.20], os parâmetros do robot manipulador foram fixados em

$$m_1=0.5 \text{ Kg}, m_2=6.25 \text{ Kg}, r_1=1 \text{ m}, r_2=0.8 \text{ m} \quad (4.12)$$

$$J_1=5 \text{ Kgm}, J_2=5 \text{ Kgm}$$

e as experiências de controlo de posição, pretendem mover o manipulador do estado inicial

$$q_1=-2.784 \text{ rad}, \dot{q}_1=0 \text{ rad/sec}, q_2=-1.204 \text{ rad}, \dot{q}_2=0 \text{ rad/sec} \quad (4.13)$$

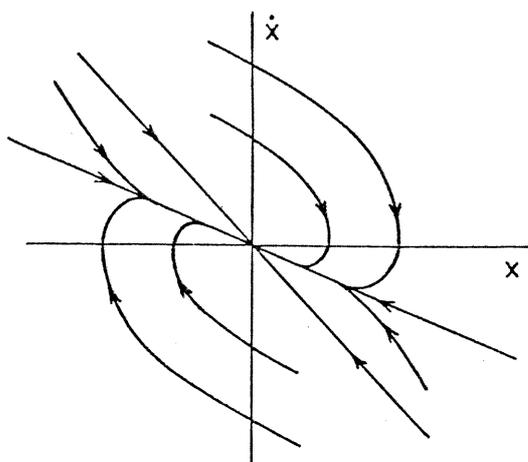
até ao estado final

$$q_1=0 \text{ rad}, \dot{q}_1=0 \text{ rad/sec}, q_2=0 \text{ rad}, \dot{q}_2=0 \text{ rad/sec} \quad (4.14)$$

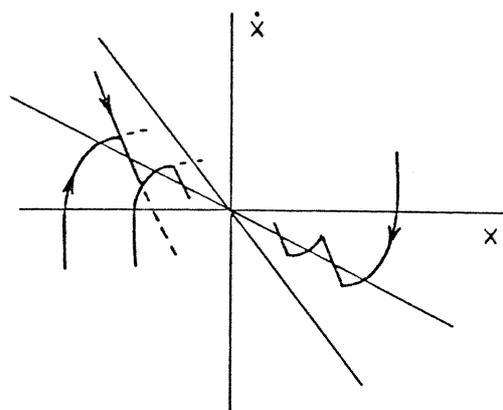
Os valores a atribuir aos parâmetros do controlador e o posterior procedimento de ajuste heurístico, assentam em pressupostos de razoabilidade, nomeadamente:

a) É desejada uma resposta transitória do tipo sobreamortecida ( $e>1$ ) ou com amortecimento crítico ( $e=1$ ); todavia, a segunda hipótese fica desde logo excluída já que, conforme pode ser observado na Fig. 4.3, a frequência finita de amostragem implica uma resposta longe da desejada, e do tipo subamortecido. Consequentemente, o valor do parâmetro  $e$  deve ser próximo mas superior à unidade.

b) Os valores atribuídos a  $w_n$  deverão reflectir as capacidades do robot manipulador. Valores demasiado elevados para  $w_n$ , correspondem a optar-se por um modelo demasiado rápido, pelo que o sistema só conseguirá acompanhar o modelo mediante a exigência de binários elevados dos actuadores. A opção inversa, isto é a



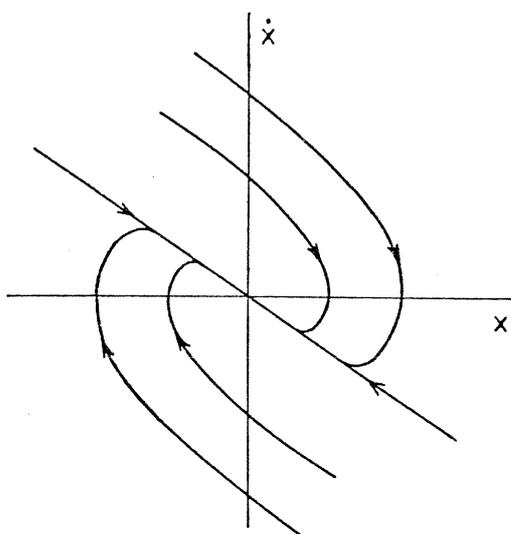
Frequência de amostragem  
infinita ( $e > 1$ )



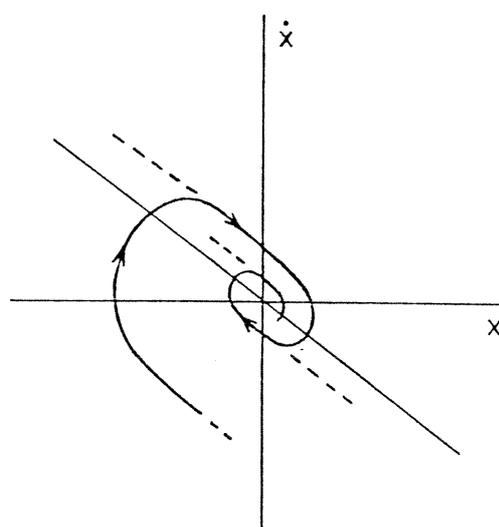
Frequência de amostragem  
finita ( $e > 1$ )

a)

a)



Frequência de amostragem  
infinita ( $e = 1$ )



Frequência de amostragem  
finita ( $e = 1$ )

b)

Fig. 4.3 Efeito do coeficiente de amortecimento ( $e$ ) e da frequência de amostragem ( $f$ ) na resposta do SEV (o efeito de  $f$  está exagerado para melhor observação).

a) Para  $e > 1$  o SEV converge com uma resposta do tipo sobreamortecido.

b) Para  $e = 1$  o SEV converge com uma resposta do tipo subamortecido.

atribuição de baixos valores para  $w_n$ , acarreta, certamente, baixos níveis dos binários dos actuadores (próximos dos níveis requeridos pelos termos gravitacionais), os quais sendo inferiores às possibilidades dos actuadores significa um sub-aproveitamento e, portanto, um desempenho inferior ao máximo possível. A solução mais eficiente é pois aquela que implica exigências aos actuadores, próximas dos seus limites máximos.

Verifica-se no capítulo 5 que para valores "razoáveis" (entenda-se por valores "razoáveis" aqueles que satisfazem os valores de desempenho desejados, todavia, compatíveis com as possibilidades reais do sistema) das variáveis posição, velocidade e aceleração, tais como ( $i=1,2$ )

$$-\pi \text{ rad} \leq q_i \leq \pi \text{ rad} \quad (4.15a)$$

$$-1 \text{ rad/s} \leq \dot{q}_i \leq 1 \text{ rad/s} \quad (4.15b)$$

$$-1 \text{ rad/s}^2 \leq \ddot{q}_i \leq 1 \text{ rad/s}^2 \quad (4.15c)$$

se obtém

$$-151.5 \text{ Nm} \leq T_1 \leq 151.5 \text{ Nm}, \quad -69.1 \text{ Nm} \leq T_2 \leq 69.1 \text{ Nm} \quad (4.16)$$

Este raciocínio será confirmado pelas experiências efectuadas seguidamente, e mostra que a observação dos níveis de binário exigidos aos actuadores, é um dos indicadores do desempenho a ter em consideração.

c) Os valores de  $D$  e  $\delta$  deverão ser de uma ordem de grandeza próxima dos binários máximos em jogo.

d) Deve observar-se a condição  $0 < K \ll 1$ , já que prevendo-se um eficiente comportamento do algoritmo de controlo, bem como elevadas frequências de amostragem, a actualização ( $KT_{\epsilon v}$ ) da componente "suave" do binário ( $T_s$ ) deverá ser de pequena ampli-

tude.

e) Um ajuste heurístico dos parâmetros do controlador, implica a adopção de procedimentos simples e compatíveis com a experiência e intuição actualmente existentes. Um método simples deverá pois conseguir uma certa "independência" entre os efeitos produzidos pelo ajuste dos diferentes parâmetros. A introdução, algo intuitiva, dos dois blocos do controlador de estrutura variante, sugere um certo desacoplamento entre os dois subsistemas (i. e. modelo de segunda ordem e lei de controlo). As experiências realizadas confirmam esta suposição pelo que o procedimento de ajuste vem consideravelmente simplificado. Assim, o ajuste de  $w_n$  influencia, primordialmente, a rapidez do sistema (e deverá manter-se superior mas próximo de 1) com as consequentes implicações nos binários requeridos aos actuadores. O ajuste de  $D$  e  $\delta$  tem influência na capacidade de seguimento do modelo de segunda ordem escolhido previamente, e na contenção de eventuais fenómenos vibratórios que surjam quando  $\delta$  assuma valores demasiadamente baixos (i. e. quando a característica proporcional/saturação fique próxima da de um relé (on-off)).

O ajuste inicial do controlador consistiu nos seguintes valores

$$e_1=e_2=2, w_{n1}=w_{n2}=10, K_1=K_2=0.1 \quad (4.15)$$

$$D_1=D_2=100, \delta_1=\delta_2=100$$

As simulações correspondentes a estes valores (Fig. 4.4) revelam trajectórias consideravelmente oscilatórias no plano de fase, o que pode ser interpretado heurísticamente como uma imagem das dificuldades sentidas pelo controlador. Aliás esta interpre-

tação coincide com a observação dos valores máximos, excessivamente elevados, dos binários exigidos aos actuadores ( $T_{1MAX}=1690$  Nm,  $T_{2MAX}=581$  Nm).

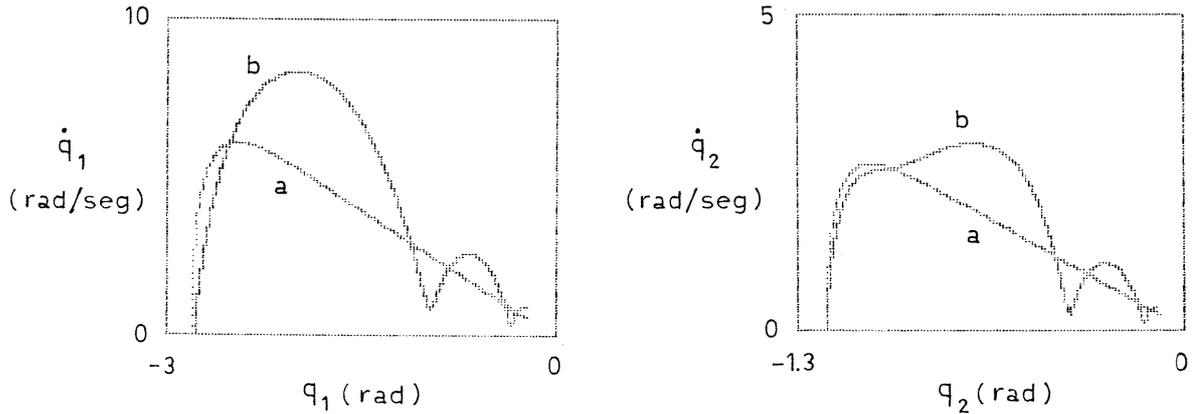


Fig. 4.4 Trajectórias no plano de fase para  $(i=1,2)$   
 $e_i=2$ ,  $w_{ni}=10$ ,  $D_i=100$ ,  $\delta_i=100$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida  
 $T_{1MAX}=1690$  Nm,  $T_{2MAX}=581$  Nm.

Para corrigir este comportamento foram experimentadas duas alternativas: o coeficiente de amortecimento  $e_i$  foi duplicado (Fig. 4.5;  $e_i=4$ ) e a frequência natural não amortecida  $w_{ni}$  foi reduzida de uma década (Fig. 4.6;  $w_{ni}=1$ ).

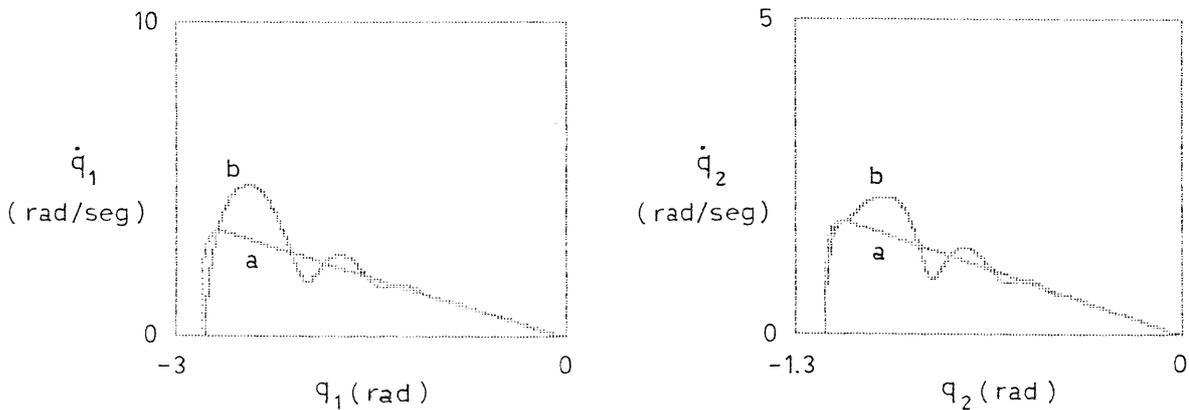


Fig. 4.5 Trajectórias no plano de fase para  $(i=1,2)$   
 $e_i=4$ ,  $w_{ni}=10$ ,  $D_i=100$ ,  $\delta_i=100$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

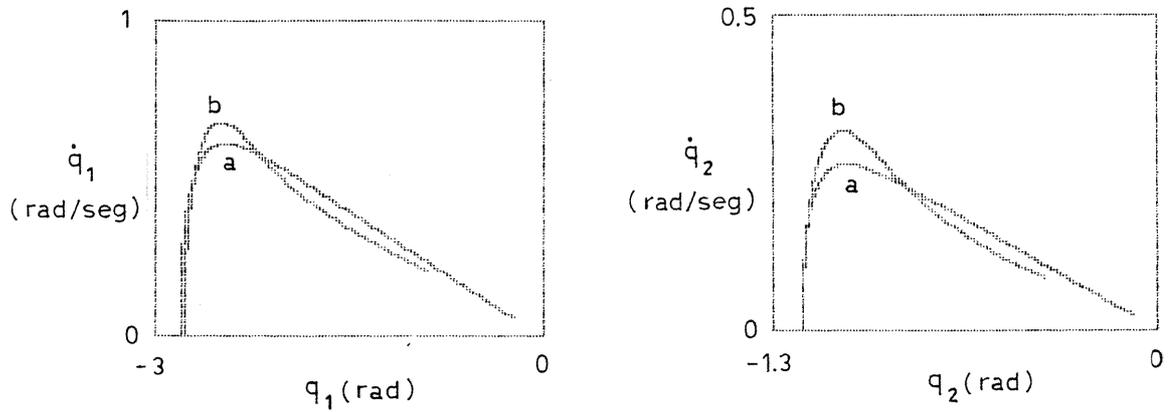


Fig. 4.6 Trajectórias no plano de fase para (i=1,2)  
 $e_i=2$ ,  $w_{ni}=1$ ,  $D_i=100$ ,  $\delta_i=100$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

Ambas as situações correspondem a sistemas mais lentos, mas somente o segundo caso mostrou melhoramentos. Este comportamento é compreensível já que uma vez definido um sistema de segunda ordem sobreamortecido, a sua dinâmica é muito mais influenciada por  $w_{ni}$  do que por  $e_i$ .

Intuitivamente é de esperar que o segundo elo da estrutura do robot, apresente uma resposta mais rápida que o primeiro. Para testar esta hipótese, foi experimentado (Fig. 4.7) um modelo com um maior valor para  $w_{n2}$  (i.e.  $w_{n1}=1$ ,  $w_{n2}=10$ ).

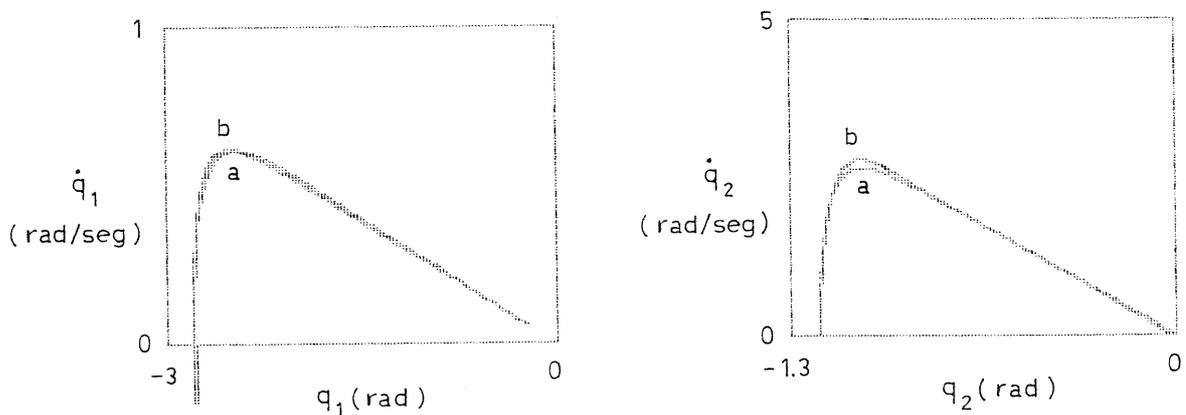


Fig. 4.7 Trajectórias no plano de fase para (i=1,2)  
 $e_i=2$ ,  $w_{n1}=1$ ,  $w_{n2}=10$ ,  $D_i=100$ ,  $\delta_i=20$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

Outras simulações revelaram que maiores ganhos na acção de controlo (4.7) permitem um melhor acompanhamento do modelo ideal; contudo, para ganhos demasiado elevados verifica-se, novamente, o aparecimento de oscilações, indicando conseqüentemente que a escolha óptima reside numa gama de valores intermédios. No entanto, as evoluções temporais da posição, velocidade (Fig. 4.8)

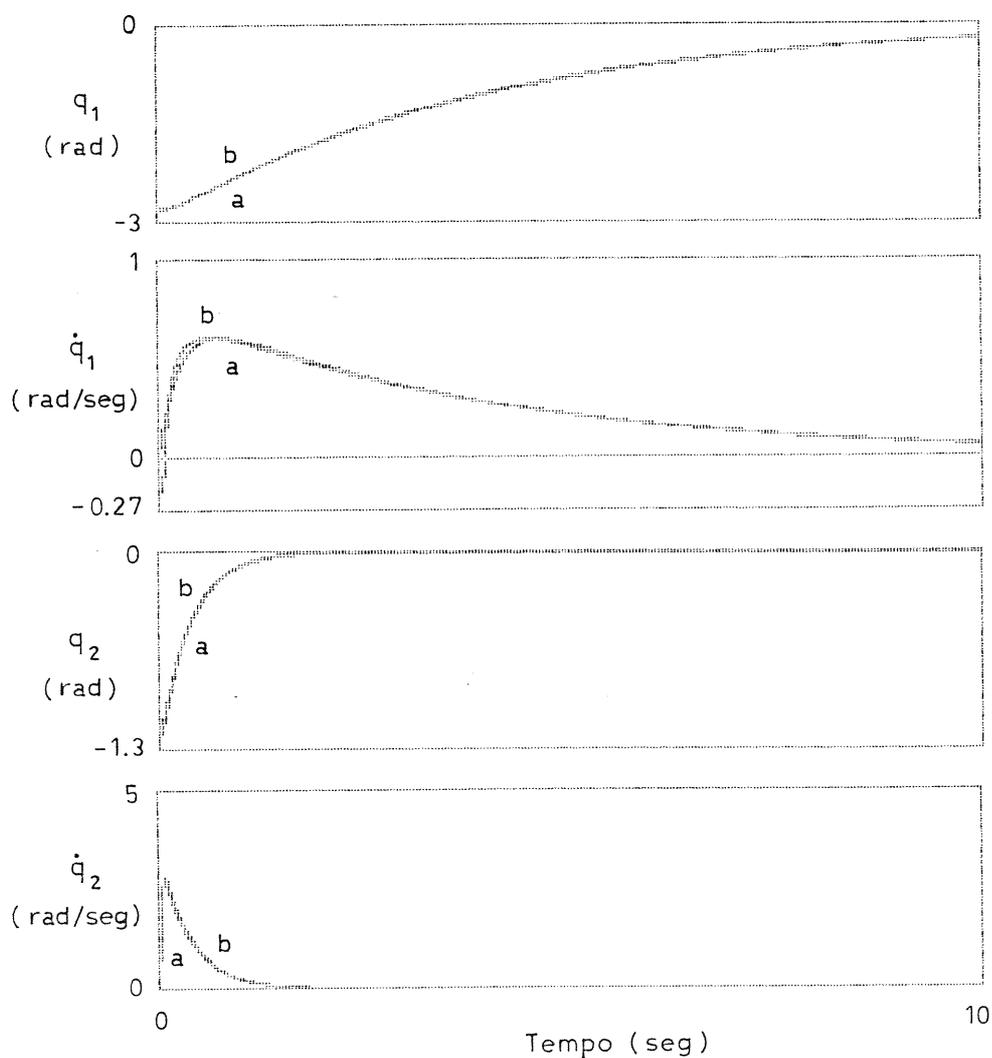


Fig. 4.8 Evolução temporal das posições e velocidades para um controlador com os parâmetros ( $i=1,2$ ):  
 $e_i=2$ ,  $w_{n1}=1$ ,  $w_{n2}=10$ ,  $D_i=100$ ,  $\delta_i=20$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

e binário (Fig. 4.9), correspondentes às trajectórias no plano de fase representadas na figura anterior, indicam dois fenómenos indesejáveis:

- Existe uma oscilação, de frequência elevada, nas curvas de binário. Na posição e na velocidade esta oscilação aparece filtrada, logo de amplitude desprezável; todavia, a sua presença pode estimular modos ressonantes da estrutura mecânica do robot.

- Durante a fase inicial do transitório são exigidos elevados binários ( $T_{1\text{MAX}}=519\text{ Nm}$ ,  $T_{2\text{MAX}}=225\text{ Nm}$ ) pelo que seriam preferíveis valores mais razoáveis.

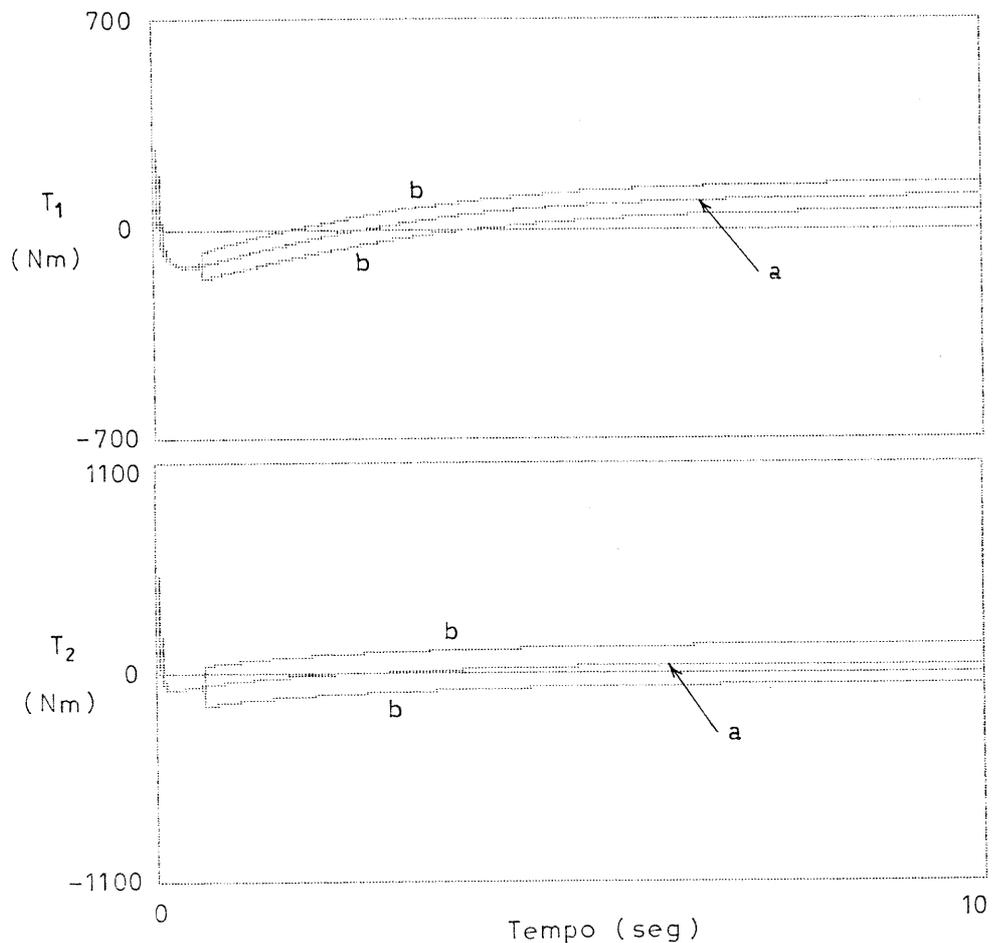


Fig. 4.9 Evolução temporal dos binários para um controlador com os parâmetros ( $i=1,2$ ):  
 $e_i=2$ ,  $w_{n1}=1$ ,  $w_{n2}=10$ ,  $D_i=100$ ,  $\delta_i=20$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida  
 $T_{1\text{MAX}}=519\text{ Nm}$ ,  $T_{2\text{MAX}}=225\text{ Nm}$ .

Ambas as situações podem ser eliminadas à custa de ganhos mais baixos na expressão (4.7), contudo, conforme já foi referido anteriormente, esse procedimento acarreta um pior acompanhamento da trajectória ideal, no plano de fase. Torna-se pois necessário um compromisso entre ambos os critérios. Com base neste pressuposto, foi testado um modelo menos rápido para o segundo elo (i.e.  $w_{n1}=1$ ,  $w_{n2}=5$ ); a resposta no plano de fase (Fig. 4.10), bem como a evolução temporal (Figs. 4.11 e 4.12), mostram claramente os excelentes resultados conseguidos. Conforme referido anteriormente a observação dos binários exigidos aos actuadores é também um possível índice do desempenho conseguido. Deste modo, os binários  $T_{1MAX}=119$  Nm e  $T_{2MAX}=97$  Nm obtidos a partir do último conjunto de parâmetros do controlador, têm amplitudes na zona de grandeza referida anteriormente e são, portanto, outros indicadores da boa adaptação conseguida entre o desempenho dinâmico exigido e as capacidades do robot manipulador.

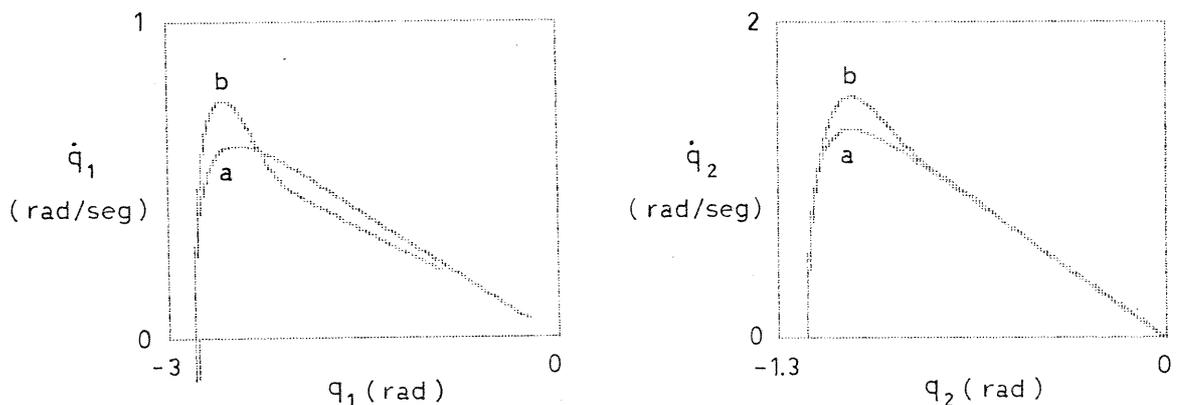


Fig. 4.10 Trajectórias no plano de fase para (i=1,2)  
 $e_i=2$ ,  $w_{n1}=1$   $w_{n2}=5$ ,  $D_i=100$ ,  $\delta_i=100$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

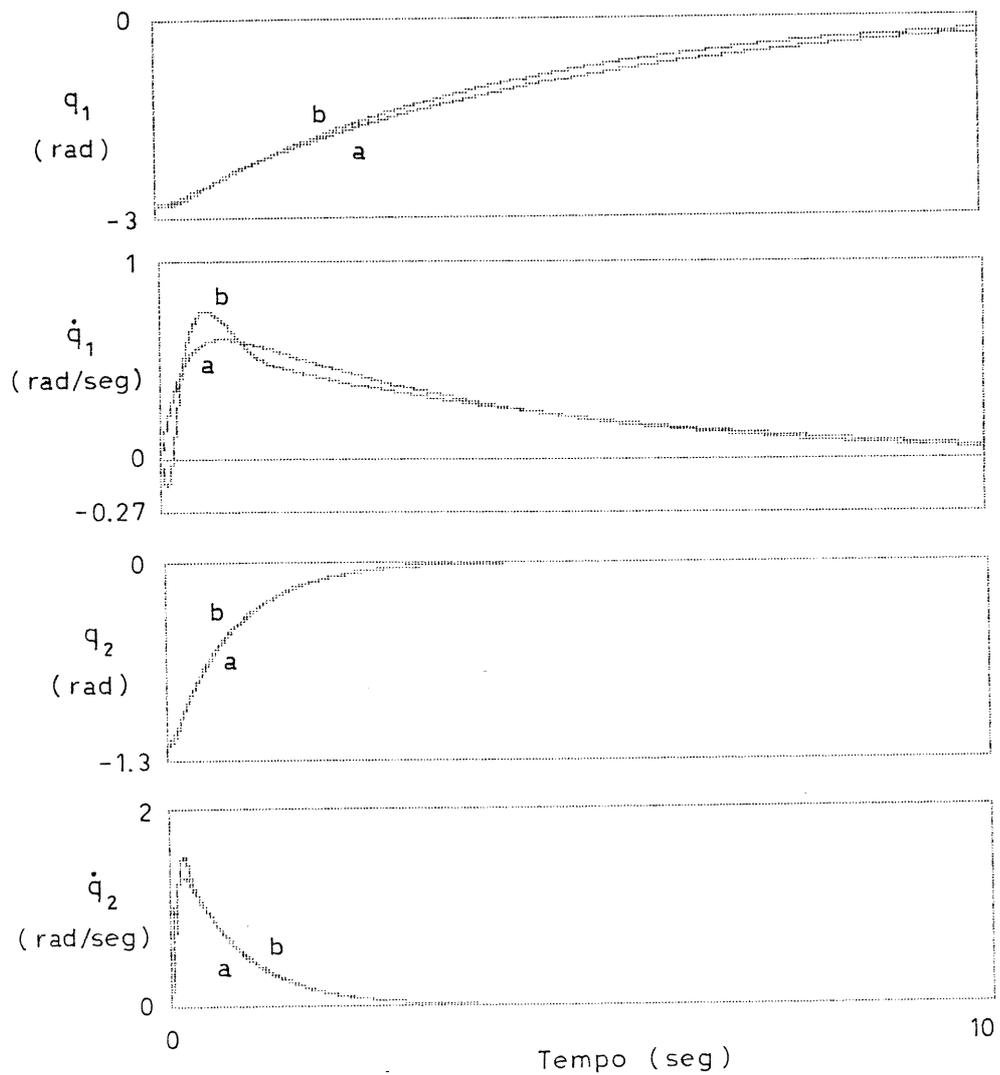


Fig. 4.11 Evolução temporal das posições e velocidades para um controlador com os parâmetros ( $i=1,2$ ):  
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

Com vista a testar a robustez do controlador relativamente a flutuações da carga, foi considerada uma redução desta para  $m_z=0.5$  Kg (Fig. 4.13). A resposta manteve-se de acordo com o modelo de referência, conforme desejado.

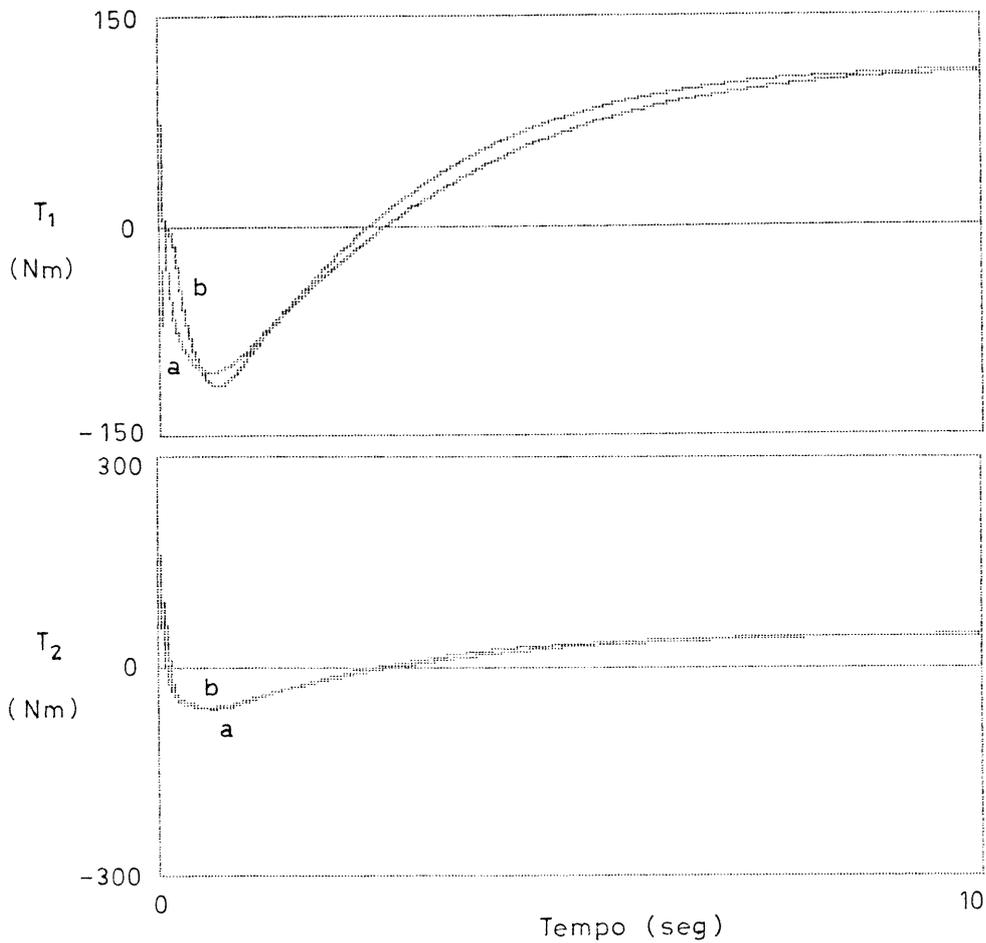


Fig. 4.12 Evolução temporal das binários para um controlador com os parâmetros ( $i=1,2$ ):  
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida  
 $T_{1\text{MAX}}=119$  Nm,  $T_{2\text{MAX}}=97$  Nm.

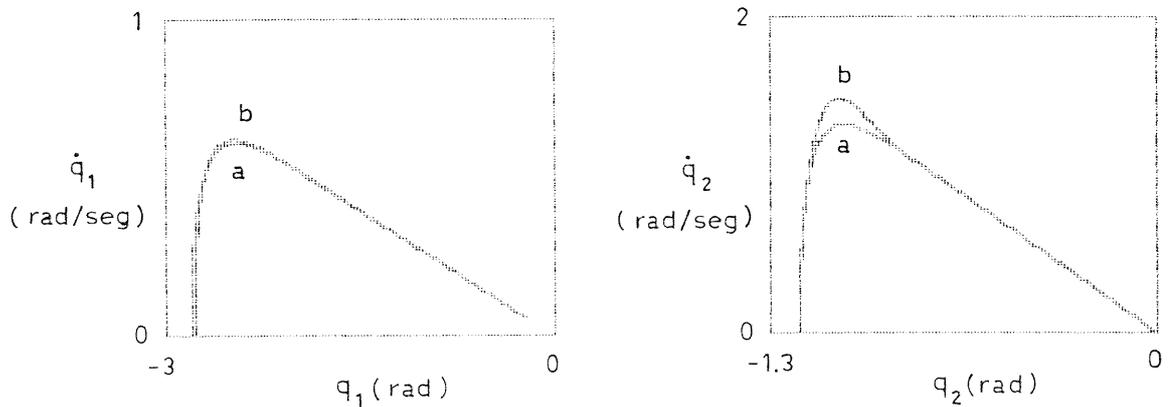


Fig. 4.13 Trajectórias no plano de fase para ( $i=1,2$ )  
 $m_2=0.5$  Kg  
 $e_1=2$ ,  $w_{n1}=1$   $w_{n2}=5$ ,  $D_1=100$ ,  $\delta_1=100$   
 frequência do controlador=2 KHz  
 a-trajectória ideal, b-trajectória obtida

Foram também experimentadas várias combinações de valores dos parâmetros do controlador, mantendo um modelo de referência com amortecimento crítico (i.e.  $e=1$ ). Tal como se esperava, as respostas manifestaram-se oscilatórias, pelo que valores do coeficiente de amortecimento, próximos e inferiores ao unitário, devem ser evitados.

Além da robustez e do desacoplamento entre eixos, os melhoramentos introduzidos no controlador de estrutura variante vêm ainda mais realçados, quando se presta atenção à evolução temporal dos binários. Este aspecto é frequentemente menosprezado na literatura científica; os binários com evoluções temporais altamente descontínuas, fornecidos pelos controladores convencionais de estrutura variante, impõem uma fadiga prejudicial, quer aos actuadores quer à estrutura mecânica, já que a corrente (pressão) dos motores eléctricos (hidráulicos), tenta seguir essa referência de controlo. O algoritmo apresentado elimina esse problema, porquanto as curvas de binário (Fig. 12) são contínuas e sem qualquer comutação.

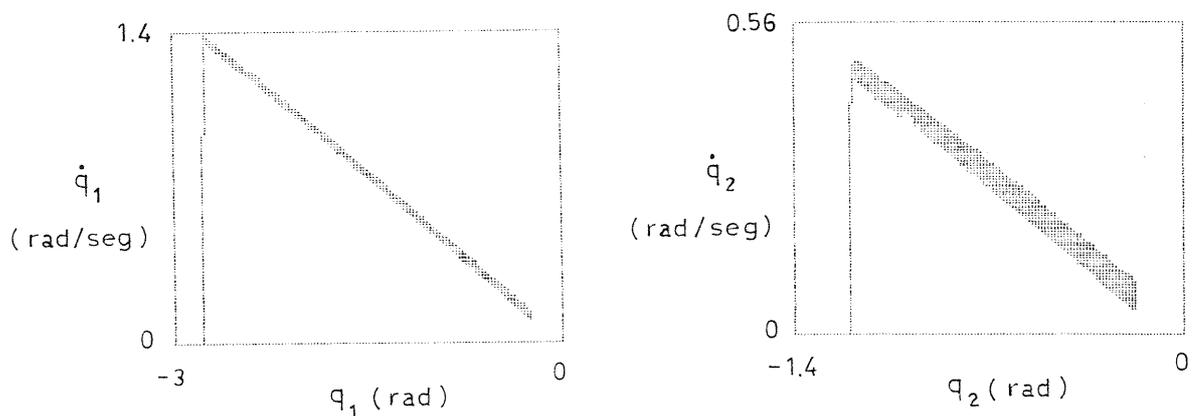


Fig. 4.14 Trajectórias no plano de fase para o controlador proposto por Young.

Para efeitos de comparação foram efectuadas simulações com o controlador proposto por Young [4.3], para os mesmos parâmetros do robot e a mesma frequência de amostragem (i.e. 2 KHz). As trajectórias no plano de fase (Fig. 4.14), bem como a evolução temporal (Figs. 4.15 e 4.16), mostra bem o elevado nível do sinal de comutação.

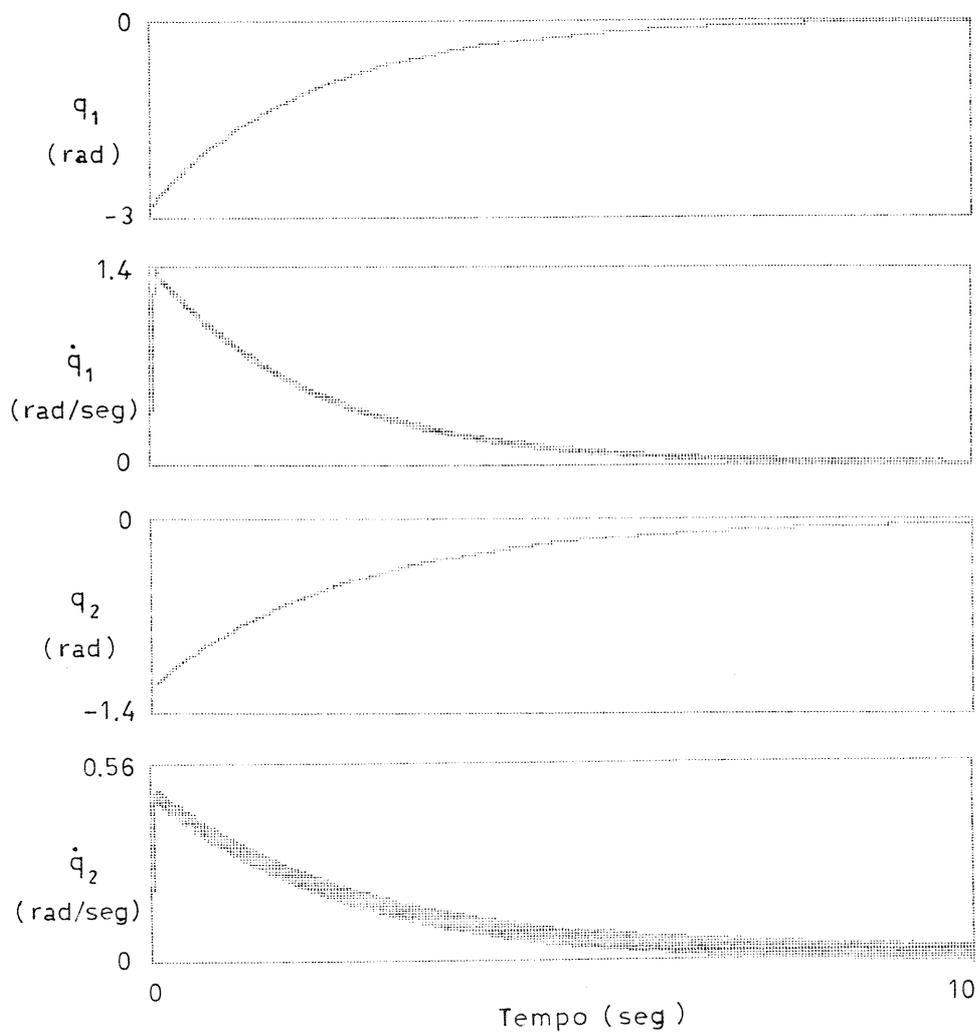


Fig. 4.15 Evolução temporal das posições e velocidades para o sistema de controlo proposto por Young.

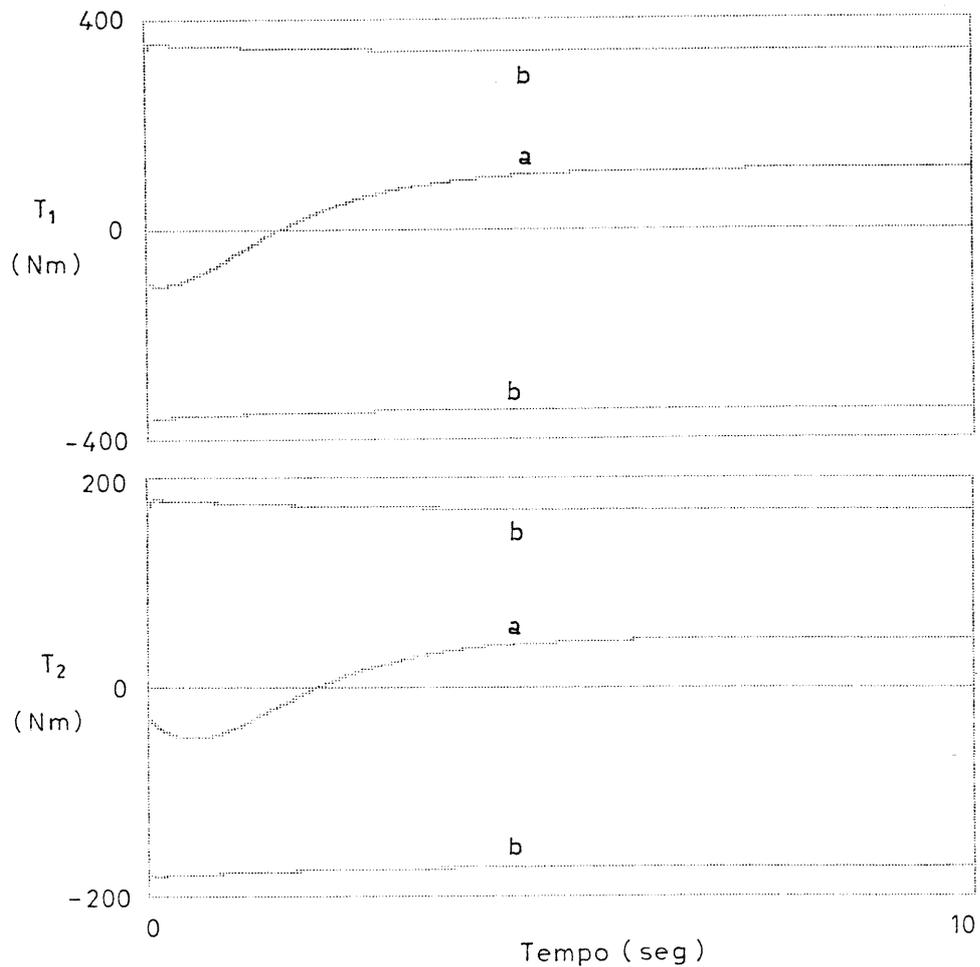


Fig. 4.16 Evolução temporal dos binários para o controlador proposto por Young.  
 $T_{1\text{MAX}}=359 \text{ Nm}$ ,  $T_{2\text{MAX}}=182 \text{ Nm}$

Além disso, considerando como índice de desempenho os binários máximos exigidos, obteve-se neste caso  $T_{1\text{MAX}}=359 \text{ Nm}$  e  $T_{2\text{MAX}}=182 \text{ Nm}$ , enquanto que para o algoritmo proposto (Fig. 4.12) se conseguiram valores muito mais moderados, nomeadamente  $T_{1\text{MAX}}=119 \text{ Nm}$  e  $T_{2\text{MAX}}=97 \text{ Nm}$ .

#### 4.4. DISCUSSÃO E CONCLUSÕES

Neste ponto começamos por discutir a metodologia empregue no desenvolvimento do novo controlador. Conforme foi observado, a integração de uma realimentação de aceleração na arquitectura de controlo com estrutura variante, resultou num considerável melhoramento do seu funcionamento. Esta estratégia corresponde a um deslocamento de perspectiva na gestão do sistema {controlador + manipulador}. De facto passou-se dos sofisticados algoritmos de controlo não-lineares ou adaptativos, para um sistema mais simples mas, em contrapartida, baseado num mais largo conjunto de informação sensorial. Os benefícios da integração na estrutura do controlador de mais informação que, meramente, as medidas de posição e velocidade, foi também referida por outros investigadores, que em filosofias de controlo linear consideraram realimentações de aceleração [4.23-4.28], binário [4.29-4.30] e corrente [4.31]. Devido à grande largura de banda destes sinais, pode encontrar-se um maior nível de ruído; todavia, estamos agora confrontados com questões do domínio da instrumentação, as quais podem ser resolvidas através de técnicas de filtragem apropriadas. Os relatórios das experiências efectuadas indicam que esses obstáculos têm sido removidos sem grandes problemas, pelo que a estratégia apresentada se afigura como uma alternativa válida.

Além disso, os resultados obtidos demonstram claramente que o processo heurístico de ajuste do controlador tem fortes semelhanças com o dos sistemas convencionais. É de referir que o procedimento heurístico adoptado não obedece à clássica metodologia de projecto a partir de expressões analíticas, todavia, mostrou a sua capacidade de condução a arquitecturas de controlo

eficientes e bem adaptadas. Se bem que no presente quadro do conhecimento científico, procedimentos heurísticos careçam ainda de uma base sistemática e teórica, são certamente de ter em consideração, tanto mais que até para os simples controladores PID lineares usados industrialmente no controlo de processos, esta perspectiva tem dado excelentes provas das suas possibilidades. Isto significa que a estrutura apresentada permite uma gestão "inteligente" dos recursos existentes a nível industrial, na medida em que aproveita da experiência e intuição já existentes. Assim, não ocorrem problemas de "incompatibilidade", frequentes nos controladores propostos e que, em grande parte, têm impedido a passagem da fase de protótipo para a de industrialização.

A estrutura de controlo desenvolvida neste capítulo, possibilitando um ajuste heurístico dos seus parâmetros, está ainda bem adaptada a um ambiente de Manufatura Integrada por Computador (CIM) gerida por um sistema pericial. De facto, o sistema pericial pode memorizar informação (previamente adquirida por experimentação) relativa aos parâmetros  $\{e, w_n, D, \delta\}$  mais apropriados a cada tipo de carga, e assim mudá-los, em linha, sempre que necessário. Este "novo" sistema de controlo mantém as características do anterior, mas permite um melhor aproveitamento das capacidades reais do manipulador. Por exemplo, para cargas pequenas ou nulas o sistema pericial pode aumentar o valor de  $w_n$  e tornar assim o sistema mais rápido, sem que sobrevenham problemas de estabilidade, já que o novo valor de  $w_n$  é escolhido de entre aqueles que o sistema pericial reconhece como mais apropriado.

Em conclusão, pode-se dizer que o algoritmo desenvolvido é uma extensão destas ideias à filosofia de sistemas de controlo com estrutura variante. O uso de modelos de segunda ordem associados a uma lei de controlo apropriada elimina a fase de aproximação, bem como a comutação verificada na fase de deslizamento. As simulações efectuadas mostram grande robustez de funcionamento, um acoplamento entre eixos com valor desprezável, baixo peso computacional e facilidade de ajuste do controlador, o que é de grande importância, em especial se se pensar na sua utilização em ambientes industriais.

## APÊNDICE 1

Para ilustrar alguns dos conceitos fundamentais na teoria dos sistemas de estrutura variante (SEV) consideremos, a título exemplificativo, um sistema linear de segunda ordem

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = ax_2 - bu \end{cases} \quad a, b > 0 \quad (A1.1)$$

$$x^T = (x_1, x_2)$$

uma lei de controlo de estrutura variante

$$u = \phi x_1 \quad (A1.2)$$

com  $\phi$  descontínuo e da forma

$$\phi = \begin{cases} \alpha & \text{se } x_1 s > 0 \\ -\alpha & \text{se } x_1 s < 0 \end{cases} \quad \alpha > 0 \quad (A1.3)$$

e onde  $s=0$  define a linha de comutação

$$s = cx_1 + x_2 = 0 \quad , \quad c > 0 \quad (A1.4)$$

Suponha-se ainda que

$$\begin{aligned} a^2/4 &< b\alpha \\ a/2 - (a^2/4 + b\alpha)^{1/2} &< -c \end{aligned} \quad (A1.5)$$

Ambas as estruturas do sistema (i. e. quando  $\phi = \alpha$  e quando  $\phi = -\alpha$ ) são instáveis (Figs. A1.1a) e b)), contudo a sua "combinação" através de (A1.4) dá origem a um sistema estável (Fig. A1.1c)). A selecção de  $\alpha$  pelas condições (A1.5) permite que  $s$  e  $\dot{s}$  tenham sinais opostos na vizinhança de  $s=0$ , ou seja

$$\lim_{s \rightarrow -0} \dot{s} > 0 \quad \text{e} \quad \lim_{s \rightarrow +0} \dot{s} < 0 \quad (A1.6)$$

ou equivalentemente

$$\dot{s} < 0 \quad (A1.7)$$

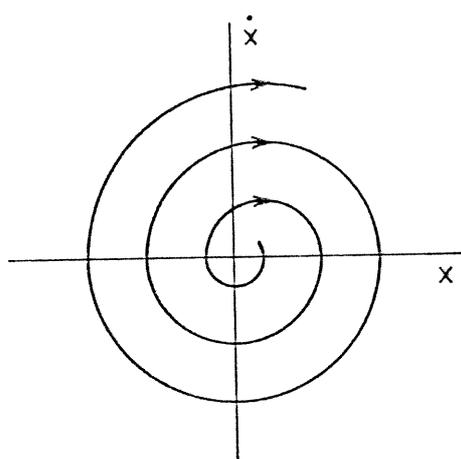
Este procedimento garante uma das propriedades fundamentais dos SEV, nomeadamente a possibilidade de se obterem trajectórias não inerentes a qualquer das sub-estruturas, e conhecida como fase de deslizamento. De facto as trajectórias de fase representadas na Fig. A1.1c), estão dirigidas para a lei de comutação (A1.4), pelo que as variáveis  $\{x, \dot{x}\}$  são compelidas a evoluir segundo essa lei. Isto implica que é assim conseguido um diferente modo de evolução (deslizamento) que não corresponde a nenhuma das trajectórias existentes nos sistemas representados nas Figs. A1.1a) e b). A equação (A1.4) determina a dinâmica do sistema durante a fase de deslizamento, a qual depende exclusivamente do parâmetro  $c$  escolhido pelo projectista. Esta invariância face aos parâmetros do sistema ou a perturbações, é de extrema importância quando se pretende controlar processos variantes no tempo ou sistemas não-lineares.

Durante a referida fase de deslizamento, e supondo uma implementação ideal da lei (A1.3) (i. e. com uma comutação em tempo nulo), então o comando  $u_{eq}(t)$  (chamado de controlo equivalente) pode ser achado a partir da equação  $\dot{s}=0$

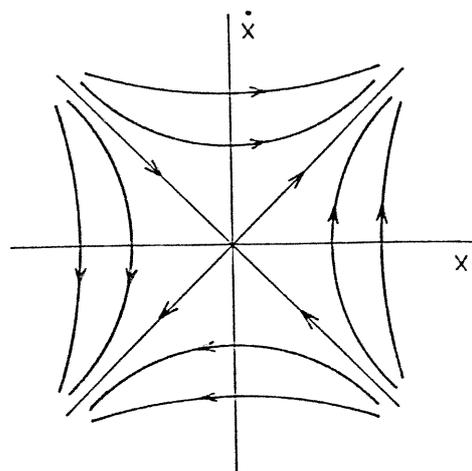
$$\dot{s} = cx_2 + ax_2 - bu = 0 \quad (A1.8)$$

Na prática a acção de controlo é comutada num tempo não nulo originando trajectórias não ideais com o aspecto de uma vibração. Além disso, há ainda a notar que antes da fase de deslizamento as trajectórias dependem da dinâmica do sistema (A1.1), e consequentemente, essa fase prévia, designada por fase de aproximação, é

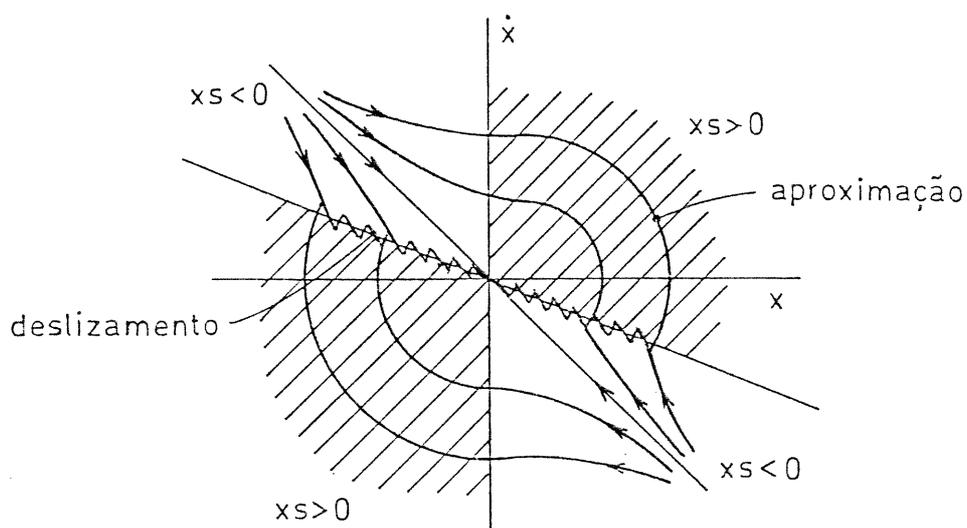
sensível a variações dos parâmetros do sistema.



a)



b)



c)

Fig. A1.1 SEV estável obtido a partir de duas estruturas instáveis.

- a) Trajectórias de fase do sistema (A1.1) para  $\phi = \alpha$ .
- b) Trajectórias de fase do sistema (A1.1) para  $\phi = -\alpha$ .
- c) Trajectórias de fase do SEV.

## APÊNDICE 2

Neste apêndice é desenvolvida uma expressão aproximada para as condições de convergência do sistema, tendo em consideração uma implementação do controlo a altas frequências de amostragem.

Para um controlador com estrutura variante, a expressão (4.3) é uma condição suficiente mas não necessária para a convergência do sistema. Todavia, tendo em vista a obtenção de expressões simplificadas, vamos usar a condição mais exigente

$$s\dot{s} < -\beta |\mu(s)| \quad (\text{A2.1})$$

onde  $\beta$  representa um qualquer número real positivo, e  $\mu(s)$  é uma função real da variável  $s$ . Para elevadas frequências de amostragem,  $\dot{s}$  pode ser aproximada pela formula de diferenciação finita

$$\dot{s} = [s(j) - s(j-1)]f \quad (\text{A2.2})$$

A partir da expressão da dinâmica do robot manipulador (4.5), e sabendo que a matriz inercial  $J(q)$  é não-singular, então temos

$$\ddot{q} = J^{-1}(q)[T - C(q, \dot{q}) - G(q)] \quad (\text{A2.3})$$

Substituindo a aceleração (A2.3) na formula (4.4) de determinação de  $s$ , depois na expressão (A2.2), e finalmente após se desprezarem os termos dependentes da posição e da velocidade (dado que somente têm uma influência de segunda ordem), resulta

$$\dot{s} \approx J^{-1}(q)[T(j) - T(j-1)]f \quad (\text{A2.4})$$

As leis de controlo (4.6)-(4.9) dão origem a

$$T(j) - T(j-1) = (1+K)T_{EV}(j) - T_{EV}(j-1) \quad (\text{A2.5})$$

O pressuposto de altas frequências de amostragem, leva a que existam pequenas variações no binário de controlo entre duas amostragens consecutivas, devido à continuidade promovida pela expressão (4.4), bem como pelas leis (4.6)-(4.9). Assim, a expressão (A2.5) pode ser aproximada por

$$T(j) - T(j-1) \approx K T_{Ev}(j) \quad (A2.6)$$

e substituindo em (A2.4) resulta para qualquer instante de amostragem

$$\dot{s} = J^{-1}(q) K_f T_{Ev} \quad (A2.7)$$

Se reduzirmos a lei de controlo (4.7) à forma mais compacta

$$T_{Ev} = -D\alpha(s) \quad (A2.8)$$

onde

$$\alpha(s) = \begin{cases} \text{sgn}(s) & \text{se } \text{abs}(s) > \delta \\ s/\delta & \text{se } \text{abs}(s) < \delta \end{cases} \quad (A2.9)$$

e reescrevendo (A2.1), vem

$$J^{-1}(q) K_f D \alpha(s) s > \beta |\mu(s)| \quad (A2.10)$$

Fazendo

$$\mu(s) = \begin{cases} s & \text{se } \text{abs}(s) > \delta \\ s^2 & \text{se } \text{abs}(s) < \delta \end{cases} \quad (A2.11)$$

para  $s \ll 0$ , (A2.10) dá origem a

$$J^{-1}(q) D > \begin{cases} \beta (K_f)^{-1} & \text{se } \text{abs}(s) > \delta \\ \beta \delta (K_f)^{-1} & \text{se } \text{abs}(s) < \delta \end{cases} \quad (A2.12)$$

A partir desta expressão, o vector  $D$  pode ser calculado de modo a que o controlador de estrutura variante garanta a convergência do sistema. Por outro lado (A2.12) sugere diversas interpretações para os parâmetros do controlador, nomeadamente:

-Os parâmetros  $K$  e  $f$  têm influência semelhante

-Os binários Coriolis/centrípetos e gravitacionais têm uma influência de segunda ordem

-Quanto maiores forem as inércias do robot manipulador tanto maior será o valor necessário para  $D$ . Isto quer dizer que o algoritmo de estrutura variante poderá ser chamado a providenciar maiores binários, dado que o deslizamento é especialmente dependente das inércias a controlar.

-Transições suaves no bloco de estrutura variante (4.7), conseguidas à custa de maiores valores de  $\delta$ , podem impôr restrições sobre os menores valores possíveis para  $D$ .

## REFERÊNCIAS

- [4.1] Chang-Huan, Yen-Ming Chen  
Multi-Microprocessor-Based Cartesian-Space Control  
Techniques for a Mechanical Manipulator  
IEEE J. Robotics and Automation, vol. RA-2, pp. 110-  
115, June 1986.
- [4.2] Vadim I. Utkin  
Variable Structure Systems With Sliding Modes  
IEEE Trans. Automat. Contr., vol. AC-22, pp. 212-222,  
April, 1977.
- [4.3] Kar-Keung D. Young  
Controller Design for a Manipulator Using Theory of  
Variable Structure Systems  
IEEE Trans. Syst., Man, Cybern., vol. SMC-8, pp. 101-  
109, Feb. 1978.
- [4.4] J. J. Slotine, S. S. Sastry  
Tracking Control of Non-Linear Systems Using Sliding  
Surfaces, with Application to Robot Manipulators  
Int. J. Control, vol. 38, n. 2, pp. 465-492, 1983.
- [4.5] S. Lefebvre, S. Richter, R. De Carlo  
Decentralised Variable Structure Control Design for a  
Two-Pendulum System  
IEEE Trans. Automat. Contr., vol. AC-28, pp. 1112-1114,  
Dec., 1983.
- [4.6] Russel G. Morgan, Umit Ozguner  
A Decentralized Variable Structure Control Algorithm  
for Robotic Manipulators  
IEEE J. Robotics and Automation, vol. RA-1, pp. 57-65,  
March, 1985.
- [4.7] Jean-Jacques E. Slotine  
Robustness Issues in Robot Control  
IEEE Int. Conf. Rob. Automat., St. Louis, Missouri,  
1985.
- [4.8] Jean-Jacques E. Slotine  
The Robust Control of Robot Manipulators  
The Int. J. Robotics Research, vol. 4, n. 2, pp. 49-64,  
Summer, 1985.
- [4.9] Kazuhiro Kasuge, Katsuhisa Furuta  
Variable Structure Control of a Robot Arm  
Proc. 15th ISIR, Tokyo, Japan, 1985.
- [4.10] F. Harashima, J. X. Xu, H. Hashimoto, T. Ichiyama  
Tracking Control of Robot Manipulator Using Sliding  
Mode  
Proc. 15th ISIR, Tokyo, Japan, 1985.

- [4.11] K. David Young  
A Variable Structure Model Following Control Design For  
Robotic Applications  
IEEE Int. Conf. Rob. Automat., San Francisco,  
California, 1986.
- [4.12] Charles A. Klein, John J. Maney  
Real-Time Control of a Multiple-Element Mechanical  
Linkage with a Microcomputer  
IEEE Trans. Ind. Electron. Contr. Instrum., vol. IECI-  
26, pp. 227-234, Nov., 1979.
- [4.13] Masato Hiroi, Masayuki Hojo, Yukio Hashimoto,  
Yoshikazu Abe, Yasuhiko Dote  
Microprocessor-Based Decoupled Control of Manipulator  
Using Modified Model-Following Method with Sliding Mode  
IEEE Trans. Ind. Electron., vol. IE-33, pp. 110-113,  
May, 1986.
- [4.14] A. Staszulonek, H. Van Brussel  
Inertially Decoupled, Sliding Mode Controller Design  
for Trace and Pick-Up Robot  
Proc. 16th ISIR, Brussels, Belgium, 1986.
- [4.15] F. Harashima, H. Hashimoto, K. Maruyama  
Practical Robust Control of Robot Arm Using Variable  
Structure System  
IEEE Int. Conf. Rob. Automat., San Francisco,  
California, 1986.
- [4.16] Hideki Hashimoto, Koji Maruyama, Fumio Harashima  
A Microprocessor-Based Robot Manipulator Control with  
Sliding Mode  
IEEE Trans. Ind. Electron., vol. IE-34, pp. 11-18,  
Feb., 1987.
- [4.17] Hideki Hashimoto, Fumio Harashima  
Variable Structure Strategy in Motion Control  
Proc. 10th IFAC World Congress on Automatic Control,  
Munich, FRG, 1987.
- [4.18] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
A Smooth Variable Structure Control Algorithm for Robot  
Manipulators  
IEE Control '88, Oxford, UK, 1988.
- [4.19] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
A Variable Structure Control Algorithm for Robot  
Manipulators Using Acceleration Feedback  
NATO ARW on Highly Redudant Sensing in Robotic Systems,  
Il Ciocco, Italy, 1988.
- [4.20] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
A New Variable Structure Controller for Robot  
Manipulators  
Third IEEE Int. Symp. on Intelligent Control,  
Arlington, USA, 1988.

- [4.21] Richard P. Paul  
Robot Manipulators: Mathematics, Programming and Control  
Cambridge, MA: Mass. Inst. Tech., 1981.
- [4.22] Michael Brady, John M. Hollerbach, Timothy L. Johnson, Tomas Lozano-Perez, Matthew T. Mason  
Robot Motion: Planning and Control  
Cambridge, MA: Mass. Inst. Tech., 1982.
- [4.23] Sigeru Futami, Nobuhiro Kyura, Shujiro Hara  
Vibration Absorption Control of Industrial Robots by Acceleration Feedback  
IEEE Trans. Ind. Electron., vol. IE-30, pp. 299-305, Aug., 1983.
- [4.24] Paul T. Kotnik, Stephen Yurkovich, Umit Ozguner  
Acceleration Feedback for Control of a Flexible Manipulator Arm  
J. Robotic Systems, vol. 5, pg. 181-196, June, 1988.
- [4.25] Gong-Liang Luo, George N. Saridis  
Robust Compensation for a Robotic Manipulator  
IEEE Trans. Automat. Contr., vol. AC-29, pp. 564-567, June, 1984.
- [4.26] G. L. Luo, G. N. Saridis  
L-Q Design of PID Controllers for Robot Arms  
IEEE J. Robotics and Automation, vol. RA-1, pp. 151-157, Sept., 1985.
- [4.27] John Studenny, Pierre R. Bélanger  
Robot Manipulator Control by Acceleration Feedback: Stability, Design and Performance Issues  
Proc. 25th IEEE Conf. Decision and Control, Athens, Greece, 1986.
- [4.28] F. Berlin, W. M. Grimm, P. M. Frank  
On Robot Motion Control with Acceleration Feedback  
IFAC Symp. on Robot Control, Karlsruhe, FRG, 1988.
- [4.29] H. Van Brussel, L. Vastmans  
A Compensation Method for Controlling Time-Varying Robot Configurations  
Proc. 14th ISIR, Gothenburg, Sweden, 1984.
- [4.30] J. Y. S. Luh, William D. Fisher, Richard P. Paul  
Joint Torque Control by a Direct Feedback for Industrial Robots  
IEEE Trans. Automat. Contr., vol. AC-28, pp. 153-161, Feb., 1983.
- [4.31] J. O'Shea, A. B. Turgeon  
Benefits of Positive Current Feedback Inner Loop for Robot Manipulators  
Proc. IFAC/IFIP/IMACS Symp. Theory of Robots, Vienna, Austria, 1986.

CAPÍTULO 5

ASPECTOS COMPUTACIONAIS  
NO CONTROLO DE  
ROBOTS MANIPULADORES

## 5. ASPECTOS COMPUTACIONAIS NO CONTROLO DE ROBOTS MANIPULADORES

### 5.1. INTRODUÇÃO

Este capítulo debruça-se sobre a optimização dos recursos computacionais envolvidos no cálculo, em tempo real, da cinemática, dinâmica e controlo de robots manipuladores.

Optou-se por seguir, como exemplo base, o cálculo da dinâmica inversa de manipuladores (i. e. cálculo dos binários requeridos para uma dada concretização dos valores desejados da posição, velocidade e aceleração nos eixos); todavia, como se verificará, a generalização dos conceitos desenvolvidos aos demais blocos da cinemática e do controlo é simples e imediata.

O cálculo da dinâmica inversa de robots tem estado, geralmente, ligada à forma sob a qual tem sido "materializado" o respectivo modelo. De facto, considerando um manipulador como um conjunto de elos rígidos interligados através de articulações accionadas por actuadores ideais (i. e. com resposta instantânea, sem saturação, etc), desprezando efeitos não-lineares tais como folgas (backlash), atritos, ou mesmo flexibilidades, e usando os métodos clássicos de modelização, obtém-se um algoritmo descritivo da respectiva dinâmica ideal. Contudo, a complexidade dos cálculos algébricos envolvidos restringe a obtenção de expressões simbólicas, por meios manuais, a robots com um máximo de três graus de liberdade. Assim, para obviar a esta impossibilidade prática no caso de robots com um maior número de graus de liberdade, desenvolveram-se, nas décadas de sessenta e setenta, metodologias baseadas no formalismo de Lagrange [5.1, 5.2] (ver capítulo 2). Em 1980 Hollerbach [5.3] culminou esses esforços propondo um algoritmo que tira o máximo partido desta estratégia

sistemática. Entretanto, também em 1980, Luh, Walker e Paul [5.4] desenvolveram um método alternativo - método de Newton-Euler - também baseado na física clássica, todavia, não sendo guiado (e condicionado) pela estrutura sistemática e formal de Lagrange. A obtenção deste algoritmo, baseada em processos "não-sistemáticos" mostrou ser complexa embora, em contrapartida, proporcione implementações bastante mais eficientes. Curiosamente ambos os métodos dão origem a algoritmos de cálculo da dinâmica inversa com a "forma" de expressões numéricas recursivas. Aliás, em 1982, Silver [5.5] provou serem os dois métodos equivalentes (o que seria de esperar já que correspondem a "materializações" dissemelhantes de um mesmo modelo), sendo a melhoria da eficiência uma consequência da adopção de referenciais estrategicamente melhor situados.

Paralela e alternativamente a esta "forma" de cálculo da dinâmica inversa, foram estudados por Albus [5.6, 5.7] e Raibert [5.8-5.11] esquemas baseados no uso total ou parcial da memória como expediente para aliviar a carga computacional envolvida. Não obstante, estes métodos propondo uma troca das exigências em cálculo por exigências em memória, só muito recentemente tiveram seguimento (Hirzinger e outros [5.12-5.15], Lange [5.16], Miller III et. al. [5.17, 5.18]), tendo permanecido algo "esquecidos" devido ao apuramento da eficiência de cálculo conseguida com o método de Newton-Euler. Efectivamente, durante alguns anos considerou-se ter sido atingido o estado de máxima optimização do algoritmo de cálculo, restando como caminho para futuros progressos o desenvolvimento de arquitecturas computacionais multiprocessador com a eventual concepção de circuitos VLSI dedicados. A

pesquisa neste sector, iniciada em 1982 por Luh e Lin [5.19], e presentemente ainda em desenvolvimento, mostrou ser complexa e inúmeros trabalhos (entre outros [5.20-5.24]) segundo esta perspectiva atestam o desempenho fortemente limitado alcançado com as arquitecturas computacionais entretanto concebidas. Estes problemas, que serão detalhados posteriormente, sugerem que o cálculo por multiprocessamento do algoritmo de Newton-Euler talvez não seja a única, ou a melhor via para progressão no sentido de estruturas mais e mais eficientes.

Já em 1984 Horak [5.25] dava o mote a esta perspectiva, propondo um algoritmo repartido (partitioned algorithm). Para os três primeiros graus de liberdade de um robot industrial (i. e. o braço) a dinâmica inversa poderia ser calculada mediante fórmulas simbólicas, e nos três graus de liberdade finais (i. e. o pulso) usar-se-ia o já referido método de Newton-Euler. Os "truques" inerentes a este processo advêm da facilidade na obtenção manual das fórmulas simbólicas para uma estrutura mecânica com três graus de liberdade, bem como a notável simplificação do método de Newton-Euler conseguida para a estrutura do punho. De facto, muitas das constantes envolvidas no modelo do punho são nulas devido ao cruzamento dos três eixos num único ponto, já que este é frequentemente construído de modo a comportar-se como uma única articulação esférica com três graus de liberdade (punho esférico). Este efeito é conseguido associando três eixos, cada com um grau de liberdade e que se intersectam num só ponto. Horak refere um desempenho algumas vezes superior ao conseguido pelo algoritmo de Newton-Euler.

A percepção da eventual existência de novos e melhores métodos computacionais, aliada ao aparecimento de sistemas infor-

máticos e programas apropriados ao tratamento e simplificação de expressões simbólicas levou a que em 1986 Faessler [5.26], Leu e Hemati [5.27] e Koplic e Leu [5.28] anunciassem a obtenção da dinâmica inversa na "forma" simbólica, a partir de métodos computacionais automáticos. O tempo de cálculo, em tempo real, mostrou ser ainda mais rápido que anteriormente. Por outro lado, as expressões obtidas possibilitam uma análise mais detalhada dos fenômenos dinâmicos presentes, estudo que segundo Yang e Tzeng [5.29] pode conduzir a novas estruturas mecânicas, projectadas tendo em vista a eliminação de alguns desses efeitos, com a consequente simplificação de fórmulas e respectivo cálculo.

Mais recentemente, em 1987, Neuman e Murray [5.30-5.32] mostraram que, quer o cálculo a partir do método de Newton-Euler, quer o cálculo baseado nas expressões simbólicas, poderia ser otimizado se se tomassem em consideração particularidades da estrutura mecânica de cada robot tais como: eixos perpendiculares ou paralelos, punhos esféricos, matrizes de inércia e vectores de centros de massa esparsos e outras constantes nulas. Assim, qualquer dos algoritmos referidos uma vez utilizado na sua forma geral fica restringido a uma eficiência muito inferior à máxima possível, a qual se pode obter aliviando-os, previamente (off-line) de cálculos desnecessários, nomeadamente:

- Adições e multiplicações por zero.
- Multiplicações por um ou por menos um.
- Usando um procedimento sistemático de organização, que remove cálculos repetidos.

Este método, apelidado de computação por medida (customized computing), tira partido dessas particularidades e mostra que,

uma vez otimizados segundo esta óptica, existem situações favoráveis ao cálculo pelo método de Newton-Euler, e outras mais indicadas à computação a partir das fórmulas simbólicas. A partir da observação dos resultados obtidos para diversas estruturas de robots presentemente usados, Neuman e Murray desenvolveram uma expressão heurística, baseada em vários parâmetros tais como o número total de graus de liberdade ( $n$ ) ou o número de graus de liberdade rotacionais ( $n_R$ ), que indica qual o método mais apropriado ao cálculo em tempo real. Em particular verificou-se que para manipuladores com  $n=n_R=6$  era preferível o algoritmo de Newton-Euler, enquanto que para manipuladores com  $n=3$  ( $n_R=1, 2$  ou  $3$ ), ou manipuladores com  $n=6$ ,  $n_R=0$  ou  $1$ , a opção deveria incidir no cálculo a partir de expressões simbólicas.

É interessante observar que as estratégias propostas desde 1984 tiveram, até ao momento, repercussões pouco significativas no desenvolvimento de estruturas de multiprocessamento, o que, a verificar-se no futuro, poderá fornecer novas indicações sobre este tema.

É ainda de referir a apresentação de três técnicas alternativas propostas na literatura científica, e que se demonstrou partirem de conceitos incorrectos ou imprecisos. Paul [5.33] propôs a eliminação de termos presentes na dinâmica inversa, desde que, para os valores previsíveis de posição ( $q$ ), velocidade ( $\dot{q}$ ), aceleração ( $\ddot{q}$ ) e carga ( $m_L, J_L$ ), tivessem amplitudes máximas significativamente inferiores aos demais. Não obstante, este caminho é incorrecto pois não deverão ser comparados valores máximos independentemente dos vectores  $\{q, \dot{q}, \ddot{q}, m_L, J_L\}$ , mas sim as amplitudes de todos os termos para o mesmo valor dos referidos vectores. A eliminação será somente possível quando a amplitude

for sempre (i. e. qualquer que seja o valor dos vectores) marcadamente inferior à dos demais termos. Numa ordem de ideias algo semelhante foi também considerada a eliminação dos termos Coriolis/centrípetos (aos quais corresponde a maior carga computacional), devido a serem de pequena amplitude aquando do início e fim de um movimento, onde as velocidades apresentam valores relativamente baixos. Segundo este raciocínio os maiores erros apareceriam no "meio" da trajectória, onde as velocidades atingem valores mais elevados. Deste modo, e na perspectiva de utilização destes cálculos num esquema de controlo, os erros referidos seriam de menor importância já que usualmente só se pretende uma grande precisão de posicionamento no início e no fim de cada movimento. Hollerbach [5.34] provou também ser errónea esta simplificação, já que o reescalonamento no tempo (i. e. fazendo  $t'=ct$ ,  $c$ -constante) de uma trajectória implica reduções/ampliações semelhantes nas amplitudes dos termos inerciais e Coriolis/centrípetos ( $[C(q,\dot{q})]'=c^2C(q,\dot{q})$ ,  $[J(q)\ddot{q}]'=c^2J(q)\ddot{q}$ ). Uma terceira técnica proposta [5.35] consistiu na substituição das computações em vírgula flutuante por computações em vírgula fixa. Se bem que mais rápida em tempo real, podem sobrevir problemas de precisão, pelo que esta técnica fica confinada a casos onde sejam aceitáveis as limitações referidas.

O método computacional apresentado neste capítulo beneficia do conhecimento adquirido com as diversas aproximações referidas ao longo desta introdução. Contudo o seu ponto de aplicação reside na optimização da fase de compilação do algoritmo escolhido. Na realidade, enquanto que as investigações referidas se dedicam à geração de algoritmos, cada vez mais eficientes, de

cálculo da dinâmica inversa, a fase seguinte correspondente à conversão do algoritmo para código máquina (compilação), tem sido algo esquecida. Segundo esta perspectiva, no próximo parágrafo os conceitos envolvidos são sistematizados e apresentados numa sequência que conduz à elaboração de um compilador (compilador dedicado) capaz de gerir muito eficientemente todos os recursos disponíveis. Tendo em vista uma melhor explanação desta técnica, ela é então aplicada (parágrafo 5.3) ao protótipo já usado (robot 2R). Uma observação atenta dos resultados conseguidos revela importantes propriedades, com implicações na concepção de novas arquitecturas computacionais paralelas, no desenvolvimento de processadores dedicados e na metodologia de modelização. Assim, no parágrafo 5.4 exploram-se os caminhos patenteados pelo compilador dedicado, e são apontadas algumas pistas para futuros melhoramentos. Finalmente, no parágrafo 5.5 é discutida toda a estratégia desenvolvida e são apresentadas as conclusões resultantes.

## 5.2. OPTIMIZAÇÃO DA GESTÃO DOS RECURSOS COMPUTACIONAIS: O COMPILADOR DEDICADO

### 5.2.1. Algumas considerações relativas à síntese do compilador dedicado

Conforme se pode constatar a partir da introdução, a redução do tempo de cálculo, em tempo real, tem vindo a passar pela sucessiva melhoria do algoritmo previamente gerado. Por outras palavras, tem-se aumentado o investimento computacional prévio (i. e. off-line), com vista à diminuição do tempo de cálculo em tempo real. Este investimento tem incidido na geração do algoritmo, contudo é lícito pensar-se no alargamento desta estratégia

a outros procedimentos também envolvidos. A fase de compilação, isto é, a conversão do algoritmo para código máquina, pode ser otimizada segundo esta perspectiva. De facto, relativamente à compilação pode-se dizer que:

- Usualmente, o algoritmo é convertido para o código interno de representação da informação no computador, através de um compilador de uso geral tal como Fortran ou C.

- O código resultante impõe ao microprocessador uma pesada carga computacional, devido ao elevado número de cálculos efectuados em vírgula flutuante.

- Computações em vírgula flutuante dão origem a extensas e complexas rotinas em código máquina.

- Estas computações devem ser efectuadas com alta precisão, de modo a serem evitados os erros numéricos associados a uma representação da informação por palavras com precisão finita.

- Cálculos em vírgula flutuante e com elevada precisão fazem apelo a uma representação da informação através de palavras de extensão considerável.

- Operações analíticas efectuadas no sistema de vírgula flutuante, originam um extenso e complexo código máquina consideravelmente pesado para o microprocessador.

Evidentemente que este tipo de implementação está longe de ser satisfatório, tanto mais que se podem tecer diversas considerações adicionais:

- A representação numérica, usada internamente pelo computador para tratamento da informação, tem muito maior exactidão do que a da electrónica exterior usada no manipulador (conversores A/D e D/A, sensores, etc), usualmente situada entre 8 e 16 bits

- Se o cálculo em tempo real for efectuado pelo computador

usando uma precisão semelhante à da electrónica externa, então este será significativamente mais rápido.

- No entanto, a representação da informação interna através de uma precisão similar à externa, sem o conseqüente aparecimento de erros numéricos, implica a adopção de um método de cálculo baseado noutros princípios, diferentes daqueles que estão associados à álgebra clássica.

- Um método alternativo para uma computação eficiente em tempo real, necessita de uma álgebra "especial", a qual acarretará uma gestão mais "inteligente" dos recursos computacionais existentes.

- O método alternativo de computação deverá produzir um código máquina simples e eficiente, isto é, bem adaptado ao conjunto de instruções do microprocessador.

Em conclusão, é necessário encontrar-se um novo compilador capaz de gerar um código máquina mais eficiente e melhor adaptado ao ambiente {computador + robot e respectiva electrónica}. Este compilador pode ser chamado de "compilador dedicado" em oposição aos "compiladores de uso geral" tais como Fortran ou C.

Uma candidata possível para a realização da álgebra "especial" atrás mencionada, é a álgebra de Boole já que obedece a todas as especificações desejadas. Todavia, agora depara-se o problema de converter os cálculos efectuados através da álgebra clássica, para um sistema de computação baseado na álgebra de Boole. Se bem que sejam álgebras formalmente independentes, existe um meio de realizar esta conversão (Fig. 5.1). O algoritmo a compilar consiste num conjunto de funções multivariáveis (que envolvem a avaliação de somas, multiplicações e, eventualmente,

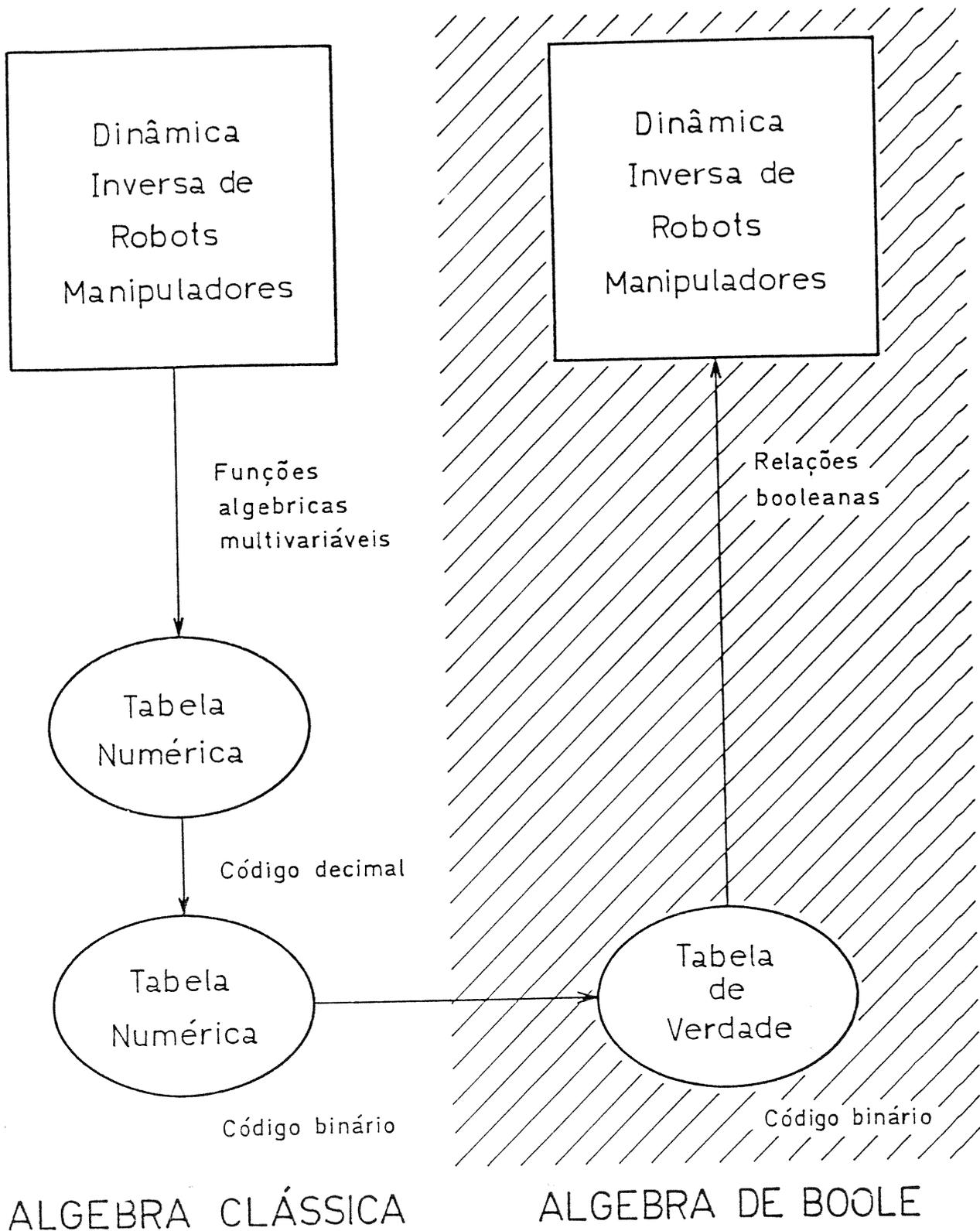


Fig. 5.1 Estratégia para a conversão da dinâmica inversa de manipuladores expressa na álgebra clássica, para uma forma segundo os conceitos da álgebra de Boole.

funções transcendentas), que como tal podem ser expressas na forma de uma tabela numérica multidimensional. Se, quer para as variáveis independentes, quer para as variáveis dependentes, os valores correspondentes à quantificação necessária ao preenchimento da tabela, forem convertidos para uma representação binária inteira, então a tabela resultante pode ser vista alternativamente, e sob o ponto de vista da álgebra de Boole, como uma tabela de verdade onde se podem aplicar as metodologias normalmente associadas à manipulação de informação booleana. Em particular, são de considerar variáveis de entrada os bits da informação (quantificada e expressa em código binário) de grandezas tais como a posição, velocidade e aceleração, e variáveis de saída os bits da informação (quantificada e expressa em código binário) correspondente aos binários requeridos por cada eixo.

Deve, desde já, referir-se que a consideração do conjunto de variáveis de entrada depende dos fins pretendidos. Assim, se o cálculo da dinâmica inversa for suposto para uma carga constante, e na ausência de forças externas (i. e. forças aplicadas ou sofridas pelo manipulador quando em contacto com o meio ambiente), então este conjunto deverá limitar-se a {posição, velocidade, aceleração}. Contudo, para cargas variantes ou para situações correspondentes à interacção com o meio externo, este conjunto deverá incluir parâmetros (quantificados e expressos em código binário) da carga (massa, inércia, etc) ou da referida interacção (forças ou binários externos), resultando um conjunto de variáveis de entrada do tipo {posição, velocidade, aceleração, parâmetros característicos da carga, forças/binários externos}. Como é evidente, e em certa medida à semelhança do que acontece no cálculo clássico, a segunda situação impõe uma maior complexi-

dade que corresponde ao preço da generalização do método a um maior número de situações. No seguimento deste trabalho adoptar-se-á, implicitamente, a primeira situação não só por ser mais simples, mas também por ser aquela que normalmente é considerada noutros trabalhos relativos ao cálculo da dinâmica inversa de manipuladores, sendo, contudo, a generalização ao segundo caso óbvia e imediata.

Prosseguindo o raciocínio relativo à concepção do compilador dedicado baseado na álgebra de Boole, podem-se listar as exigências (máximas) colocadas pela implementação das tabelas de verdade:

$$N_E = \sum_{k=1}^n \sum_{j=1}^l X_{jk} \quad (5.1a)$$

$$N_S = \sum_{i=1}^n Y_i \quad (5.1b)$$

onde  $N_E$  é o número total de bits correspondente às variáveis de entrada

$N_S$  é o número total de bits correspondente às variáveis de saída

$X_{jk}$  é o número total de bits usado na representação de cada variável booleana de entrada

$Y_i$  é o número total de bits usado na representação de cada variável booleana de saída

$n$  é o número de graus de liberdade do robot manipulador

$l$  é o número de variáveis de entrada; para o conjunto {posição, velocidade, aceleração} vem  $l=3$ .

Considerando a situação usual de se exigir uma precisão semelhante, quer nas variáveis de entrada, quer nas variáveis de saída, então vem

$$X_{jk} = Y_i = z \text{ bits, para quaisquer } i, j, k \quad (5.2)$$

resultando a necessidade de uma tabela de verdade que requer:

$$N_E = nlz \quad (5.3a)$$

$$N_s = nz \quad (5.3b)$$

Dado que as dimensões da tabela evoluem exponencialmente com  $N_E$  e linearmente com  $N_s$ , resulta uma tabela de verdade que necessita, teoricamente, de um número total de bits  $N_T$  dado pela expressão:

$$N_T = (N_E 2^{N_E}) + (N_s 2^{N_E}) = (N_E + N_s) 2^{N_E} \quad (5.4)$$

Como é evidente, esta situação ideal não é praticável já que coloca exigências extremamente pesadas devido às dimensões da tabela. A título de exemplo refira-se que para um robot com seis graus de liberdade ( $n=6$ ), sem a manipulação de cargas variantes ou quaisquer interacções com o ambiente externo ( $l=3$ ), e para uma precisão da electrónica de oito bits ( $Z=8$ ), segundo (5.4) seria necessária uma tabela de verdade com  $N_T \approx 4.3 \cdot 10^{45}$  bits.

Não obstante, esta metodologia sugere um ponto de partida para a obtenção de alternativas praticáveis e que suportem as propriedades exigidas ao compilador dedicado.

A resposta alternativa é conseguida através de um compromisso entre a optimização pretendida e a exequibilidade necessária. As computações pela álgebra clássica ou através da álgebra de Boole, podem ser consideradas como dois casos extremos, relativamente aos quais poderão conceber-se métodos de cálculo híbridos [5.36-5.39] e que, conseqüentemente, terão vantagens e inconvenientes dependentes do grau de utilização de cada um dos métodos. Desta maneira, o compromisso pretendido passa pela

coexistência de cálculos baseados na álgebra clássica e de cálculos baseados na álgebra de Boole. Neste trabalho estuda-se a viabilidade de dois métodos: computação híbrida a partir da soma aritmética e da álgebra de Boole, e computação híbrida baseada na soma e multiplicação aritméticas e na álgebra de Boole.

Como se verificará, as conclusões obtidas com o estudo dos dois métodos mencionados, são facilmente generalizáveis às demais possibilidades de cálculo híbrido pelo que são gerais e sistemáticas.

### 5.2.2. Computação híbrida usando a soma aritmética e a álgebra de Boole

Se se tomarem como ponto de partida para o desenvolvimento deste método, as expressões simbólicas da dinâmica inversa de robots manipuladores que, conforme foi visto em capítulos anteriores, são da forma:

$$T=J(q)\ddot{q}+C(q,\dot{q})+G(q) \quad (5.5)$$

onde  $J(q)$  é a matriz (simétrica) inercial, de dimensão  $n \times n$

$C(q,\dot{q})$  é o vector,  $n$  dimensional, dos binários/forças Coriolis/centrípetos

$G(q)$  é o vector,  $n$  dimensional, dos binários/forças gravitacionais.

então um processo de computação híbrida, consistirá no cálculo de cada componente de  $T$  (i. e. cada  $T_i$ ,  $i=1, \dots, n$ ), por intermédio de somas binárias e da álgebra booleana.

Para o efeito, a expressão simbólica do cálculo de  $T_i$  deverá sofrer o seguinte tratamento:

a) Separação das expressões simbólicas em diversas parcelas.

Esta acção de separação deve incidir nos pontos onde existe uma

operação de soma.

b) As diversas quantidades presentes são:

b1) Quantificadas de acordo com a precisão desejada

b2) Expressas num código binário inteiro apropriado

c) As sub-expressões correspondentes a cada parcela, são calculadas à luz dos princípios da álgebra de Boole, e de acordo com a filosofia referida no ponto 5.2.1.

d) As diversas parcelas (calculadas em c) são somadas através da soma aritmética binária.

Este procedimento torna-se mais claro se se reescrever a expressão (5.5) na forma:

$$T_i = \sum_{k=1}^n j_{ik}(q) \ddot{q}_k + \sum_{p=1}^n \sum_{k=1}^n c_{ipk}(q) \dot{q}_p \dot{q}_k + g_i(q) \quad (5.6)$$

onde  $j_{ik}(q)$  é a componente de ordem  $i,k$  da matriz  $J(q)$

$c_{ipk}(q) \dot{q}_p \dot{q}_k$  é uma das  $n^2$  parcelas existentes na componente  $i$  do vector  $C(q, \dot{q})$

$g_i(q)$  é a componente  $i$  do vector  $G(q)$

Como é evidente, o cálculo de  $T_i$  apresenta dois tipos de computações: avaliação das parcelas  $j_{ik}(q)\ddot{q}_k$ ,  $c_{ipk}(q)\dot{q}_p\dot{q}_k$ ,  $g_i(q)$  e realização das respectivas somas; assim, é natural a alocação destas duas computações às duas álgebras em consideração, isto é à álgebra booleana e à álgebra clássica, respectivamente. Devem ainda notar-se os seguintes pontos:

- Esta distribuição dos cálculos necessários pelas duas álgebras em consideração, pressupõe um tratamento adequado da informação envolvida, nomeadamente quantificação segundo a precisão requerida e posterior conversão para um código binário inteiro.

- Cálculo das somas através da álgebra clássica, mas usando quantidades inteiras expressas em código binário.

- Cálculo através da álgebra de Boole, das parcelas  $j_{ik}(q) \ddot{q}_k$ ,  $c_{ipk}(q) \dot{q}_p \dot{q}_k$ ,  $g_i(q)$ , expressas no código binário referido; todavia, a parcela  $g_i(q)$  pode decompor-se, se se entender necessário, em diversas sub-parcelas da forma

$$g_i(q) = \sum_{k=1}^i g_{ik}^*(q) \quad (5.7)$$

sendo  $g_i(q)=0$  quando a estrutura do robot leva a que o actuador do eixo  $i$  não suporte efeitos gravitacionais (por exemplo, para um eixo rotacional vertical, onde a gravidade é suportada pelos rolamentos).

- O uso da operação soma introduz uma certa degradação da precisão, pelo que se torna necessária a utilização de palavras com um maior número de bits. Este número adicional de bits ( $m$ ), necessário devido à utilização da soma aritmética, vem dado pela fórmula:

$$m = \text{INT}[\log_2(p+1)] \quad (5.8)$$

onde  $p$  representa o número total de parcelas. Não obstante, o crescimento de  $m$  com  $p$  é relativamente baixo, pelo que não acarreta problemas significativos. Em consequência da degradação da precisão mencionada, quer as somas aritméticas, quer a avaliação booleana das diversas parcelas, deverá ser efectuada com uma precisão de  $\hat{z}=z+m$  bits, para se obterem  $z$  bits significativos em  $T_i$  ( $i=1, \dots, n$ ).

- Relativamente ao cálculo ideal, totalmente baseado na álgebra de Boole, e que requereria uma tabela de verdade gigantesca, a computação por este método híbrido necessita para

cada  $T_i$  até um máximo de

$$n \text{ parcelas inerciais} \quad (5.9a)$$

$$n(n+1)/2 \text{ parcelas Coriolis/centrípetas} \quad (5.9b)$$

$$i \text{ parcelas gravitacionais} \quad (5.9c)$$

Sabendo que, quer os termos inerciais, quer os termos Coriolis/centrípetos não dependem de  $q_1$ , então o número de variáveis de entrada (v. e.), requeridas pelas diversas tabelas, pode ascender até um máximo de

$$n \text{ v. e. para os termos do tipo } j_{ik}(q)\ddot{q}_k \quad (5.10a)$$

$$n+1 \text{ v. e. para o termos do tipo } c_{ipk}(q)\dot{q}_p\dot{q}_k, (p \neq k) \quad (5.10b)$$

$$n \text{ v. e. para o termos do tipo } c_{ipp}(q)\dot{q}_p^2 \quad (5.10c)$$

$$n \text{ v. e. para termos do tipo } g_i(q) \quad (5.10d)$$

Pode-se concluir que este método híbrido atinge os fins pretendidos (exequibilidade e optimização do cálculo em tempo real), pois não necessita da enorme tabela de verdade exigida pelo método de cálculo totalmente booleano mantendo, no entanto, características semelhantes no tocante ao bom desempenho em tempo real. De facto, ao invés da situação ideal agora sobrevem um maior número de tabelas de verdade, sendo, todavia, cada uma delas de dimensões substancialmente inferiores. Além disso, o tratamento booleano (i. e. compilação) pode ser efectuado separadamente para cada uma destas tabelas; neste caso, as maiores exigências dimensionais em termos da memória necessária para o armazenamento do "código fonte" (i. e. armazenamento da tabela de verdade) postas por este método híbrido, correspondem às exigências da maior tabela de verdade presente nas parcelas  $j_{ik}(q)\ddot{q}_k$ ,  $c_{ipk}(q)\dot{q}_p\dot{q}_k$ ,  $g_i(q)$ . O "preço" desta redução das exigências -

necessidade de  $m$  bits adicionais, que compensam a degradação da precisão introduzida pela soma aritmética - é de influência pouco significativa, não só pelo pequeno aumento das dimensões das tabelas, mas também pelo facto da eficiência da avaliação, em tempo real, ter modificações de pequena monta, já que existe uma compensação de efeitos. A computação ideal baseada numa única tabela de verdade (5.3) e com  $N_s$  expressões booleanas complexas dependentes de  $N_e$  bits de entrada, equilibra-se com a situação presente onde aparecem um maior número de expressões booleanas, todavia consideravelmente mais simples.

Finalmente é de mencionar que, quer os requisitos durante a "compilação" (off-line), quer o desempenho conseguido na computação (on-line), podem ser sobremaneira optimizados aplicando a cada caso, em particular, um conjunto de regras a definir no parágrafo 5.2.4. Verificar-se-á que estas regras são extremamente úteis e eficazes no presente método híbrido, mas de muito menor aplicabilidade no caso do "compilador ideal"; desta maneira, cava-se um fosso ainda maior entre a relação exigências/desempenho conseguidas pelos processos ideal e híbrido.

### 5.2.3. Computação híbrida usando as soma e multiplicação aritméticas, e a álgebra de Boole

Este método é consequência directa dos conceitos apresentados no parágrafo anterior. Com efeito, observando mais uma vez a fórmula (5.5) ou, alternativamente, a expressão (5.6), verifica-se que estão envolvidos outros tipos de avaliação, nomeadamente multiplicações. Deste modo parece razoável estender a filosofia subjacente ao método híbrido anterior, a uma metodologia onde a partição das expressões se efectue não somente nas somas, mas

também nas multiplicações.

Nesta perspectiva o número de variáveis de entrada (v. e.) exigidas pelas diversas tabelas de verdade, correspondentes à avaliação dos coeficientes  $j_{ik}(q)$ ,  $c_{ipk}(q)$ ,  $g_i(q)$ , reduzem-se a

$$n-1 \text{ v. e. para os coeficientes do tipo } j_{ik}(q) \quad (5.11a)$$

$$n-1 \text{ v. e. para os coeficientes do tipo } c_{ipk}(q) \quad (5.11b)$$

$$n \text{ v. e. para os coeficientes do tipo } g_{ik}(q) \quad (5.11c)$$

Contudo, esta modificação de influências, aliviando as computações booleanas por troca com cálculos aritméticos mais pesados, não acarreta implicações tão claras e eficientes como no caso anterior. Efectivamente, se bem que à primeira vista exista uma certa redução nas dimensões das tabelas, esse efeito pode ser ultrapassado pela necessidade de um muito maior número ( $m$ ) de bits adicionais que compensem a degradação da precisão introduzida não só pela soma (binária), mas principalmente pela multiplicação (binária).

Atendendo aos argumentos apresentados, entendeu-se ser questionável, e mesmo prejudicial, a passagem do método apresentado no ponto anterior, para outras computações híbridas baseadas num maior conjunto de avaliações segundo a álgebra clássica. Consequentemente, todo o estudo desenvolvido em seguida é dedicado única e exclusivamente ao compilador apresentado no ponto 5.2.2.

#### 5.2.4. Procedimentos para a optimização do método híbrido

Neste ponto é desenvolvido um conjunto de regras e procedimentos capazes de aliviar substancialmente as exigências manifestadas pelo sistema de compilação dedicada. Assim, podem-se

distribuir estas regras por dois grandes conjuntos - Conjunto de Regras Estratégicas e Conjunto de Regras Táticas - que em termos gerais têm a seguinte caracterização:

- Conjunto de Regras Estratégicas (R.E.)

As R.E. devem ser estudadas e adoptadas desde o início da compilação, sendo impossível a recuperação (i. e. compensação) de uma estratégia sub-ótima, excepto com a reinicialização de toda uma nova compilação baseada na estratégia mais apropriada. A escolha da estratégia óptima faz apelo a procedimentos "inteligentes", já que assenta em regras fortemente interdependentes e com acções "cruzadas", onde o equilíbrio óptimo dos respectivos efeitos é complexo. Ainda assim, estes procedimentos podem e devem ser programados convenientemente, e incorporados no sistema de compilação.

- Conjunto de Regras Táticas (R.T.)

Em princípio, as R.T. devem ser utilizadas posteriormente às R.E. mas antes da compilação propriamente dita; todavia, devido à interdependência de efeitos das diversas regras, quer R.E., quer R.T., podem ocorrer situações que impliquem uma ou mais reorganizações da estrutura de procedimentos adoptada. O uso destas regras, provoca uma diminuição das exigências colocadas no "arranque" da compilação, isto é, nas dimensões das tabelas de verdade necessárias. Ao contrário do que acontece com as R.E., o esquecimento ou a adopção de R.T. em opções sub-óptimas, é compensado automaticamente pelo compilador, se bem que exigindo maiores tabelas de verdade, que gera sempre o mesmo código final.

Passando a expôr as diversas regras vem:

## REGRAS ESTRATÉGICAS

### R.E.1. Adopção de um código binário apropriado

As relações booleanas obtidas a partir das tabelas de verdade, têm forte dependência dos códigos utilizados na representação das variáveis envolvidas. Verifica-se que:

a) Normalmente as variáveis  $q$ ,  $\dot{q}$  e  $\ddot{q}$  têm amplitudes com excursões simétricas relativamente ao valor zero, o que origina excursões igualmente simétricas dos termos  $j_{ik}(q)\ddot{q}_k$ ,  $c_{ipk}(q)\dot{q}_p\dot{q}_k$  e  $g_i(q)$ .

b) As variáveis  $q$  influenciam estes termos através de funções sinusóidais o que, para eixos rotacionais, implica a existência de uma periodicidade entre os valores de  $q$  e os valores dos coeficientes  $j_{ik}(q)$ ,  $c_{ipk}(q)$  e  $g_i(q)$ .

c) Todas as funções envolvidas têm variações "suaves".

O código binário "ideal" deve reflectir as propriedades a), b) e c), e diversas experiências efectuadas confirmam estes argumentos. De facto, a opção mais eficiente na descrição das variáveis de entrada e de saída, mostrou ser a do código de Gray que é:

a) Simétrico relativamente à origem, a menos do bit mais significativo;

b) Periódico;

c) Com variações mínimas (e suaves) entre dois estados consecutivos.

### R.E.2. Avaliação booleana através de Diagramas de Decisão Binária

Até ao momento tem-se referido a aplicação da álgebra de Boole para o cálculo de cada parcela. Pode pois parecer

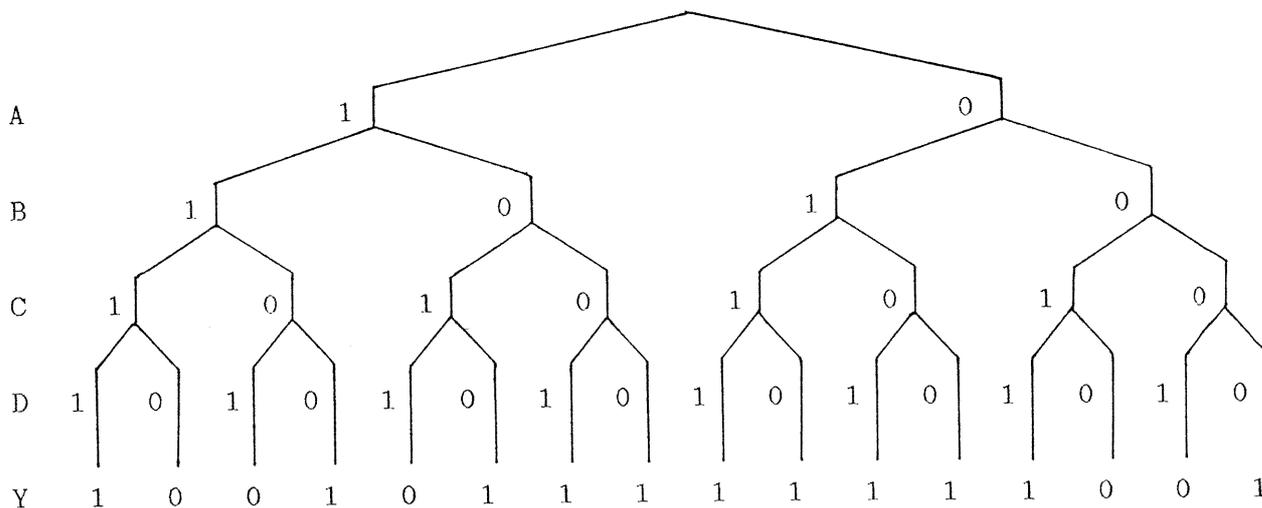
imediate a interpretação do termo "cálculo booleano" como a utilização de funções booleanas de forma a avaliarem-se as relações obtidas à luz da álgebra de Boole. Todavia, quer as exigências subjacentes ao tratamento das tabelas de verdade, quer o desempenho obtido na avaliação booleana em tempo real, podem ser reduzidos de uma forma notável mediante a adopção de "formas" mais eficientes de cálculo booleano. De facto, a avaliação através de funções booleanas dá origem a desempenhos menos eficientes, quer na fase de compilação, quer na fase de cálculo em tempo real. Na compilação, o tratamento das tabelas de verdade implica o uso de métodos de simplificação computacionalmente muito pesados (por exemplo o método de Quine-McCluskey [5.40, 5.41]). No cálculo em tempo real, as funções de Boole são também extremamente complexas, devido ao aparecimento de expressões que reflectem as inúmeras combinações possíveis dos bits de entrada, através das operações de Boole {E, OU, NÃO}. Em consequência, devem ser examinadas alternativas a este processo de representação da álgebra de Boole. Uma "forma" alternativa mais eficiente, e que foi escolhida neste trabalho, toma o nome de Diagramas de Decisão Binária (DDB) [5.42, 5.43]. Este método é, obviamente, baseado nos princípios da álgebra de Boole, reduzindo contudo de um modo significativo os problemas referidos (exigências de compilação e rapidez de cálculo em tempo real). Os DDB correspondem, conceptualmente, a estruturas em árvore binária e que são muito facilmente simplificadas e avaliadas num computador. A figura 5.2 mostra, a título exemplificativo, as realizações referidas para uma relação booleana da forma  $Y=f(A,B,C,D)$ . Como se pode observar, a simplificação de um DDB resume-se à eliminação de ramos vizinhos (i. e. ramos que correspondem aos valores 0

A	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
B	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
C	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
D	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Y	1	0	0	1	0	1	1	1	1	1	1	1	1	0	0	1

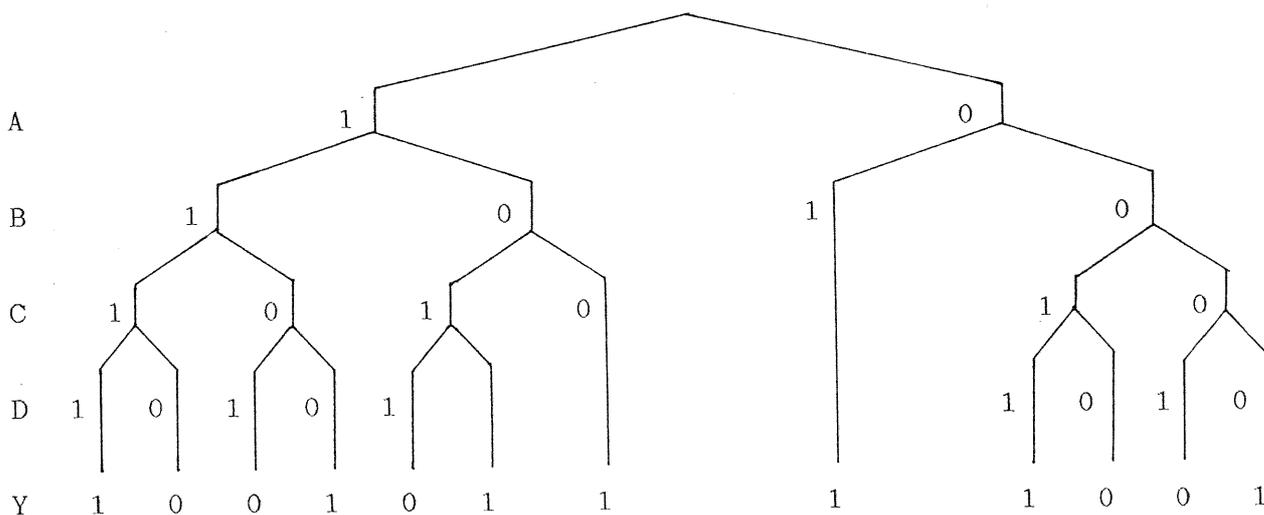
a)

$$Y = \bar{C} \cdot \bar{D} + \bar{A} \cdot B + A \cdot \bar{B} \cdot \bar{D} + A \cdot \bar{B} \cdot \bar{C} + \bar{A} \cdot C \cdot D + B \cdot C \cdot D$$

b)



c)



d)

Fig. 5.2 Formas de representação de relações booleanas.

- a) Tabela de verdade
- b) Função booleana
- c) DDB não simplificado
- d) DDB simplificado

e 1 de uma mesma variável de entrada), e o número de avaliações booleanas, no cálculo em tempo real, fica limitado, no máximo, ao número de bits de entrada.

### R.E.3. Avaliação booleana em cascata/paralelo

O cálculo booleano dos termos  $j_{ik}(q)\bar{q}_k$ ,  $c_{ipk}(q)\bar{q}_k$  e  $g_i(q)$ , pode estar baseado numa só ou, alternativamente, em diversas tabelas de verdade. O caso da estruturação das relações binárias em várias tabelas toma o nome de avaliação em cascata/paralelo. Assim, particularmente para tabelas de verdade com um grande número de variáveis de entrada, as exigências relativas ao seu tratamento podem ser reduzidas de uma forma notável, se se proceder a uma reestruturação dessa (única) tabela em diversas sub-tabelas mais pequenas, cada uma com um menor número de variáveis de entrada. O resultado final é conseguido através do reagrupamento apropriado dos diversos ramos da nova estrutura. Para uma maior clareza deste procedimento de reestruturação, examinam-se dois exemplos muito simples de funções do tipo  $Y=f(A,B,C)$ , para uma igual precisão ( $z$ ) de todas as variáveis, e onde  $\{*, +, \bar{\quad}\} = \{E, \text{OU}, \text{NÃO}\}$  e  $\{0, 1\} = \{\text{FALSO}, \text{VERDADEIRO}\}$ . As figuras 5.3 e 5.4 mostram estruturas simples e em cascata/paralelo para relações do tipo  $Y=A*B*C$  e  $Y=A*B+C$ , respectivamente. Como resulta óbvio, a avaliação em cascata/paralelo representada nas Figs. 5.3b) e 5.4b) têm as seguintes propriedades:

- Podem, eventualmente, existir múltiplas estruturas em cascata/paralelo, com uma ou mais soluções óptimas, e que dependem do tipo de relação booleana. Por exemplo, na figura 5.3b) existem três estruturas cascata/paralelo alternativas,

enquanto que na figura 5.4b) só existe uma única estrutura.

- Podem não existir soluções em cascata/paralelo alternativas à estrutura baseada numa só tabela de verdade.

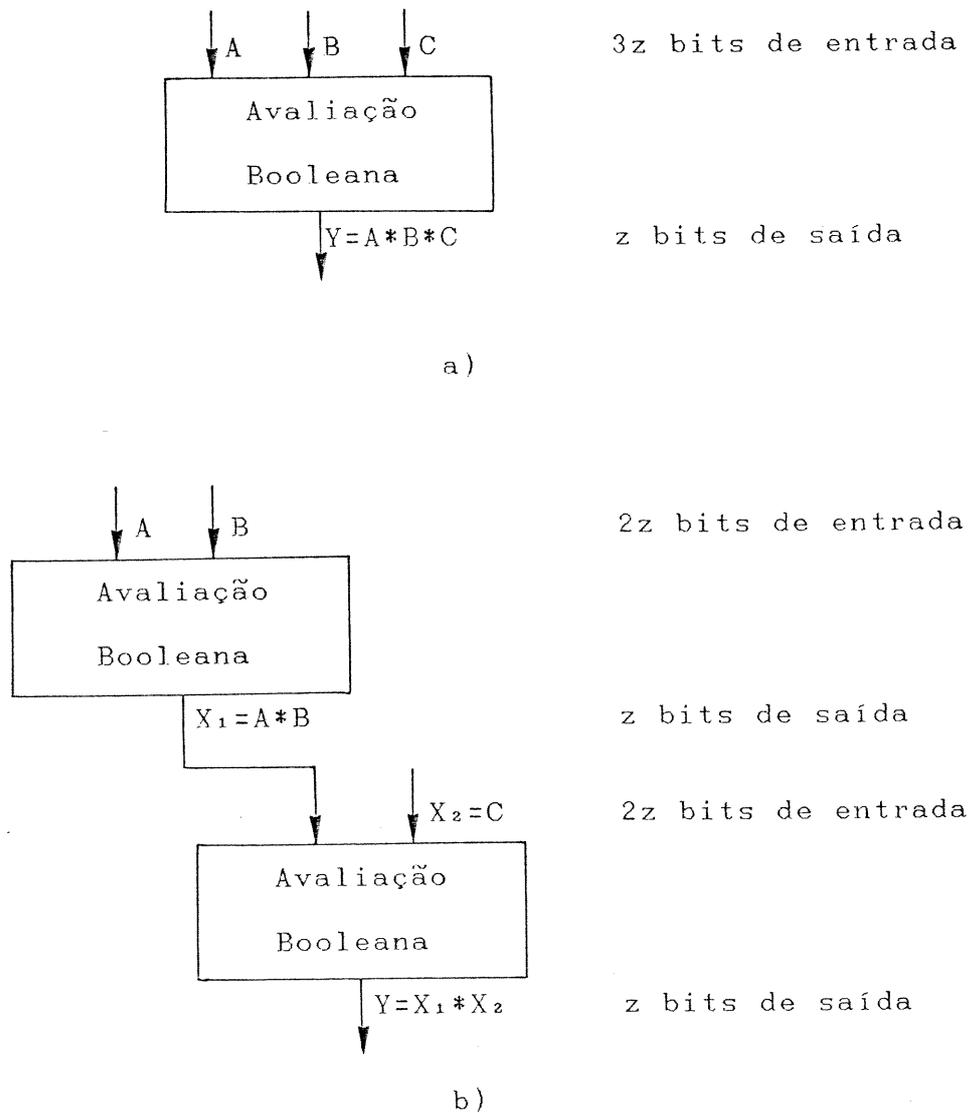


Fig. 5.3 Estruturação do cálculo da relação booleana  $Y=A*B*C$   
Cada variável tem z bits de precisão.

- a) A partir de uma única tabela exigindo 3z bits de entrada.
- b) Usando uma estrutura cascata/paralelo com duas sub-tabelas exigindo cada 2z bits de entrada.  
Estrutura da forma  $Y=X_1*X_2$  com  $X_1=A*B$  e  $X_2=C$ .  
Outras estruturas alternativas:  $X_1=A*C$ ,  $X_2=B$ ,  
 $X_1=B*C$ ,  $X_2=A$ .

- A reestruturação em diversas sub-tabelas encadeadas em cascata/paralelo fica condicionada a determinados tipos de sequenciação, que são impostos pelo tipo de relação booleana. Por exemplo, na figura 5.3b)  $X_1$  é sempre calculado antes de  $Y=X_1*X_2$ , enquanto que na figura 5.4b)  $X_1$  é sempre calculado antes de  $Y=X_1+C$ .

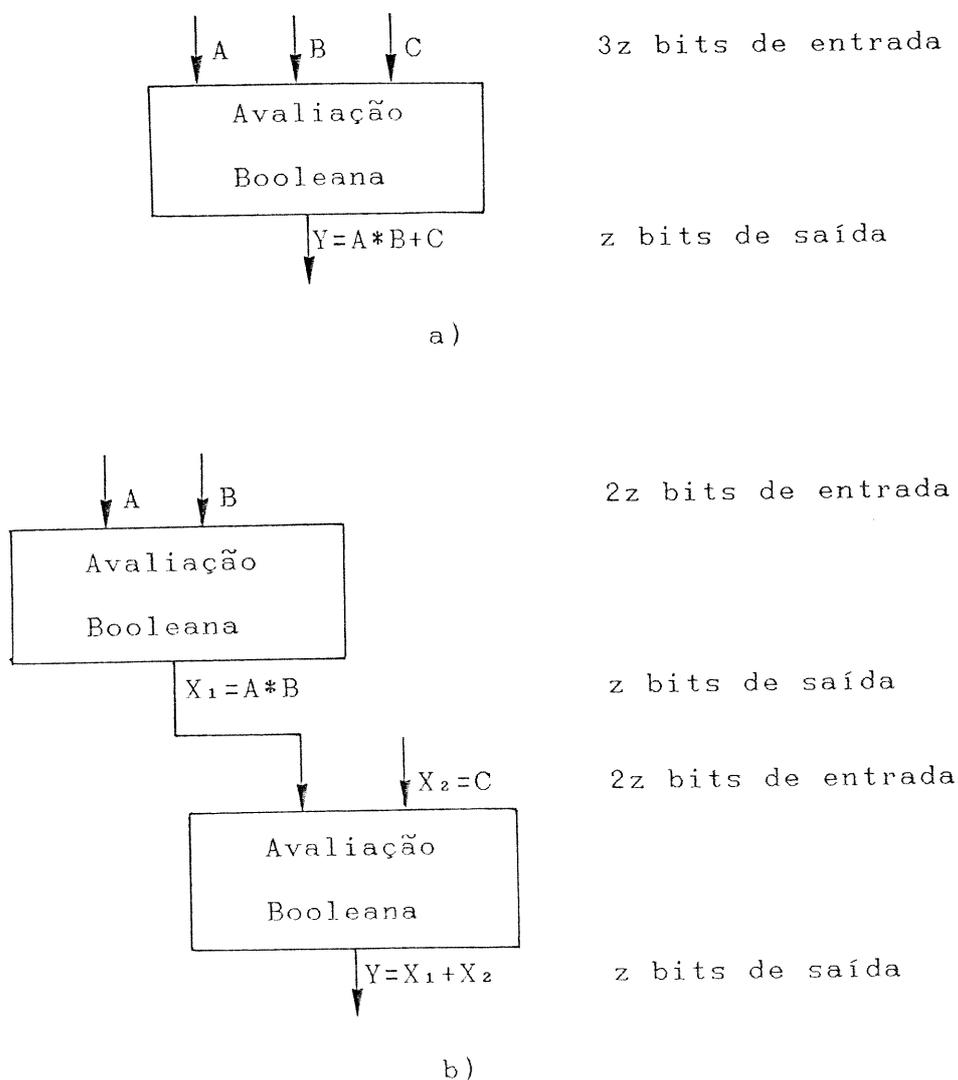


Fig. 5.4 Estruturação do cálculo da relação booleana  $Y=A*B+C$ . Cada variável tem  $z$  bits de precisão.

a) A partir de uma única tabela exigindo  $3z$  bits de entrada.

b) Usando uma estrutura cascata/paralelo com duas sub-tabelas exigindo cada  $2z$  bits de entrada. Estrutura da forma  $Y=X_1+X_2$  que possibilita uma só concretização:  $X_1=A*B$ ,  $X_2=C$ .

- A eficiência do código final, correspondente ao encadeamento apropriado das diversas sub-tabelas (i. e. correspondente à estrutura cascata/paralelo), depende do equilíbrio conseguido entre influências diversas. As sub-tabelas com menores dimensões acarretam avaliações mais simples mas, em contrapartida, a avaliação final, correspondente ao encadeamento das diversas sub-avaliações, pode ser mais complexa. O equilíbrio entre as duas influências depende de cada caso em particular.

R.E.4. Escolha do número óptimo de somas aritméticas

A expressão (5.8) que indica o número  $m$  de bits adicionais necessários devido à existência de  $p$  parcelas, varia de uma forma descontínua. Esta observação torna-se mais evidente quando se tabela a função (5.8)

Tabela 5.1 Função  $m = \text{INT}[\log_2(p+1)]$

$p$	1	2	3	...	6	7	...	14	15	...	30	31	...
$m$	1	1	2	...	2	3	...	3	4	...	4	5	...

Isto quer dizer que existe um grande número de situações onde o valor obtido para  $m$  possibilita a "ampliação" do número de somas  $p$  até um número máximo  $p_m$ , segundo a sequência da Tabela 5.2

Tabela 5.2 Número máximo de somas  $p_m = f(p)$

$p$	1	2	3	...	6	7	...	14	15	...	30	31	...
$m$	1	1	2	...	2	3	...	3	4	...	4	5	...
$p_m$	1	1	6	...	6	14	...	14	30	...	30	62	...

Esta propriedade leva a que sejam de considerar dois tipos de situações que tiram proveito do número  $p_m-p$  de "novas" somas possíveis:

#### R.E.4.1 Partição de uma parcela em $2^u$ parcelas

Se existirem  $p_m-p$  somas disponíveis, então podem reduzir-se as exigências colocadas às dimensões das tabelas, mediante a partição de uma ou mais parcelas iniciais, em várias sub-parcelas. Se uma parcela dá origem a  $2^u$  ( $u \in \mathbb{N}$ ) sub-parcelas então, cada uma delas terá amplitudes máximas  $2^u$  vezes menores, tornando-se assim prescindível o uso dos  $u$  bits mais significativos, quer das variáveis de entrada, quer das variáveis de saída.

#### R.E.4.2 Partição/reunião de parcelas gravitacionais

A utilização da R.E.2. para o caso dos termos gravitacionais pode originar situações algo complexas. De acordo com a expressão (5.7), cada termo gravitacional  $g_i(q)$  ( $i=1, \dots, n$ ) pode ser decomposto até um máximo de  $i$  sub-parcelas. Esta eventualidade pode ser encarada em ambos os sentidos (partição ou reunião de diversas sub-parcelas). De facto, o aumento do número de parcelas do valor  $p$  até a  $p_m$  favorece certas regras em desfavor de outras, e vice-versa para a situação oposta. A opção óptima corresponde a um equilíbrio das acções das diversas regras, e é encontrada analisando todos as possibilidades cruzadas.

#### R.E.5. Precisão das variáveis

Para cada parcela deve verificar-se uma precisão das variáveis de entrada, semelhante à precisão desejada para a variável de saída. Este princípio leva a que para termos com um elevado número de variáveis de entrada, a aplicação de regras tendentes a

simplificar a sua compilação acarrete um efeito muito mais pronunciado.

## REGRAS TÁCTICAS

### R.T.1. Aproveitamento dos estados de quantificação não usados

Na maior parte das vezes, os níveis máximos de quantificação têm valores consideravelmente superiores aos níveis máximos da amplitude de cada parcela em consideração (Fig. 5.5). Existem dois tipos de procedimento que visam tirar proveito deste "excesso" de níveis de quantificação:

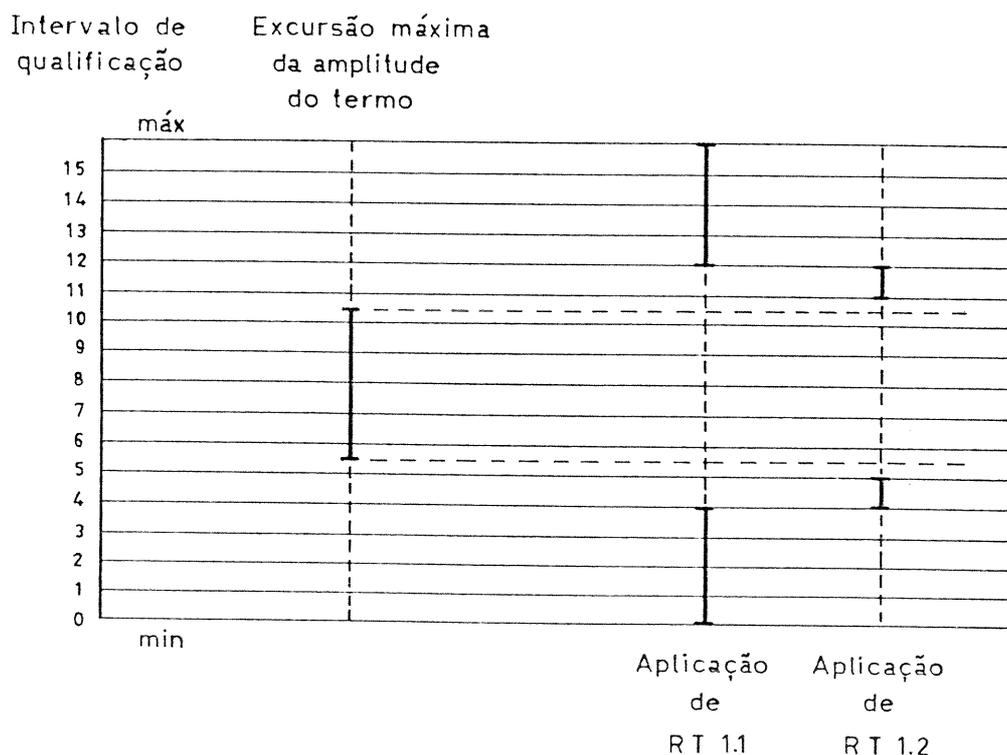


Fig. 5.5 Aproveitamento dos níveis de quantificação não usados.

#### R.T.1.1 Redução da amplitude do intervalo de quantificação por um factor de $2^v$

Se for possível reduzir a amplitude do intervalo de quantificação por um factor de  $2^v$  ( $v \in \mathbb{N}$ ), mantendo-se sempre o seu nível máximo igual ou superior ao da amplitude do termo, então pode-se prescindir dos  $v$  bits mais significativos de todas as variáveis em jogo. Note-se que esta acção tem fortes semelhanças ao procedimento referido em R.E.4.; todavia, neste caso a simplificação pode ser conseguida pelo compilador, de um modo automático, aquando da fase de simplificação da tabela de verdade.

#### R.T.1.2 Aproveitamento dos estados residuais correspondentes a níveis de quantificação superiores à amplitude máxima da parcela

Na maior parte das vezes, e mesmo após a eventual aplicação de outras regras de simplificação, o nível máximo de quantificação é ainda algo superior à amplitude máxima da tabela a simplificar. O compilador (i. e. o sistema de simplificação das tabelas de verdade) pode tirar partido dos estados de quantificação "não usados", tornando os respectivos bits das variáveis de saída como optativos (i. e. podendo tomar os valores 0 ou 1 consoante seja mais favorável para a simplificação booleana). Assim, as expressões booleanas finais serão mais simples e fáceis de calcular. Por outro lado, deve ter-se, mais uma vez, em conta que o compilador é capaz de realizar esta optimização de um modo automático.

#### R.T.2. Eliminação do bit mais significativo das variáveis para funções pares expressas segundo o código de Gray

Para funções pares com uma só variável de entrada, tais como

$y=\cos(x)$  ou  $y=x^2$ , expressas no código de Gray, verificou-se que se pode prescindir da utilização do bit mais significativo de cada palavra. Esta propriedade é consequência da simetria quer da função, quer do código de Gray. Mais uma vez, no caso do "esquecimento" desta regra, o sistema de compilação encarrega-se, automaticamente, de fazer a respectiva simplificação.

As interações entre as diversas regras anteriormente mencionadas, estão representadas na Fig. 5.6.

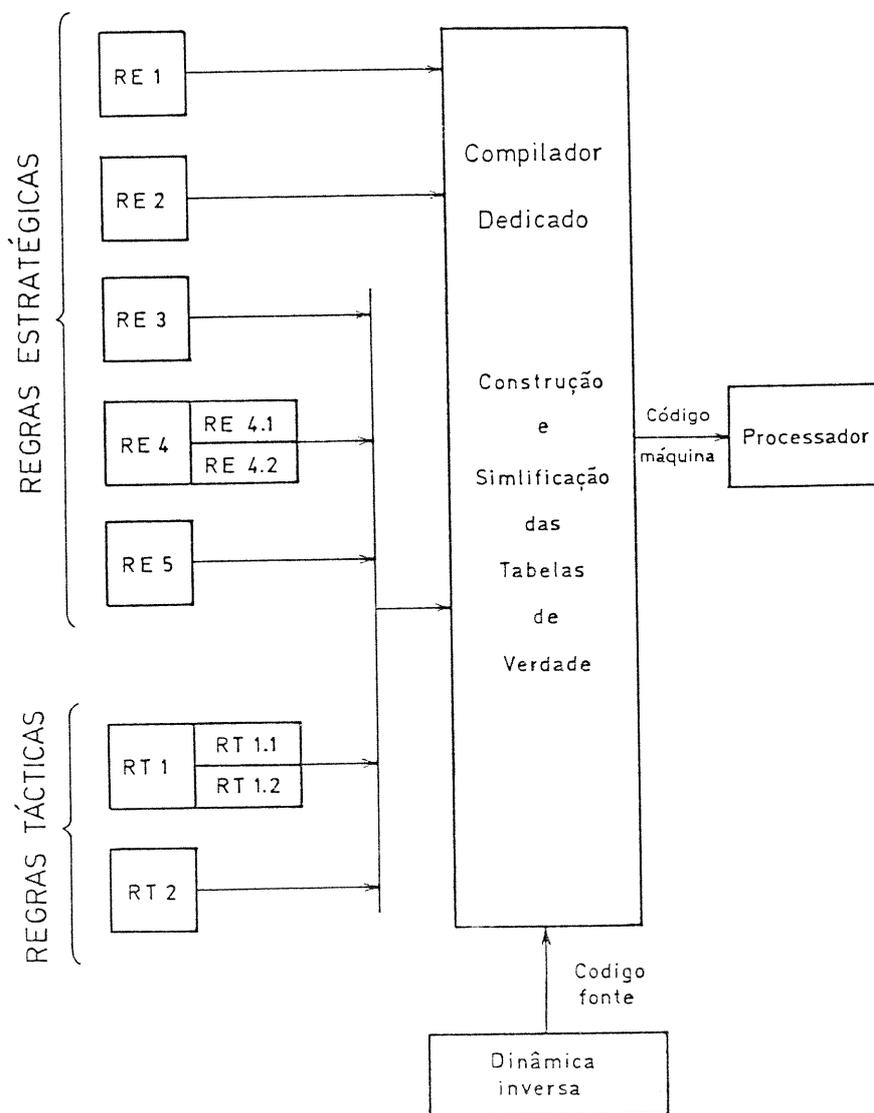


Fig. 5.6 Interações entre as R. E. e as R. T.

### 5.3. IMPLEMENTAÇÃO DO COMPILADOR DEDICADO NO ROBOT 2R

Para ilustrar as características e o desempenho conseguidos com o sistema de compilação dedicada, neste parágrafo estes conceitos são aplicados ao cálculo da dinâmica inversa do robot 2R. Assim, tomando como ponto de partida as expressões da dinâmica inversa deste robot, tem-se:

$$J(q) = \begin{bmatrix} m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1 & m_2 (r_2^2 + r_1 r_2 C_2) \\ m_2 (r_2^2 + r_1 r_2 C_2) & m_2 r_2^2 + J_2 \end{bmatrix} \quad (5.12a)$$

$$C(q, \dot{q}) = \begin{bmatrix} -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 - m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ m_2 r_1 r_2 S_2 \dot{q}_1^2 \end{bmatrix} \quad (5.12b)$$

$$G(q) = \begin{bmatrix} g[m_1 r_1 C_1 + m_2 (r_2 C_{12} + r_1 C_1)] \\ g m_2 r_2 C_{12} \end{bmatrix} \quad (5.12c)$$

onde  $C_1 = \cos(q_1)$ ,  $C_2 = \cos(q_2)$ ,  $C_{12} = \cos(q_1 + q_2)$  e  $S_2 = \sin(q_2)$ .

Consideram-se, também, os valores numéricos utilizados nos capítulos anteriores:

$$m_1 = 0.5 \text{ Kg}, \quad m_2 = 6.25 \text{ Kg}, \quad r_1 = 1 \text{ m}, \quad r_2 = 0.8 \text{ m} \quad (5.13)$$

$$J_1 = 5 \text{ Kgm}, \quad J_2 = 5 \text{ Kgm}$$

Assumindo como "razoáveis" os intervalos de quantificação das variáveis de entrada ( $i=1,2$ ):

$$-\pi \text{ rad} \leq q_i \leq \pi \text{ rad} \quad (5.14a)$$

$$-1 \text{ rad/s} \leq \dot{q}_i \leq 1 \text{ rad/s} \quad (5.14b)$$

$$-1 \text{ rad/s}^2 \leq \ddot{q}_i \leq 1 \text{ rad/s}^2 \quad (5.14c)$$

onde "razoável" é entendido como estando dentro dos valores

verificados pela experiência. Então, as variáveis de saída têm amplitudes situadas nos intervalos:

$$-151.5 \text{ Nm} \leq T_1 \leq 151.5 \text{ Nm}, \quad -69.1 \text{ Nm} \leq T_2 \leq 69.1 \text{ Nm} \quad (5.15)$$

Pressupondo que o cálculo dos binários é correspondente a um mero bloco inserido numa arquitectura de controlo mais vasta, onde a influência de outros blocos pode levar à existência de valores mais elevados para  $T_1$  e  $T_2$ , consideraram-se maiores intervalos de quantificação das variáveis de saída, nomeadamente:

$$-200 \text{ Nm} \leq T_1 \leq 200 \text{ Nm}, \quad -100 \text{ Nm} \leq T_2 \leq 100 \text{ Nm} \quad (5.16)$$

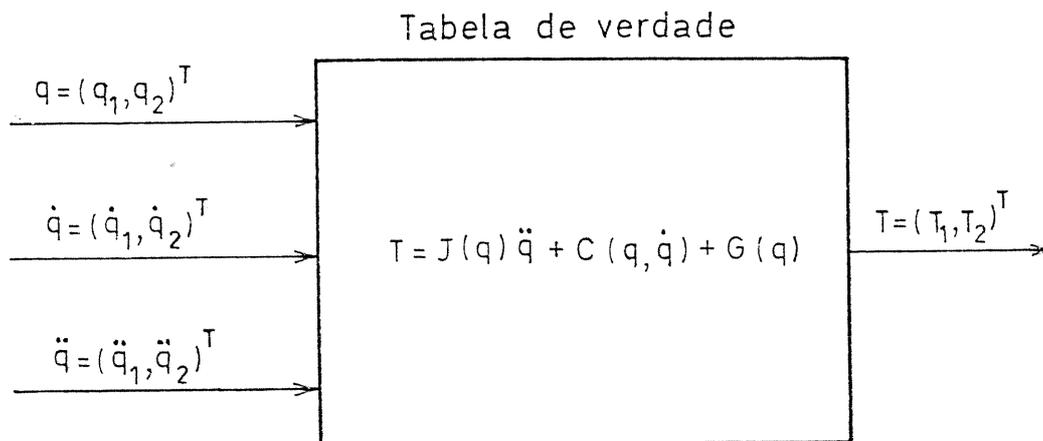


Fig. 5.7 Exigências da tabela de verdade requerida pelo compilador ideal.

Precisão desejada:  $z=8$  bits

Número extra de bits de precisão:  $m=0$

Precisão requerida a cada variável:  $\hat{z}=8$  bits

Intervalo de quantificação de  $T_1$ :  $-200 \text{ Nm} \leq T_1 \leq 200 \text{ Nm}$

Intervalo de variação de  $T_1$ :  $-151.5 \text{ Nm} \leq T_1 \leq 151.5 \text{ Nm}$

Intervalo de quantificação de  $T_2$ :  $-100 \text{ Nm} \leq T_2 \leq 100 \text{ Nm}$

Intervalo de variação de  $T_2$ :  $-69.1 \text{ Nm} \leq T_2 \leq 69.1 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_1, q_2, \dot{q}_1, \dot{q}_2, \ddot{q}_1, \ddot{q}_2\}$ :  $48 \cdot 2^{4 \cdot 8}$

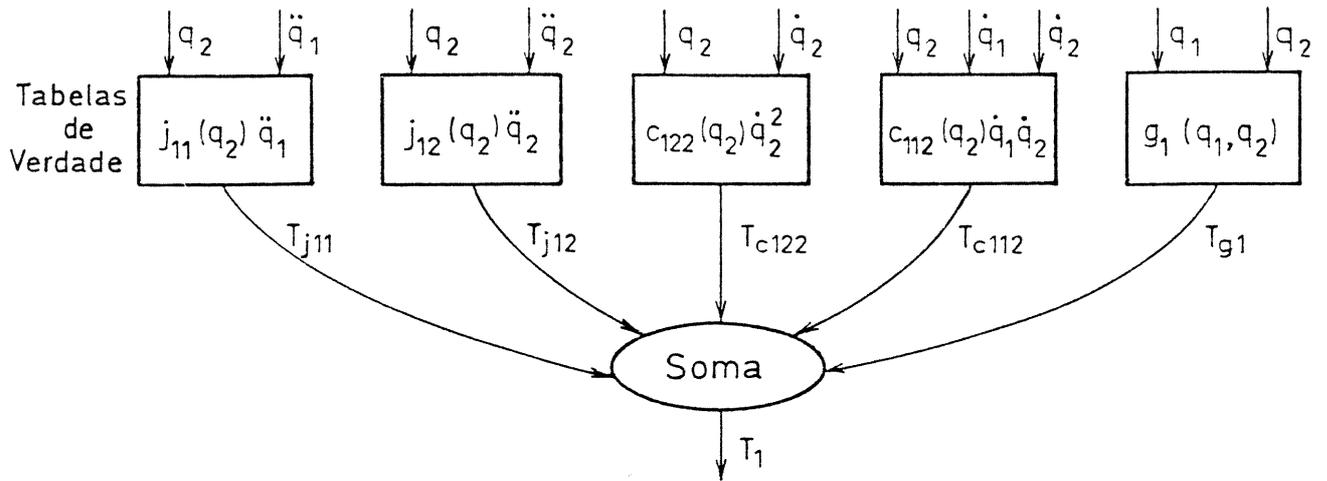
variáveis de saída  $\{T_1, T_2\}$ :  $16 \cdot 2^{4 \cdot 8}$

total:  $64 \cdot 2^{4 \cdot 8}$

Uma vez estabelecidos os intervalos de quantificação das variáveis de entrada (5.14) e das variáveis de saída (5.16), pode-se prosseguir para as fases de construção e simplificação das tabelas de verdade.

A utilização de uma única tabela de verdade para cada variável de saída, ou seja a aplicação do compilador na sua forma ideal, acarreta as exigências representadas na Fig. 5.7. Para uma utilização do método híbrido sem, todavia, se recorrer às R.E. e R.T. as tabelas de verdade e respectivas exigências estão representadas na Fig. 5.8 (ver Apêndice 1). Por último, tendo em conta as R.E. e as R.T. as Figs. 5.9 e 5.10 (ver Apêndices 2 e 3) mostram duas possíveis estruturas de computação, que diferem entre si somente pela adopção em  $T_1$  da R.E.3 (avaliação booleana em cascata/paralelo) na parcela Coriolis  $-2r_1r_2m_1S_2\dot{q}_1\dot{q}_2$ . Dado que, em termos de exemplificação com o robot 2R, este detalhe é de menor importância, entendeu-se como mais interessante atentar-se no desempenho conseguido pelo novo sistema de compilação. Assim, tomando os termos  $a=(m_2r_2^2+m_2r_1r_2C_2)\ddot{q}_2$ ,  $b=-m_2r_1r_2S_2\dot{q}_2^2$ ,  $c=g(m_1+m_2)r_1C_1$  presentes em  $T_1$  e  $d=(m_2r_2^2+J_2)\ddot{q}_2$ ,  $e=m_2r_1r_2S_2\dot{q}_1^2$  presentes em  $T_2$ , compararam-se os tempos de cálculo quando efectuados segundo o método clássico e através do compilador dedicado. Ambos os códigos estão escritos em Turbo Pascal V4.0 e correm numa máquina 8086, 8 MHz, sob o sistema operativo MSDOS V3.2. É evidente que se pode optar por outra linguagem; especialmente para o caso do novo compilador, a conversão para código máquina, pode efectuar-se sem a existência da fase "intermédia" de representação em Pascal ou noutra linguagem com características semelhantes. Não obstante, achou-se conveniente a comparação

Estruturação do método híbrido em  $T_1$



Estruturação do método híbrido em  $T_2$

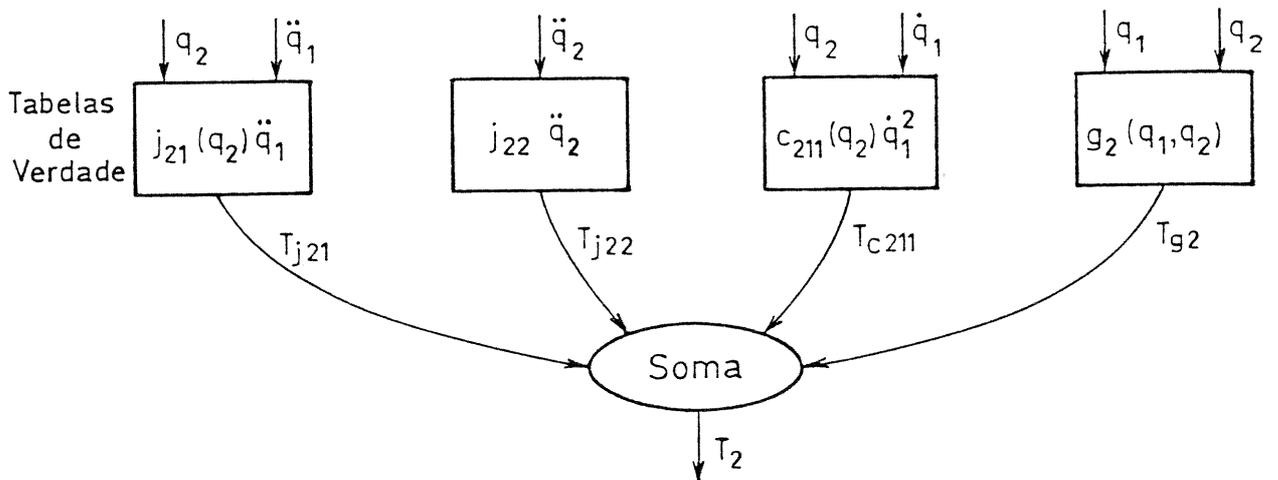
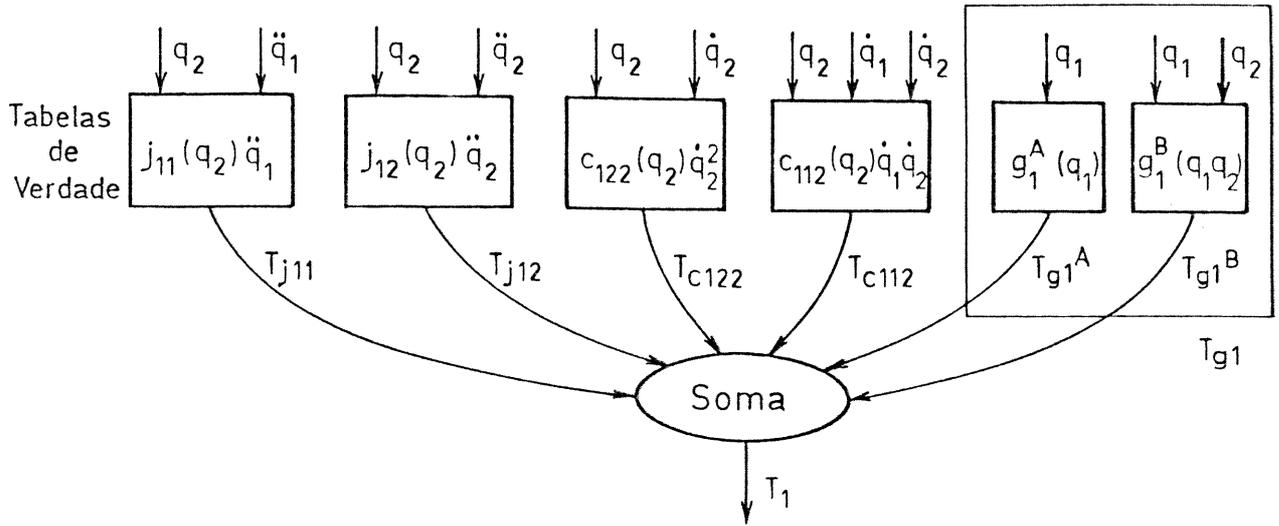


Fig. 5.8 Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, sem a aplicação das R.E. e R.T. ( $z=8$ ,  $m=2$ ,  $\hat{z}=10$ ).

$$\begin{cases} T_{j11} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\ T_{j12} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\ T_{c122} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ T_{c112} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 \\ T_{g1} = g [m_1 r_1 C_1 + m_2 (r_2 C_{12} + r_1 C_1)] \\ T_{j21} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \\ T_{j22} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\ T_{c211} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\ T_{g2} = g m_2 r_2 C_{12} \end{cases}$$

Estruturação do método híbrido em  $T_1$



Estruturação do método híbrido em  $T_2$

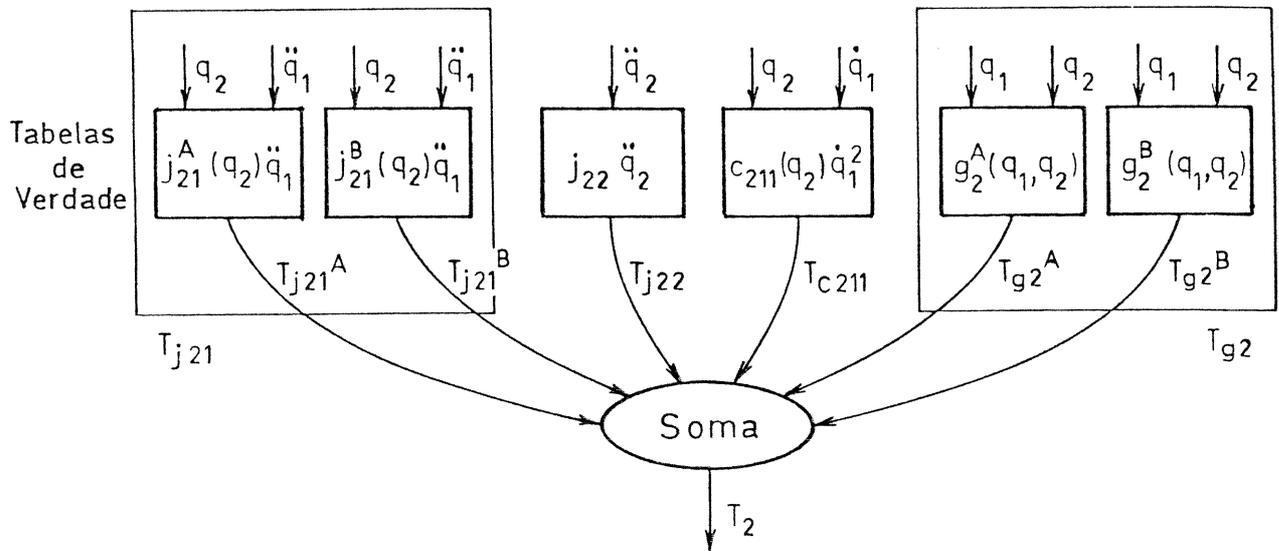


Fig. 5.9 Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, com a aplicação das R.E. e R.T., excepto a R.E.3 ( $z=8, m=2, \hat{z}=10$ ).

$$\left\{ \begin{array}{l}
 T_{j11} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\
 T_{j12} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\
 T_{c122} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\
 T_{c112} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 \\
 \left\{ \begin{array}{l}
 T_{g1}^A = g(m_1 + m_2) r_1 C_1 \\
 T_{g1}^B = g m_2 r_2 C_{12} \\
 T_{g1} = T_{g1}^A + T_{g1}^B
 \end{array} \right. \\
 T_{j21}^A = T_{j21}^B = \{ [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \} / 2 \\
 T_{j21} = T_{j21}^A + T_{j21}^B \\
 T_{j22} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\
 T_{c211} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\
 \left\{ \begin{array}{l}
 T_{g2}^A = T_{g2}^B = (g m_2 r_2 C_{12}) / 2 \\
 T_{g2} = T_{g2}^A + T_{g2}^B
 \end{array} \right.
 \end{array} \right.$$

Estruturação do método híbrido em  $T_1$

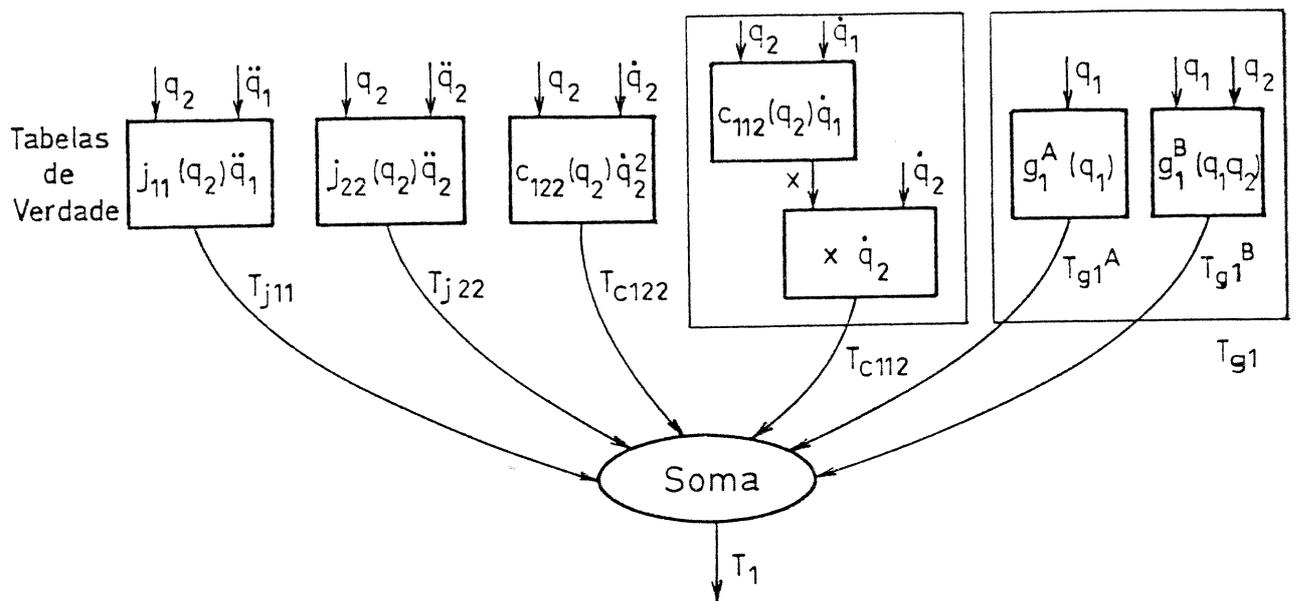


Fig. 5.10 Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, com a aplicação das R.E. e R.T., incluindo a R.E.3 ( $z=8, m=2, \hat{z}=10$ ).

$$\left\{ \begin{array}{l}
 T_{j_{11}} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\
 T_{j_{12}} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\
 T_{c_{122}} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\
 X = -2m_2 r_1 r_2 S_2 \dot{q}_1 \\
 T_{c_{112}} = X \dot{q}_2 \\
 \left\{ \begin{array}{l}
 T_{g_{1A}} = g(m_1 + m_2) r_1 C_1 \\
 T_{g_{1B}} = g m_2 r_2 C_{12} \\
 T_{g_{1}} = T_{g_{1A}} + T_{g_{1B}}
 \end{array} \right. \\
 T_{j_{21A}} = T_{j_{21B}} = \{ [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \} / 2 \\
 T_{j_{21}} = T_{j_{21A}} + T_{j_{21B}} \\
 T_{j_{22}} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\
 T_{c_{211}} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\
 \left\{ \begin{array}{l}
 T_{g_{2A}} = T_{g_{2B}} = (g m_2 r_2 C_{12}) / 2 \\
 T_{g_{2}} = T_{g_{2A}} + T_{g_{2B}}
 \end{array} \right.
 \end{array} \right.$$

dos dois métodos de compilação a partir de uma plataforma comum, de modo a serem claramente estabelecidas as respectivas características. Métodos de implementação do compilador dedicado são mais eficientes, mas não proporcionam uma tão grande clareza na apreciação dos resultados, pelo que se entendeu necessária uma comparação em "condições semelhantes".

A Fig. 5.11 mostra os histogramas dos tempos de cálculo necessários a cada um dos termos a, b, c, d, e, usando os dois métodos referidos, enquanto que na Fig. 5.12 se comparam os termos entre si. Como se conclui destas figuras, mesmo para uma situação desfavorável relativamente ao novo compilador, este é capaz de gerar um código máquina notavelmente mais rápido. Além disso, a Fig. 5.12 mostra, ainda, uma outra característica importante:

- Enquanto que para o termo d, com uma expressão analítica extremamente simples, os tempos de cálculo são semelhantes, para os demais termos a, b, c, e, com expressões mais complexas, isso não se verifica já que o tempo de cálculo com o código gerado pelo novo compilador é consideravelmente inferior ao tempo obtido para o método convencional. Desta maneira, a "sensibilidade" do tempo de cálculo face à complexidade das expressões analíticas é muito menor no caso do compilador dedicado. Esta característica é particularmente importante, pois indica que para robots com um maior número de graus de liberdade, onde se encontram expressões analíticas mais complexas, a diferença de "sensibilidades" acarretará uma diferença entre os dois desempenhos ainda mais pronunciada.

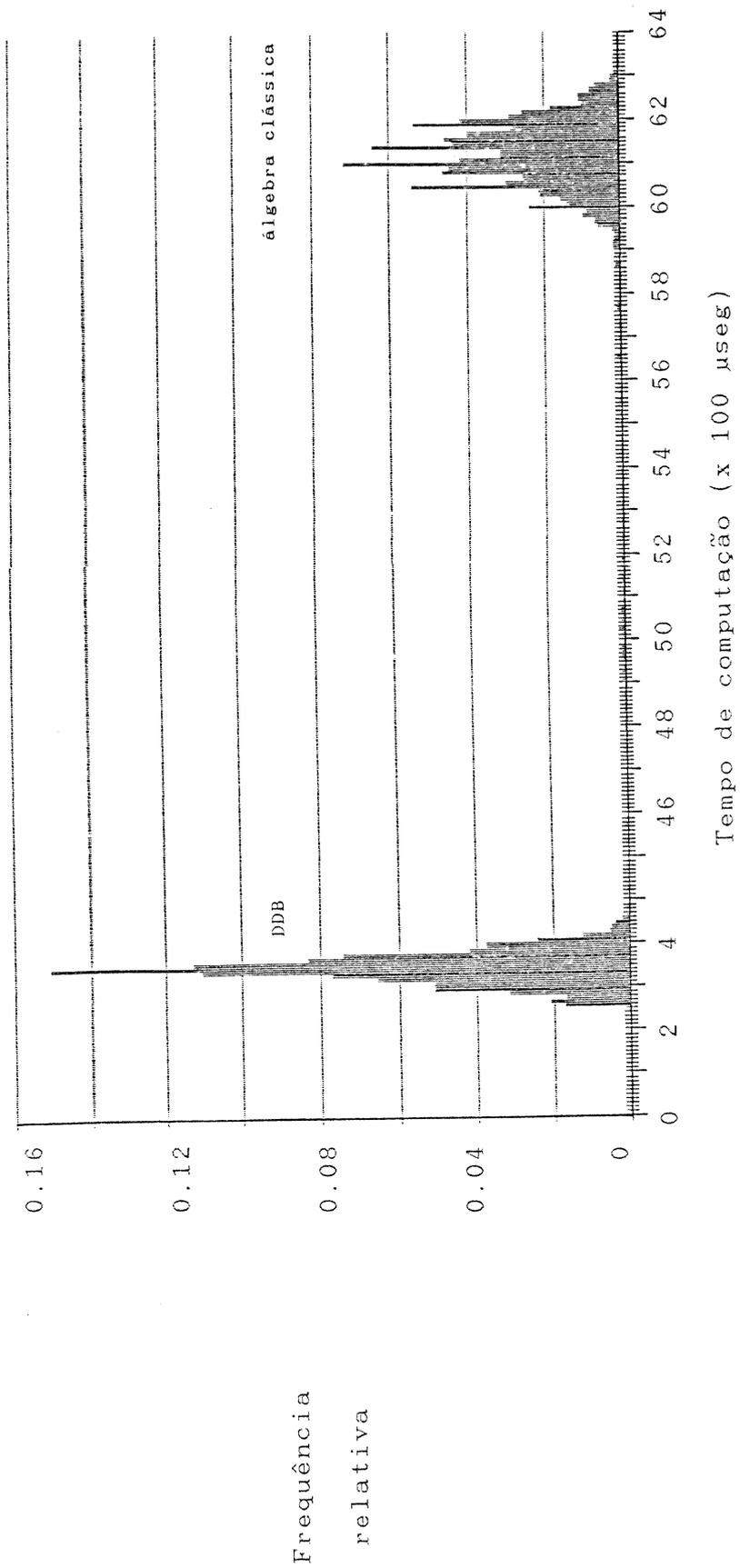


Fig. 5.11a) Histograma dos tempos necessários à computação do termo  $a=(m_2r_2^2+m_2r_1r_2C_2)q_2$  presente em T<sub>1</sub>, usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2.

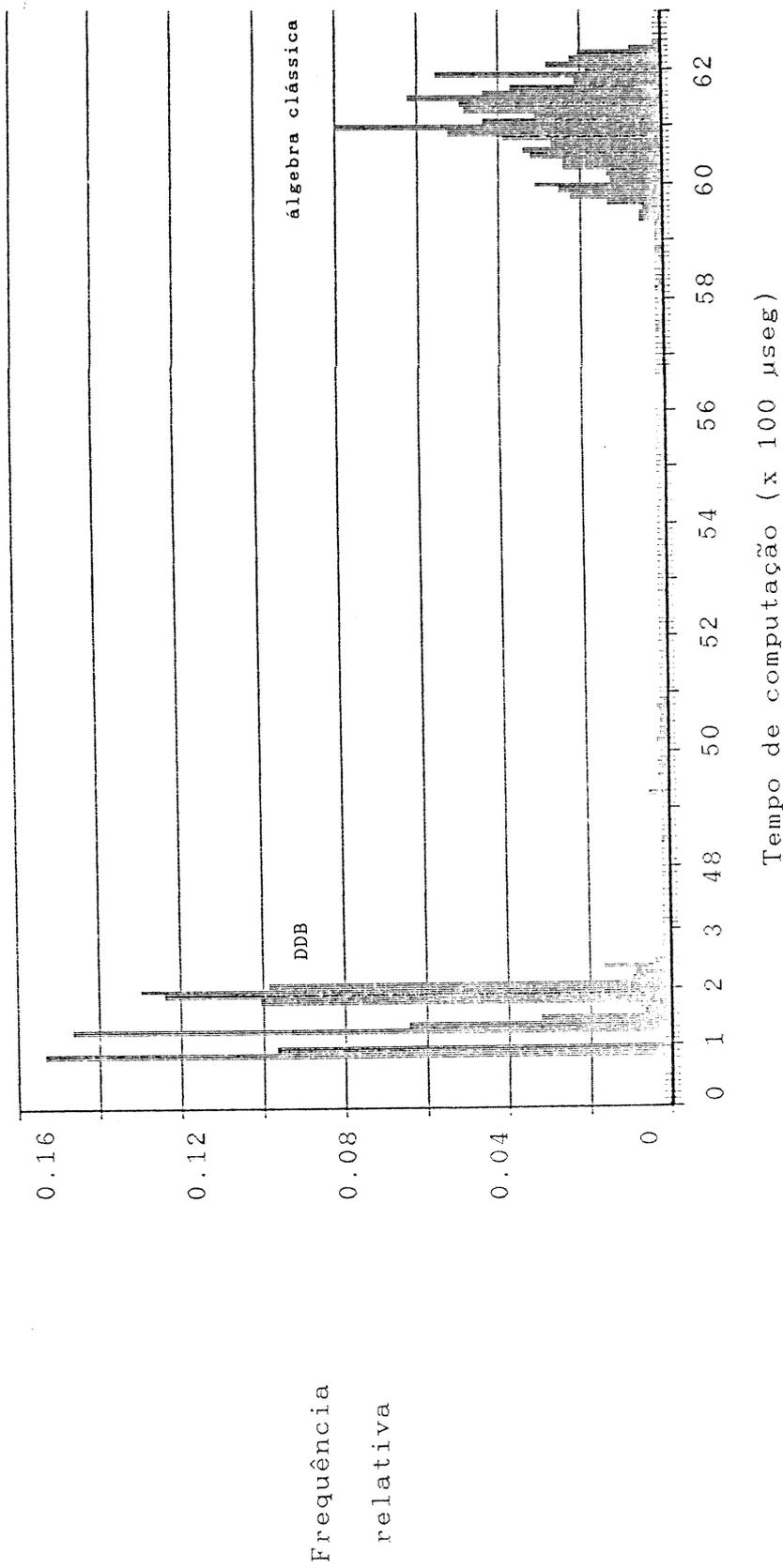


Fig. 5.11b) Histograma dos tempos necessários à computação do termo  $b = m_2 r_1 r_2 S_2 \dot{q}_2^2$  presente em  $T_1$ , usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2.

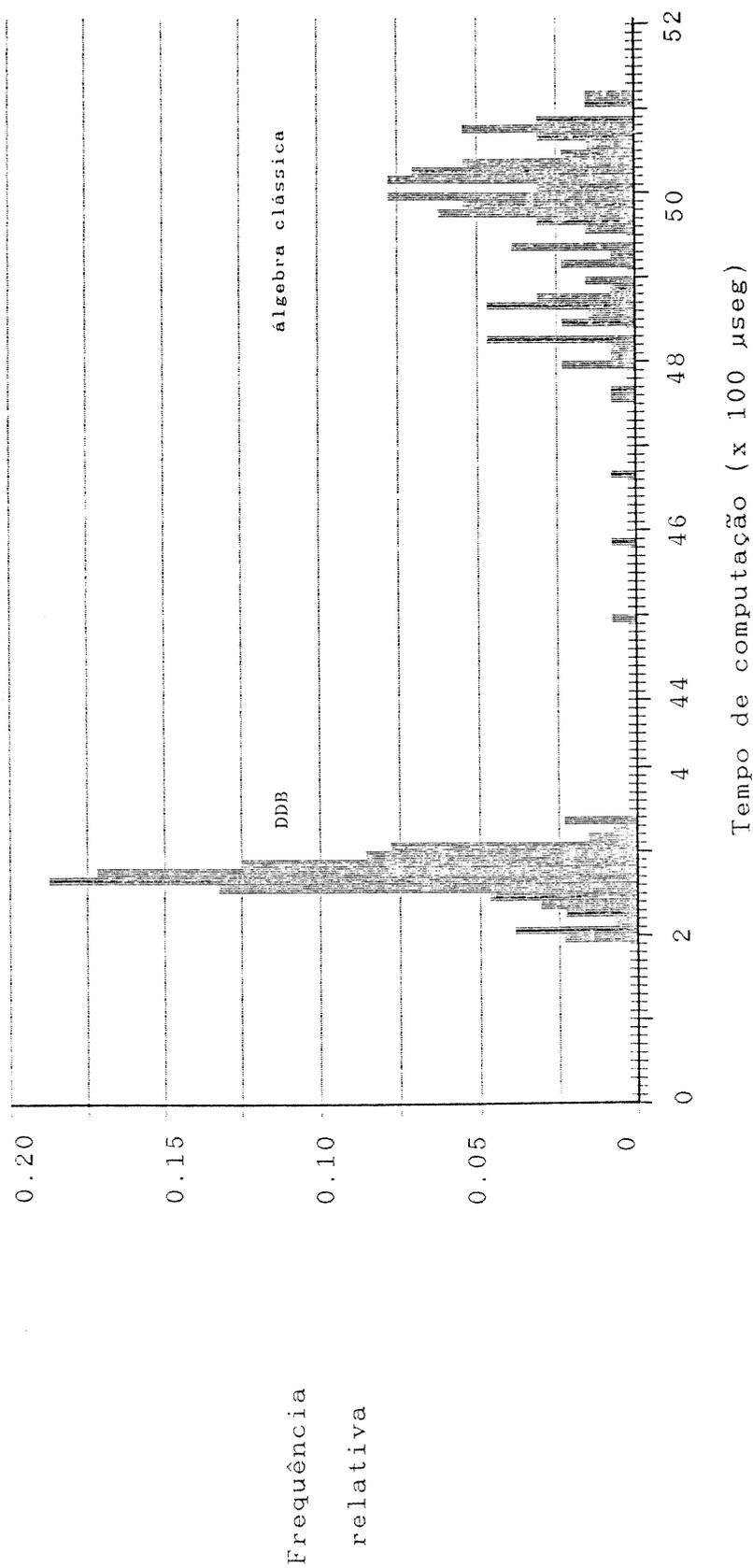


Fig. 5.11c) Histograma dos tempos necessários à computação do termo  $c=g(m_1+m_2)+r_1C_1$ , presente em T<sub>1</sub>, usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2.

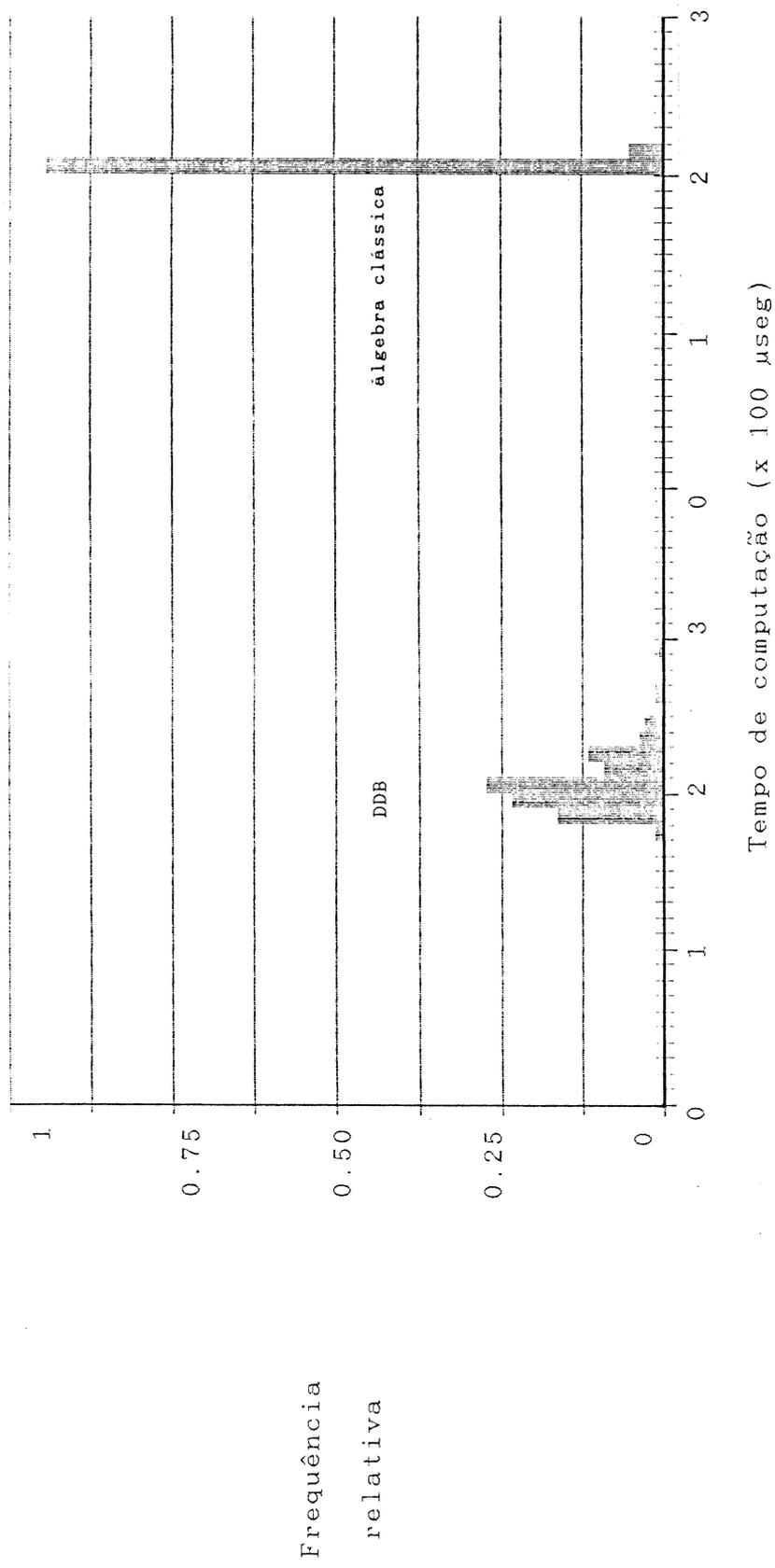


Fig. 5.11d) Histograma dos tempos necessários à computação do termo  $d=(m_2r_2^2+J_2)\ddot{\alpha}_2$  presente em  $T_2$ , usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2.

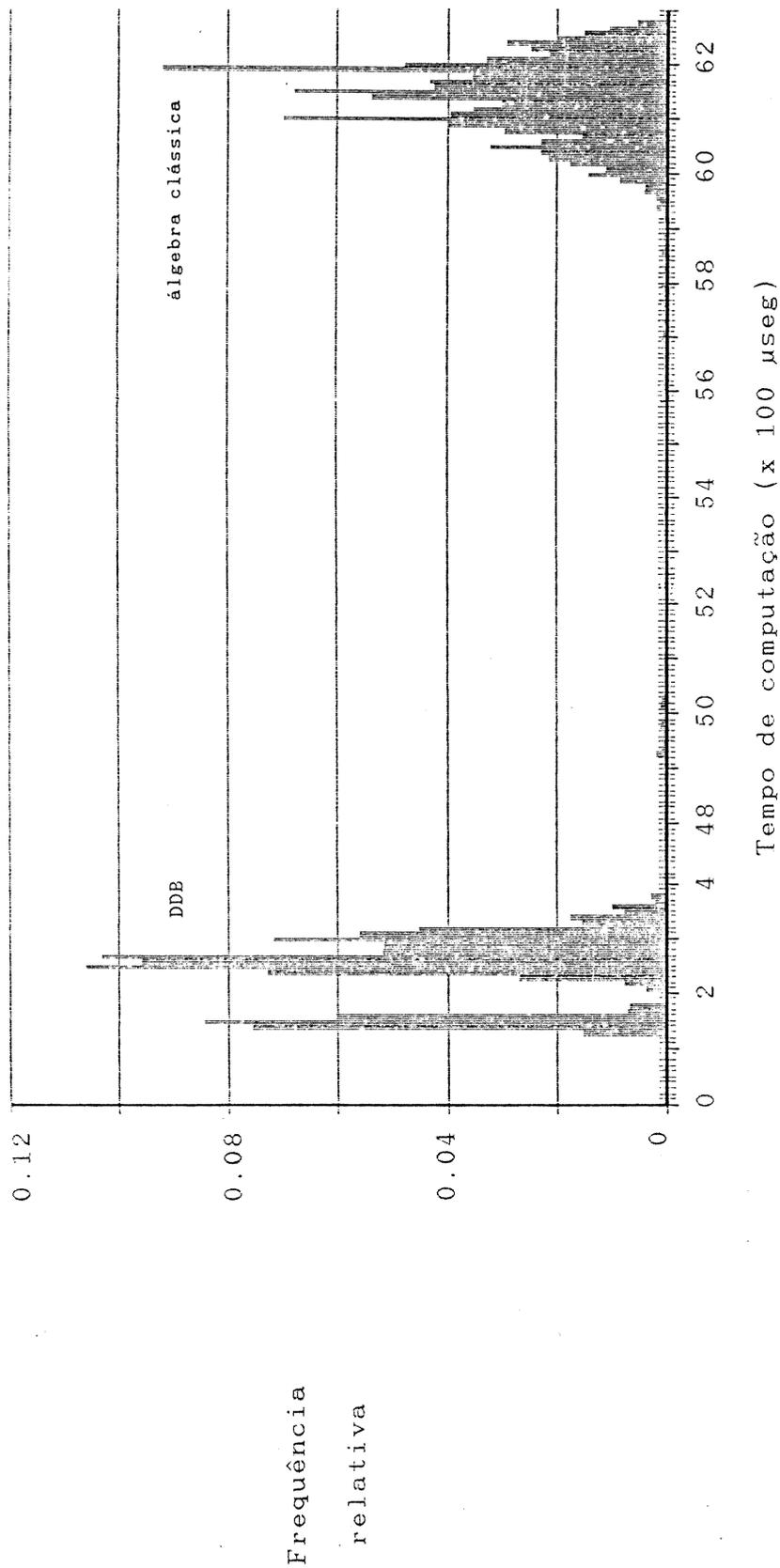


Fig. 5.11e). Histograma dos tempos necessários à computação do termo  $e = m_2 r_1 r_2 S_2 q_1^2$  presente em  $T_2$ , usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2.

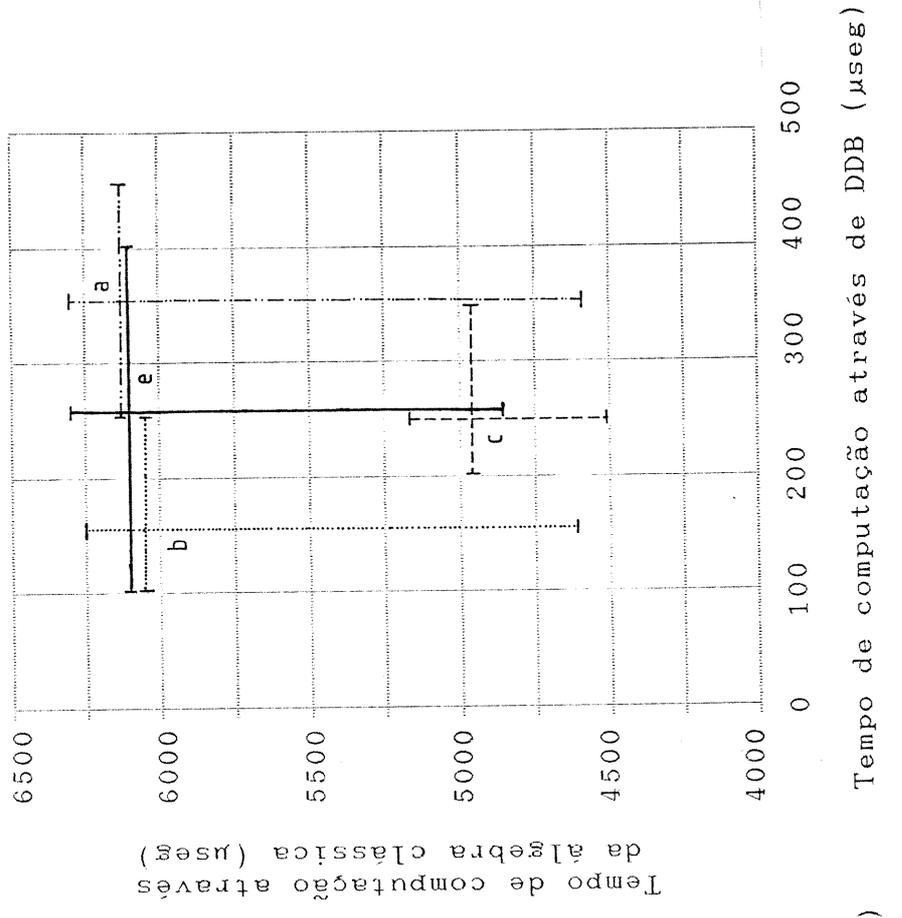
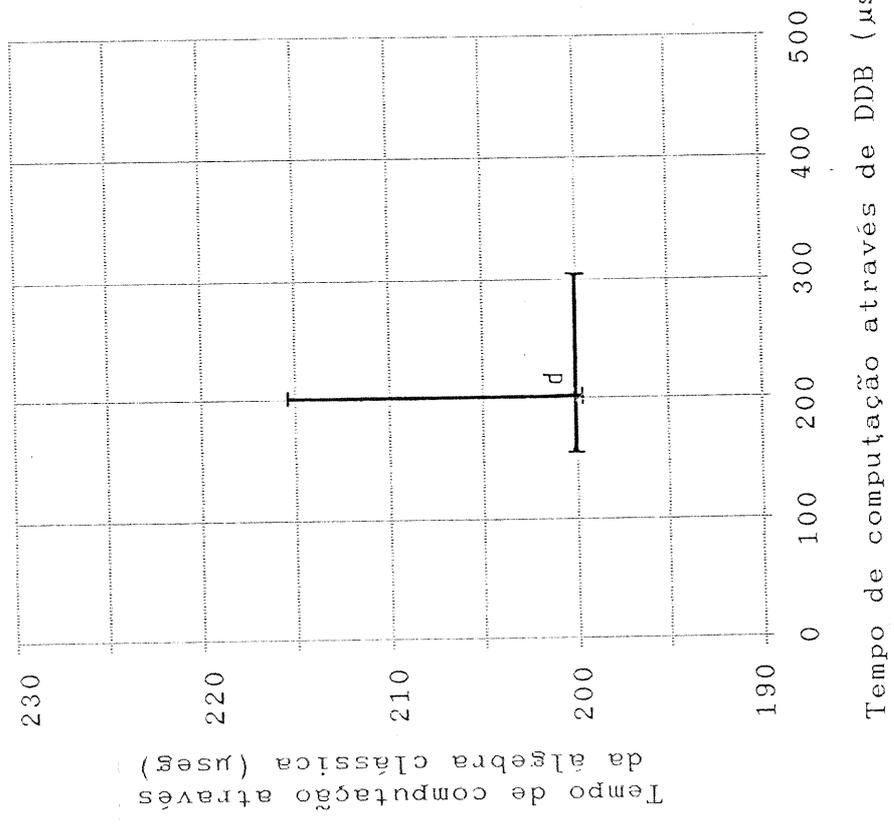


Fig. 5.12 Comparação dos tempos de computação requeridos pelos termos  $a=(m_2r_2^2+m_2r_1r_2C_2)\ddot{q}_2$ ,  $b=-m_2r_1r_2S_2\dot{q}_2^2$  e  $c=g(m_1+m_2)+r_1C_1$  presentes em  $T_1$  e  $d=(m_2r_2^2+J_2)\ddot{q}_2$  e  $e=m_2r_1r_2S_2\dot{q}_1^2$  presentes em  $T_2$  usando cálculos através de DDB e cálculos baseados na álgebra clássica. Computação efectuada em Turbo Pascal V4.0, e correndo numa máquina 8086, 8 MHz, sob MSDOS V3.2. As barras estão centradas nos valores médios, e variam desde o valor mínimo até ao valor máximo.

#### 5.4. ARQUITECTURAS COMPUTACIONAIS PARALELAS SUGERIDAS PELO COMPILADOR DEDICADO

Anteriormente mostrou-se que o método de compilação dedicada permite desempenhos consideravelmente superiores quando implementados numa máquina clássica com monoprocessamento sequencial.

Neste parágrafo estudam-se as implicações deste método na concepção de novas arquiteturas computacionais paralelas que, conforme se verificará, não apresentam as restrições manifestadas pelas demais estruturas propostas na literatura científica.

O compilador dedicado gera um código máquina que difere, de uma forma notável, do tipo de código produzido pelos compiladores convencionais. Assim, para o primeiro caso (compilador dedicado) sobrevêm as seguintes características:

- O código máquina gerado está distribuído por um pequeno conjunto de instruções, mormente do tipo {E, OU, NÃO} ou seu equivalente para uma realização da álgebra de Boole mediante DDB.

- A computação de cada binário é repartida por diversas parcelas as quais, por sua vez, dão origem, implicitamente, a uma partição dos cálculos por cada um dos seus bits.

- A partição da comutação, primeiramente pelas parcelas e posteriormente pelos respectivos bits, é conseguida de uma forma que permite um "desacoplamento" dos diversos cálculos envolvidos, isto é, possibilita a realização dos vários cálculos sem a existência de quaisquer tipos de interações.

No segundo caso (uso de um "compilador geral") verifica-se que:

- Contrariamente ao primeiro método, um "compilador geral" (Pascal, C, etc) produz um código máquina que, normalmente, se distribui mais amplamente pelas diversas instruções suportadas

pelo microprocessador.

- Neste caso, a partição da computação de cada binário é substancialmente mais complexa. Conforme foi mencionado na introdução, actualmente o algoritmo tido por mais eficiente consiste no "cálculo por medida" baseado no método de Newton-Euler [5.30-5.32]. No entanto, a partição das computações por vários processadores [5.19-5.24] tem-se mostrado difícil e ineficiente devido às complexas interacções existentes, estando o melhoramento máximo restringido a um factor de  $n$ , e que só é possível se forem permitidos alguns pequenos erros [5.23].

- Quaisquer que sejam os algoritmos de computação, a utilização de um "compilador geral" gera um código cuja computação é impossível de distribuir por cada bit da palavra da variável de saída.

Com base nestas propriedades conclui-se que o compilador dedicado produz um código máquina extremamente fácil de distribuir - desacopuladamente - por vários processadores em paralelo. Além disso, a distribuição do código máquina por um pequeno número de instruções, aliada ao facto das operações envolvidas se realizarem em cada bit separadamente, apontam como via de melhoramento a concepção de novas estruturas do próprio processador. À luz desta observação, a adopção de microprocessadores com palavras de oito, dezasseis ou trinta e dois bits, com um grande conjunto de instruções, e que necessitam de muitos ciclos máquina para efectuarem cada instrução, é manifestamente ineficiente. A concepção de procesadores dedicados, com palavras de um só bit, e tratando esta informação através de um conjunto reduzido de instruções, está mais indicada para as realizações pretendidas.

A Fig. 5.13 mostra uma das possíveis arquitecturas paralelas, e cuja adopção é agora exequível devido às características manifestadas pelo sistema de compilação dedicada.

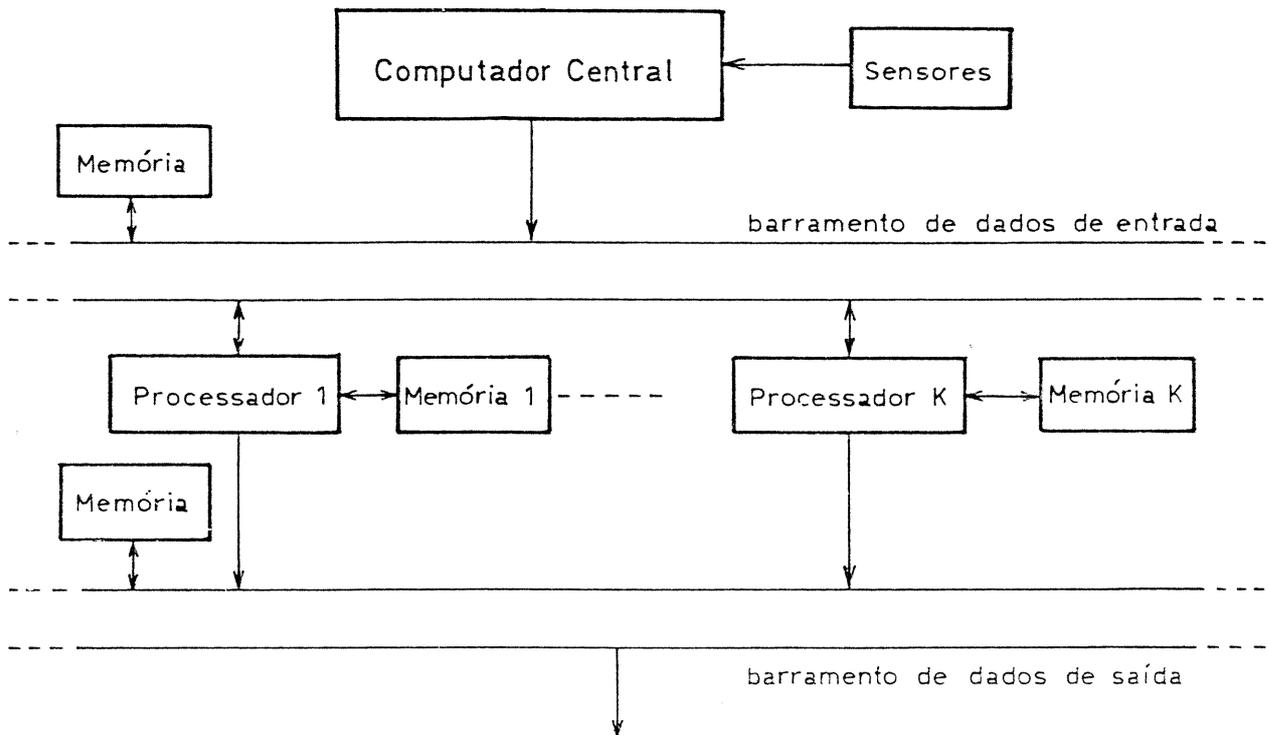


Fig. 5.13 Uma arquitectura computacional paralela sugerida pelo compilador dedicado. O código objecto é distribuído, desacopladamente, pelos diversos processadores em paralelo, de modo a evitarem-se problemas de "engarrafamento" dos cálculos.

## 5.5. DISCUSSÃO E CONCLUSÕES

Neste capítulo foi apresentada uma nova filosofia de compilação - compilação dedicada - que origina estruturas de computação muito mais rápidas do que as conseguidas por meios convencionais. Este método está baseado numa filosofia de "gestão efi-

ciente dos recursos estruturais" quer a nível do sistema computacional, quer a nível da estrutura mecânica e electrónica do robot. Assim, foi tomado como exemplo guia o cálculo da dinâmica inversa de manipuladores, mas a generalização a outros blocos de computação, nomeadamente à cinemática e ao controlo é, obviamente, imediata. Aliás, esta generalização pode ainda efectuar-se a um nível mais amplo, se se tomarem em consideração as características de muitos processos e respectivas estruturas de comando actualmente existentes. De facto, para a maioria destes casos constata-se a coexistência de especificações diferentes entre o computador e os demais sub-sistemas. A manutenção da filosofia referida, pode levar à aplicação do compilador desenvolvido ou, eventualmente, a outros métodos adaptados a cada processo em particular.

No tocante ao cálculo da dinâmica inversa de manipuladores, há ainda a referir que se seguiu como exemplo guia a utilização do compilador dedicado para um robot 2R com a respectiva dinâmica expressa na forma simbólica. No entanto, este compilador pode ser aplicado a outras formas de descrição da dinâmica inversa, nomeadamente ao método de Newton-Euler. A fácil generalização possível com este método, advem da sua "independência" relativamente ao algoritmo, isto é, por estar focado numa fase que é posterior à da derivação da forma do algoritmo.

Outro ponto extremamente importante e ainda não realçado, consiste na possibilidade de se usarem dados experimentais em contrapartida ao uso de dados obtidos a partir de um modelo. Efectivamente, o primeiro passo na aplicação deste compilador consiste na construção de tabelas numéricas correspondentes aos fenómenos dinâmicos em estudo. Desta maneira, o preenchimento da

tabela numérica pode ser baseado, alternativamente, em dados experimentais, desde que este procedimento seja compatível com a partição inerente à estrutura híbrida do compilador. Como é óbvio, se praticável, esta estratégia permite uma muito maior fidelidade na descrição do fenómeno físico, dificilmente atingível com base num modelo matemático.

Um aspecto interessante da filosofia de compilação dedicada, e que também está ligado à obtenção de dados experimentais, consiste na escolha dos intervalos de quantificação das diversas variáveis. Uma escolha cuidada destes intervalos permite "distribuir" o número  $z$  de bits de precisão muito mais eficientemente. De facto, a escolha de intervalos de quantificação, quer muito grandes, quer muito pequenos, tem efeitos indesejáveis:

- No primeiro caso, a variação de amplitude correspondente a um bit menos significativo vem demasiado elevada

- No segundo caso, poderão ocorrer excursões da amplitude da variável fora do intervalo de quantificação considerado.

Na realidade, uma escolha criteriosa pode conseguir-se com base em dados experimentais, obtidos para as diversas condições de funcionamento. Sabendo os valores máximos e mínimos das excursões medidas tem-se um quadro fiel que permite escolher intervalos de quantificação mais ajustados. Esta estratégia pode então ser interpretada como mais um processo de capitalizar os conhecimentos adquiridos experimentalmente.

Em conclusão, pode-se afirmar que a filosofia de "gestão eficiente dos recursos estruturais", aplicada no capítulo 3 a uma repartição óptima do algoritmo por diversas frequências de amostragem, e no capítulo 4 à integração no algoritmo de controlo

de um maior leque de informação sensorial, é aqui complementada com a optimização da fase final de implementação, ou seja a sua compilação. Esta complementaridade pode, conforme já foi mencionado, manifestar-se na co-utilização deste método com as estratégias referidas nos capítulos 3 e 4.

## APÊNDICE 1

Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, sem a aplicação das R.E. e R.T.

Precisão desejada:  $z=8$  bits

Número extra de bits de precisão:  $m=2$

Precisão requerida a cada variável:  $\hat{z}=10$  bits

Número de parcelas na expressão de  $T_1$ :  $p=5$

Número de parcelas na expressão de  $T_2$ :  $p=4$

$$T_1 = T_{j11} + T_{j12} + T_{c122} + T_{c112} + T_{g1}$$

$$T_2 = T_{j21} + T_{j22} + T_{c211} + T_{g2}$$

$$\left\{ \begin{array}{l} T_{j11} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\ T_{j12} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\ T_{c122} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ T_{c112} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 \\ T_{g1} = g [m_1 r_1 C_1 + m_2 (r_2 C_{12} + r_1 C_1)] \end{array} \right.$$

$$\left\{ \begin{array}{l} T_{j21} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \\ T_{j22} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\ T_{c211} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\ T_{g2} = g m_2 r_2 C_{12} \end{array} \right.$$

Parcela  $T_{j11} = j_{11}(q_2) \ddot{q}_1$

$$T_{j11} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1$$

Intervalo de quantificação:  $-200 \text{ Nm} \leq T_{j11} \leq 200 \text{ Nm}$

Intervalo de variação:  $-25.75 \text{ Nm} \leq T_{j11} \leq 25.75 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_1\}$ :  $20 \cdot 2^{2^0}$

variável de saída  $\{T_{j11}\}$ :  $10 \cdot 2^{2^0}$

total:  $30 \cdot 2^{2^0}$

Porcentagem de estados de quantificação de  $T_{j11}$  usados: 12.9 %

Parcela  $T_{j12} = j_{12}(q_2) \ddot{q}_2$

$$T_{j12} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2$$

Intervalo de quantificação:  $-200 \text{ Nm} \leq T_{j12} \leq 200 \text{ Nm}$

Intervalo de variação:  $-9 \text{ Nm} \leq T_{j12} \leq 9 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_2\}$ :  $20 \cdot 2^{2^0}$

variável de saída  $\{T_{j12}\}$ :  $10 \cdot 2^{2^0}$

total:  $30 \cdot 2^{2^0}$

Porcentagem de estados de quantificação de  $T_{j12}$  usados: 4.5 %

Parcela  $T_{c122} = c_{122}(q_2) \dot{q}_2^2$

$$T_{c122} = -m_2 r_1 r_2 S_2 \dot{q}_2^2$$

Intervalo de quantificação:  $-200 \text{ Nm} \leq T_{c122} \leq 200 \text{ Nm}$

Intervalo de variação:  $-5 \text{ Nm} \leq T_{c122} \leq 5 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_2\}$ :  $20 \cdot 2^{2^0}$

variável de saída  $\{T_{c122}\}$ :  $10 \cdot 2^{2^0}$

total:  $30 \cdot 2^{2^0}$

Porcentagem de estados de quantificação de  $T_{c122}$  usados: 2.5 %

Parcela  $T_{c112} = c_{112}(q_2)\dot{q}_1\dot{q}_2$

$$T_{c112} = -2m_2r_1r_2S_2\dot{q}_1\dot{q}_2$$

Intervalo de quantificação:  $-200 \text{ Nm} \leq T_{c112} \leq 200 \text{ Nm}$

Intervalo de variação:  $-10 \text{ Nm} \leq T_{c112} \leq 10 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_1, \dot{q}_2\}$ :  $30 \cdot 2^{30}$

variável de saída  $\{T_{c112}\}$ :  $10 \cdot 2^{30}$

total:  $40 \cdot 2^{30}$

Percentagem de estados de quantificação de  $T_{c112}$  usados: 5 %

Parcela  $T_{g1} = g_1(q_1, q_2)$

$$T_{g1} = g[m_1r_1C_1 + m_2(r_2C_{12} + r_1C_1)]$$

Intervalo de quantificação:  $-200 \text{ Nm} \leq T_{g1} \leq 200 \text{ Nm}$

Intervalo de variação:  $-115.15 \text{ Nm} \leq T_{g1} \leq 115.15 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, q_1\}$ :  $20 \cdot 2^{20}$

variável de saída  $\{T_{g1}\}$ :  $10 \cdot 2^{20}$

total:  $30 \cdot 2^{20}$

Percentagem de estados de quantificação de  $T_{g1}$  usados: 57.6 %

Parcela  $T_{j21} = j_{21}(q_2)\ddot{q}_1$

$$T_{j21} = [m_2(r_2^2 + r_1r_2C_2)]\ddot{q}_1$$

Intervalo de quantificação:  $-100 \text{ Nm} \leq T_{j21} \leq 100 \text{ Nm}$

Intervalo de variação:  $-9 \text{ Nm} \leq T_{j21} \leq 9 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \ddot{q}_1\}$ :  $20 \cdot 2^{20}$

variável de saída  $\{T_{j21}\}$ :  $10 \cdot 2^{20}$

total:  $30 \cdot 2^{20}$

Percentagem de estados de quantificação de  $T_{j21}$  usados: 9 %

Parcela  $T_{j22} = j_{22}\ddot{q}_2$

$$T_{j22} = (m_2r_2^2 + J_2)\ddot{q}_2$$

Intervalo de quantificação:  $-100 \text{ Nm} \leq T_{j22} \leq 100 \text{ Nm}$

Intervalo de variação:  $-9 \text{ Nm} \leq T_{j22} \leq 9 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{\ddot{q}_2\}$ :  $10 \cdot 2^{10}$

variável de saída  $\{T_{j22}\}$ :  $10 \cdot 2^{10}$

total:  $20 \cdot 2^{10}$

Percentagem de estados de quantificação de  $T_{j22}$  usados: 9 %

Parcela  $T_{c211} = c_{211}(q_2)\dot{q}_1^2$

$$T_{c211} = m_2r_1r_2S_2\dot{q}_1^2$$

Intervalo de quantificação:  $-100 \text{ Nm} \leq T_{c211} \leq 100 \text{ Nm}$

Intervalo de variação:  $-5 \text{ Nm} \leq T_{c211} \leq 5 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_1\}$ :  $20 \cdot 2^{20}$

variável de saída  $\{T_{c211}\}$ :  $10 \cdot 2^{20}$

total:  $30 \cdot 2^{20}$

Percentagem de estados de quantificação de  $T_{c211}$  usados: 5 %

Parcela  $T_{g2} = g_2(q_1, q_2)$

$$T_{g2} = g_{m2} r_2 C_{12}$$

Intervalo de quantificação:  $-100 \text{ Nm} \leq T_{g2} \leq 100 \text{ Nm}$

Intervalo de variação:  $-49 \text{ Nm} \leq T_{g2} \leq 49 \text{ Nm}$

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_1, q_2\}$ :  $20 \cdot 2^{20}$

variável de saída  $\{T_{g2}\}$ :  $10 \cdot 2^{20}$

total:  $30 \cdot 2^{20}$

Porcentagem de estados de quantificação de  $T_{g2}$  usados: 49 %

## APÊNDICE 2

Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, com a aplicação das R.E. e R.T. excepto a R.E.3.

Precisão desejada:  $z=8$  bits

Número extra de bits de precisão:  $m=2$

Precisão requerida a cada variável:  $\hat{z}=10$  bits

Número de parcelas na expressão de  $T_1$ :  $p=p_M=6$

Número de parcelas na expressão de  $T_2$ :  $p=p_M=6$

$$T_1 = T_{j_{11}} + T_{j_{12}} + T_{c_{122}} + T_{c_{112}} + (T_{g_1^A} + T_{g_1^B})$$

$$T_2 = (T_{j_{21}^A} + T_{j_{21}^B}) + T_{j_{22}} + T_{c_{211}} + (T_{g_2^A} + T_{g_2^B})$$

$$\left\{ \begin{array}{l} T_{j_{11}} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\ T_{j_{12}} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\ T_{c_{122}} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ T_{c_{112}} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2 \\ \left\{ \begin{array}{l} T_{g_1^A} = g(m_1 + m_2) r_1 C_1 \\ T_{g_1^B} = g m_2 r_2 C_{12} \\ T_{g_1} = T_{g_1^A} + T_{g_1^B} \end{array} \right. \end{array} \right.$$

$$\left\{ \begin{array}{l} T_{j_{21}^A} = T_{j_{21}^B} = \{ [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \} / 2 \\ T_{j_{21}} = T_{j_{21}^A} + T_{j_{21}^B} \\ T_{j_{22}} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\ T_{c_{211}} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\ \left\{ \begin{array}{l} T_{g_2^A} = T_{g_2^B} = (g m_2 r_2 C_{12}) / 2 \\ T_{g_2} = T_{g_2^A} + T_{g_2^B} \end{array} \right. \end{array} \right.$$

Parcela  $T_{j_{11}} = j_{11}(q_2) \ddot{q}_1$

$$T_{j_{11}} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1$$

Intervalo de quantificação:  $-50 \text{ Nm} \leq T_{j_{11}} \leq 50 \text{ Nm}$

Intervalo de variação:  $-25.75 \text{ Nm} \leq T_{j_{11}} \leq 25.75 \text{ Nm}$

Número de bits de  $q_2$ : 7

Número de bits de  $\ddot{q}_1$ : 8

Número de bits de  $T_{j_{11}}$ : 8

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \ddot{q}_1\}$ :  $15 \cdot 2^{15}$

variável de saída  $\{T_{j_{11}}\}$ :  $8 \cdot 2^{15}$

total:  $23 \cdot 2^{15}$

Porcentagem de estados de quantificação de  $T_{j_{11}}$  usados: 51.5 %

Parcela  $T_{j_{12}} = j_{12}(q_2) \ddot{q}_2$

$$T_{j_{12}} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2$$

Intervalo de quantificação:  $-12.5 \text{ Nm} \leq T_{j_{12}} \leq 12.5 \text{ Nm}$

Intervalo de variação:  $-9 \text{ Nm} \leq T_{j_{12}} \leq 9 \text{ Nm}$

Número de bits de  $q_2$ : 5

Número de bits de  $\ddot{q}_2$ : 6

Número de bits de  $T_{j_{12}}$ : 6

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \ddot{q}_2\}$ :  $11 \cdot 2^{11}$

variável de saída  $\{T_{j_{12}}\}$ :  $6 \cdot 2^{11}$

total:  $17 \cdot 2^{11}$  bits

Porcentagem de estados de quantificação de  $T_{j_{12}}$  usados: 72 %

Parcela  $T_{c122} = c_{122}(q_2)\dot{q}_2^2$

$$T_{c122} = -m_2 r_1 r_2 S_2 \dot{q}_2^2$$

Intervalo de quantificação:  $-6.25 \text{ Nm} \leq T_{c122} \leq 6.25 \text{ Nm}$

Intervalo de variação:  $-5 \text{ Nm} \leq T_{c122} \leq 5 \text{ Nm}$

Número de bits de  $q_2$ : 5

Número de bits de  $\dot{q}_2$ : 4

Número de bits de  $T_{c122}$ : 5

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_2\}$ :  $9 \cdot 2^0$

variável de saída  $\{T_{c122}\}$ :  $5 \cdot 2^0$

total:  $14 \cdot 2^0$

Porcentagem de estados de quantificação de  $T_{c122}$  usados: 80 %

Parcela  $T_{c112} = c_{112}(q_2)\dot{q}_1\dot{q}_2$

$$T_{c112} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2$$

Intervalo de quantificação:  $-12.5 \text{ Nm} \leq T_{c112} \leq 12.5 \text{ Nm}$

Intervalo de variação:  $-10 \text{ Nm} \leq T_{c112} \leq 10 \text{ Nm}$

Número de bits de  $q_2$ : 6

Número de bits de  $\dot{q}_1$ : 6

Número de bits de  $\dot{q}_2$ : 6

Número de bits de  $T_{j11}$ : 6

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_2, \dot{q}_1, \dot{q}_2\}$ :  $18 \cdot 2^{1^0}$

variável de saída  $\{T_{c112}\}$ :  $6 \cdot 2^{1^0}$

total:  $24 \cdot 2^{1^0}$

Porcentagem de estados de quantificação de  $T_{c112}$  usados: 80 %

Parcela  $T_{g1^A} = g_1^A(q_1)$

$$T_{g1^A} = g(m_1 + m_2) r_1 C_1$$

Intervalo de quantificação:  $-100 \text{ Nm} \leq T_{g1^A} \leq 100 \text{ Nm}$

Intervalo de variação:  $-66.15 \text{ Nm} \leq T_{g1^A} \leq 66.15 \text{ Nm}$

Número de bits de  $q_1$ : 8

Número de bits de  $T_{g1^A}$ : 8

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_1\}$ :  $8 \cdot 2^0$

variável de saída  $\{T_{g1^A}\}$ :  $8 \cdot 2^0$  bits

total:  $16 \cdot 2^0$

Porcentagem de estados de quantificação de  $T_{g1^A}$  usados: 66.2 %

Parcela  $T_{g1^B} = g_1^B(q_1, q_2)$

$$T_{g1^B} = g m_2 r_2 C_{12}$$

Intervalo de quantificação:  $-50 \text{ Nm} \leq T_{g1^B} \leq 50 \text{ Nm}$

Intervalo de variação:  $-49 \text{ Nm} \leq T_{g1^B} \leq 49 \text{ Nm}$

Número de bits de  $q_1$ : 8

Número de bits de  $q_2$ : 8

Número de bits de  $T_{g1^B}$ : 8

Número de bits da tabela de verdade para:

variáveis de entrada  $\{q_1, q_2\}$ :  $16 \cdot 2^{1^0}$

variável de saída  $\{T_{g1^B}\}$ :  $8 \cdot 2^{1^0}$

total:  $24 \cdot 2^{1^0}$

Porcentagem de estados de quantificação de  $T_{g1^B}$  usados: 98 %

Parcela  $T_{j21}^A = j_{21}^A(q_2)\ddot{q}_1$   
 $T_{j21}^A = [m_2(r_2^2 + r_1 r_2 C_2)]\ddot{q}_1/2$   
 Número de bits de  $q_2$ : 5  
 Número de bits de  $\ddot{q}_1$ : 6  
 Número de bits de  $T_{j21}^A$ : 6  
 Intervalo de quantificação:  $-6.25 \text{ Nm} \leq T_{j21}^A \leq 6.25 \text{ Nm}$   
 Intervalo de variação:  $-4.5 \text{ Nm} \leq T_{j21}^A \leq 4.5 \text{ Nm}$   
 Número de bits da tabela de verdade para:  
 variáveis de entrada  $\{q_2, \ddot{q}_1\}$ :  $11 \cdot 2^{11}$   
 variável de saída  $\{T_{j21}^A\}$ :  $6 \cdot 2^{11}$   
 total:  $17 \cdot 2^{11}$   
 Percentagem de estados de quantificação de  $T_{j21}^A$  usados: 72 %

Parcela  $T_{j22} = j_{22} \ddot{q}_2$   
 $T_{j22} = (m_2 r_2^2 + J_2)\ddot{q}_2$   
 Intervalo de quantificação:  $-12.5 \text{ Nm} \leq T_{j22} \leq 12.5 \text{ Nm}$   
 Intervalo de variação:  $-9 \text{ Nm} \leq T_{j22} \leq 9 \text{ Nm}$   
 Número de bits de  $\ddot{q}_2$ : 7  
 Número de bits de  $T_{j22}$ : 7  
 Número de bits da tabela de verdade para:  
 variáveis de entrada  $\{\ddot{q}_2\}$ :  $7 \cdot 2^7$   
 variável de saída  $\{T_{j22}\}$ :  $7 \cdot 2^7$   
 total:  $14 \cdot 2^7$   
 Percentagem de estados de quantificação de  $T_{j22}$  usados: 72 %

Parcela  $T_{c211} = c_{211}(q_2)\dot{q}_1^2$   
 $T_{c211} = m_2 r_1 r_2 S_2 \dot{q}_1^2$   
 Intervalo de quantificação:  $-6.25 \text{ Nm} \leq T_{c211} \leq 6.25 \text{ Nm}$   
 Intervalo de variação:  $-5 \text{ Nm} \leq T_{c211} \leq 5 \text{ Nm}$   
 Número de bits de  $q_2$ : 6  
 Número de bits de  $\dot{q}_1$ : 5  
 Número de bits de  $T_{c211}$ : 6  
 Número de bits da tabela de verdade para:  
 variáveis de entrada  $\{q_2, \dot{q}_1\}$ :  $11 \cdot 2^{11}$   
 variável de saída  $\{T_{c211}\}$ :  $6 \cdot 2^{11}$   
 total:  $17 \cdot 2^{11}$   
 Percentagem de estados de quantificação de  $T_{c211}$  usados: 80 %

Parcela  $T_{g2}^A = g_2^A(q_1, q_2)$   
 $T_{g2}^A = (g m_2 r_2 C_{12})/2$   
 Intervalo de quantificação:  $-25 \text{ Nm} \leq T_{g2} \leq 25 \text{ Nm}$   
 Intervalo de variação:  $-24.5 \text{ Nm} \leq T_{g2} \leq 24.5 \text{ Nm}$   
 Número de bits de  $q_1$ : 8  
 Número de bits de  $q_2$ : 8  
 Número de bits de  $T_{g2}^A$ : 8  
 Número de bits da tabela de verdade para:  
 variáveis de entrada  $\{q_1, q_2\}$ :  $16 \cdot 2^{16}$   
 variável de saída  $\{T_{g2}\}$ :  $8 \cdot 2^{16}$   
 total:  $24 \cdot 2^{16}$  bits  
 Percentagem de estados de quantificação de  $T_{g2}^A$  usados: 98 %

APÊNDICE 3

Exigências das diversas tabelas de verdade requeridas pelo compilador híbrido, com a aplicação das R.E. e R.T. incluindo a R.E.3.

Precisão desejada:  $z=8$  bits

Número extra de bits de precisão:  $m=2$

Precisão requerida a cada variável:  $\hat{z}=10$  bits

Número de parcelas na expressão de  $T_1$ :  $p=p_M=6$

Número de parcelas na expressão de  $T_2$ :  $p=p_M=6$

$$T_1 = T_{j_{11}} + T_{j_{12}} + T_{c_{122}} + T_{c_{112}} + (T_{g_1^A} + T_{g_1^B})$$

$$T_2 = (T_{j_{21}^A} + T_{j_{21}^B}) + T_{j_{22}} + T_{c_{211}} + (T_{g_2^A} + T_{g_2^B})$$

$$\left\{ \begin{array}{l} T_{j_{11}} = [m_1 r_1^2 + m_2 (r_1^2 + r_2^2 + 2r_1 r_2 C_2) + J_1] \ddot{q}_1 \\ T_{j_{12}} = [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_2 \\ T_{c_{122}} = -m_2 r_1 r_2 S_2 \dot{q}_2^2 \\ X = -2m_2 r_1 r_2 S_2 \dot{q}_1 \\ T_{c_{112}} = X \dot{q}_2 \\ \left\{ \begin{array}{l} T_{g_1^A} = g(m_1 + m_2) r_1 C_1 \\ T_{g_1^B} = g m_2 r_2 C_{12} \\ T_{g_1} = T_{g_1^A} + T_{g_1^B} \end{array} \right. \end{array} \right.$$

$$\left\{ \begin{array}{l} T_{j_{21}^A} = T_{j_{21}^B} = \{ [m_2 (r_2^2 + r_1 r_2 C_2)] \ddot{q}_1 \} / 2 \\ T_{j_{21}} = T_{j_{21}^A} + T_{j_{21}^B} \\ T_{j_{22}} = (m_2 r_2^2 + J_2) \ddot{q}_2 \\ T_{c_{211}} = m_2 r_1 r_2 S_2 \dot{q}_1^2 \\ T_{g_2^A} = T_{g_2^B} = (g m_2 r_2 C_{12}) / 2 \\ T_{g_2} = T_{g_2^A} + T_{g_2^B} \end{array} \right.$$

Parcela  $T_{c_{122}} = c_{122} (q_2) \dot{q}_1 \dot{q}_2$

$$T_{c_{122}} = -2m_2 r_1 r_2 S_2 \dot{q}_1 \dot{q}_2$$

Cascata/paralelo:  $X = -2m_2 r_1 r_2 S_2 \dot{q}_1$ ,  $T_{c_{112}} = X \dot{q}_2$

Intervalo de quantificação:  $-12.5 \text{ Nm} \leq X \leq 12.5 \text{ Nm}$

Intervalo de variação:  $-10 \text{ Nm} \leq X \leq 10 \text{ Nm}$

Intervalo de quantificação:  $-12.5 \text{ Nm} \leq T_{c_{112}} \leq 12.5 \text{ Nm}$

Intervalo de variação:  $-10 \text{ Nm} \leq T_{c_{112}} \leq 10 \text{ Nm}$

Número de bits de  $q_2$ : 6

Número de bits de  $\dot{q}_1$ : 6

Número de bits de  $\dot{q}_2$ : 6

Número de bits de  $X$ : 6

Número de bits de  $T_{c_{112}}$ : 6

Número de bits da tabela de verdade de  $X = c_{122} (q_2) \dot{q}_1$  para:

variáveis de entrada  $\{q_2, \dot{q}_1\}$ :  $12 \cdot 2^{12}$

variável de saída  $\{X\}$ :  $6 \cdot 2^{12}$

total:  $18 \cdot 2^{12}$

Porcentagem de estados de quantificação de  $X$  usados: 80 %

Número de bits da tabela de verdade de  $T_{c_{112}} = X \dot{q}_2$  para

variáveis de entrada  $\{X, \dot{q}_2\}$ :  $12 \cdot 2^{12}$

variável de saída  $\{T_{c_{112}}\}$ :  $6 \cdot 2^{12}$

total:  $18 \cdot 2^{12}$

Porcentagem de estados de quantificação de  $T_{c_{112}}$  usados: 80 %

## REFERÊNCIAS

- [5.1] Murray R. Spiegel  
Theoretical Mechanics with an Introduction of  
Lagrange's Equations and Hamiltonian Theory  
McGraw-Hill Inc., 1967.
- [5.2] Emil Jan Konopinski  
Classical Descriptions of Motion  
W. E. Freeman and Company, San Francisco, 1969.
- [5.3] John M. Hollerbach  
A Recursive Lagrangian Formulation of Manipulator  
Dynamics and a Comparative Study of Dynamics  
Formulation Complexity  
IEEE Trans. Syst., Man, Cybern., vol. SMC-10,  
pp. 730-736, Nov., 1980.
- [5.4] J. Y. S. Luh, M. W. Walker e R. P. C. Paul  
On-Line Computational Scheme for Mechanical  
Manipulators  
ASME J. Dynamic Syst. Meas. Contr., vol. 102,  
pp. 69-76, June, 1980.
- [5.5] William M. Silver  
On the Equivalence of Lagrangian and Newton-Euler  
Dynamics for Manipulators  
The Int. J. Robotics Research, vol. 1, n. 2, pp. 60-70,  
Summer, 1982.
- [5.6] J. S. Albus  
A New Approach to Manipulator Control: The Cerebellar  
Model Articulation Controller (CMAC)  
ASME J. Dynamic Syst. Meas. Contr., vol. 97,  
pp. 220-227, Sept., 1975.
- [5.7] J. S. Albus  
Data Storage in the Cerebellar Model Articulation  
Controller (CMAC)  
ASME J. Dynamic Syst. Meas. Contr., vol. 97,  
pp. 228-233, Sept., 1975.
- [5.8] B. K. P. Horn e M. H. Raibert  
Configuration Space Control  
MIT A.I. Memo, n. 458, Dec., 1977.
- [5.9] Marc H. Raibert  
Analytical Equations vs. Table Look-Up for  
Manipulation. A Unifying Concept  
Proc. IEEE Conf. Decision Contr., New Orleans La.,  
pp. 576-579, Dec., 1977.
- [5.10] M. H. Raibert  
A Model for Sensorimotor Control and Learning  
Biological Cybernetics 29, pp. 29-36, 1978.

- [5.11] M. H. Raibert e B. K. P. Horn  
 Manipulator Control Using the Configuration Space  
 Method  
 The Industrial Robot 5, n. 2, pp. 69-73, June, 1978.
- [5.12] G. Plank e G. Hirzinger  
 Controlling a Robot's Motion Speed by Force-Torque-  
 -Sensor for Deburring Problems  
 4th IFAC-IFIP Symp. on Information Control Problems,  
 Gaithersburg, Maryland, USA, Oct., 1982.
- [5.13] G. Hirzinger  
 Direct Digital Control Using a Force-Torque-Sensor  
 IFAC Symp. on Real-Time Digital Control Applications,  
 Guadalajara, Mexico, Jan., 1983
- [5.14] K. Arbter, J. Heindl, G. Hirzinger, K. Landzettel e  
 F. Lange  
 New Techniques for Teach-in, Acceleration and Learning  
 in Sensor-Controlled Robots  
 IFAC'84 9th World Congress of the Int. Federation of  
 Automatic Control, Budapest, Hungary, July, 1984.
- [5.15] Gerhard Hirzinger  
 Robot Systems Completely Based on Sensory Feedback  
 IEEE Trans. Ind. Electron., vol. IE-33, pp. 105-109,  
 June, 1986.
- [5.16] Friedrich Lange  
 Der Assoziativspeicher ME7  
 DFVLR-FB 86-31, 1986.
- [5.17] W. Thomas Miller III  
 Sensor-Based Control of Robotic Manipulators Using  
 General Learning Algorithms  
 IEEE J. Robotics and Automation, vol RA-3, pp. 157-165,  
 April, 1987.
- [5.18] W. T. Miller III, F. H. Glanz e L. G. Kraft III  
 Application of a General Learning Algorithm to the  
 Control of Robotic Manipulators  
 The Int. J. Robotics Research, vol. 6, n. 2, pp. 84-98,  
 Summer, 1987.
- [5.19] J. Y. S. Luh e C. S. Lin  
 Scheduling of Parallel Computation for a Computer-  
 -Controlled Mechanical Manipulator  
 IEEE Trans. Syst., Man, Cybern., vol. SMC-12,  
 pp. 214-234, March/April, 1982.
- [5.20] Chang-Huan Liu e Yen-Ming Chen  
 Multi-Microprocessor-Based Cartesian-Space Control  
 Techniques for a Mechanical Manipulator  
 IEEE J. Robotics and Automation, vol RA-2, pp. 110-115,  
 June, 1986.

- [5.21] Ravi Nigam e C. S. George Lee  
A Multiprocessor-Based Controller for the Control of  
Mechanical Manipulators  
IEEE J. Robotics and Automation, vol. RA-1,  
pp. 173-182, Dec., 1985.
- [5.22] T. Watanabe, M. Kametani, K. Kawata e K. Tetsuya  
Improvement in the Computing Time of Robot Manipulators  
Using a Multimicroprocessor  
ASME J. Dynamic Syst. Meas. Contr., vol. 108,  
pp. 190-197, Sept., 1986.
- [5.23] Eli E. Binder e James H. Herzog  
Distributed Computer Architecture and Fast Parallel  
Algorithms in Real-Time Robot Control  
IEEE Trans. Syst., Man, Cybern., vol. SMC-16,  
pp. 543-549, July/Aug., 1986.
- [5.24] D. I. Jones e P. M. Entwistle  
Parallel Computation of an Algorithm in Robotic Control  
IEE Int. Conf. Control'88, Oxford, UK, 1988.
- [5.25] D. T. Horak  
A Simplified Modeling and Computational Scheme for  
Manipulator Dynamics  
ASME J. Dynamic Syst. Meas. Contr., vol. 106,  
pp. 350-353, Dec., 1984.
- [5.26] H. Faessler  
Computer-Assisted Generation of Dynamic Equations for  
Multibody Systems"  
The Int. J. Robotics Research, vol. 5, n. 3,  
pp. 129-141, Fall, 1986.
- [5.27] M. C. Leu e N. Hemati  
Automated Symbolic Derivation of Dynamic Equations of  
Motion for Robotic Manipulators  
ASME J. Dynamic Syst. Meas. Contr., vol. 108,  
pp. 172-179, Sept., 1986.
- [5.28] J. Koplic e M. C. Leu  
Computer Generation of Robot Dynamic Equations and the  
Related Issues  
J. of Robotic Systems, vol. 3, n. 3, pp. 301-319, Fall,  
1986.
- [5.29] D. C. Yang e S. W. Tzeng  
Simplification and Linearization of Manipulator  
Dynamics by the Design of Inertia Distribution  
The Int. J. Robotics Research, vol.5, n. 3,  
pp. 120-128, Fall, 1986.
- [5.30] Charles P. Neuman e John J. Murray  
The Complete Dynamic Model and Customized Algorithms of  
the Puma Robot,  
IEEE Trans. Syst., Man, Cybern., vol. SMC-17,  
pp. 635-644, July/Aug., 1987.

- [5.31] Charles P. Neuman e John J. Murray  
 Customized Computational Robot Dynamics  
 J. of Robotic Systems, vol. 4, n. 4, pp. 503-526.,  
 Aug., 1987.
- [5.32] Charles P. Neuman e John J. Murray  
 Symbolically Efficient Formulations for Computational  
 Robot Dynamics  
 J. of Robotic Systems, vol. 4, n. 6, pp. 743-769, Dec.,  
 1987.
- [5.33] Richard P. Paul  
 Robot Manipulators: Mathematics, Programming and  
 Control  
 Cambridge, MA: Mass. Inst. Tech., 1981.
- [5.34] John M. Hollerbach  
 Dynamic Scalling of Manipulator Trajectories  
 ASME J. Dynamic Syst. Meas. Contr., vol. 106,  
 pp. 102-106, March, 1984.
- [5.35] Takashi Suehiro e Kunikatsu Takase  
 A Manipulation System Based on Direct-Computational  
 Task-Coordinate Servoing  
 Robotics Research The Second International Symposium,  
 pp. 257-264, Cambridge, MA: Mass. Inst. Tech., 1985.
- [5.36] J. A. Tenreiro Machado, J. L. Martins de Carvalho,  
 J. A. Silva Matos e Antonio M. C. Costa  
 A Real-Time System for Robot Manipulator Inverse  
 Dynamics Computation  
 15th IFAC/IFIP Workshop on Real-Time Programming,  
 Valencia, Spain, 1988.
- [5.37] J. A. Tenreiro Machado, J. L. Martins de Carvalho,  
 Antonio M. C. Costa e José S. Matos  
 A New Computational Scheme for Robot Manipulators  
 NATO ARW on "Robots with Redundancy: Design, Sensing  
 and Control", Salo, Itália, 1988.
- [5.38] J. A. Tenreiro Machado, J. L. Martins de Carvalho,  
 Antonio M. C. Costa e José S. Matos,  
 Dedicated Computer System for Robot Manipulators  
 IFAC 3rd Int. Symp. on Systems Analysis and Simulation,  
 Berlin, GDR, 1988.
- [5.39] J. A. Tenreiro Machado, J. L. Martins de Carvalho,  
 Antonio M. C. Costa e José S. Matos  
 Robot Manipulator Dynamics - Towards Better  
 Computational Algorithms  
 IFAC Symp. on Robot Control, Karlsruhe, FRG, 1988.
- [5.40] W. V. Quine  
 The Problem of Simplifying Truth Functions  
 Am. Math. Monthly, vol. 59, n. 8, pp. 521-531,  
 Oct., 1952.

- [5.41] E. J. McCluskey  
Minimization of Boolean Functions  
Bell Systems Tech. Journal, vol. 35, n. 5,  
pp. 1413-1444, Nov., 1956.
- [5.42] José S. Matos  
The Binary Decision Diagram: A Tool for Logic Design  
and Implementation,  
PhD Dissertation, Syracuse University, Syracuse, N. Y.,  
1983.
- [5.43] José S. Matos e John V. Oldfield  
Binary Decision Diagrams: From Abstract Representations  
to Physical Implementations  
20<sup>th</sup> IEEE/ACM Design Automation Conference,  
Miami Beach, Florida, 1983.

CAPÍTULO 6

CONCLUSÃO

## 6. CONCLUSÃO

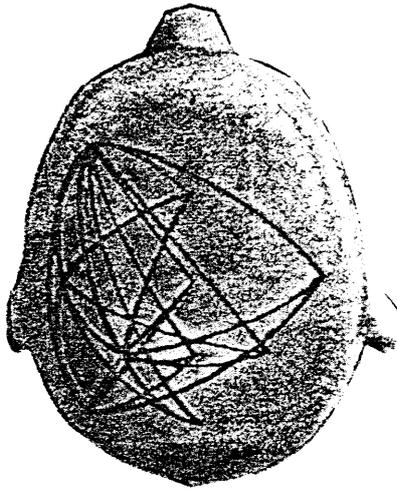
O leitor que folheia estas páginas tem estado, inadvertidamente, a demonstrar alguns dos conceitos que serviram de base ao trabalho apresentado. De facto, tarefas tão simples como folhear um livro, e que o ser humano realiza quase automaticamente, colocam problemas delicados quando se tenta implementá-las com os robots manipuladores presentemente existentes.

Uma comparação mais detalhada dos problemas envolvidos, mostra claramente quão longe está a tecnologia humana das capacidades dos sistemas biológicos. A acção de comando e controlo do braço humano, numa tarefa semelhante à descrita, apresenta um "peso computacional" insignificante para o cérebro, contudo a mímica tecnológica confronta-se com problemas que estão próximo do limite das capacidades dos computadores actuais. Os capítulos 3, 4 e 5 mostram como contornar algumas dessas dificuldades a partir de uma gestão mais eficiente (i. e. realizada mais "inteligentemente") dos recursos existentes no sistema computador e robot manipulador; todavia, muitos outros problemas persistem a nível do sistema global. Quando o leitor decide virar uma página, o seu braço segue uma dada trajectória no espaço livre até a mão entrar em contacto com o papel, ponto a partir do qual manuseia a folha mantendo uma tensão óptima, isto é, sem permitir um afrouxamento excessivo ou sem a danificar devido à aplicação de uma força demasiado elevada. Nesta "simples" acção estão subjacentes diversos e complexos procedimentos, tais como:

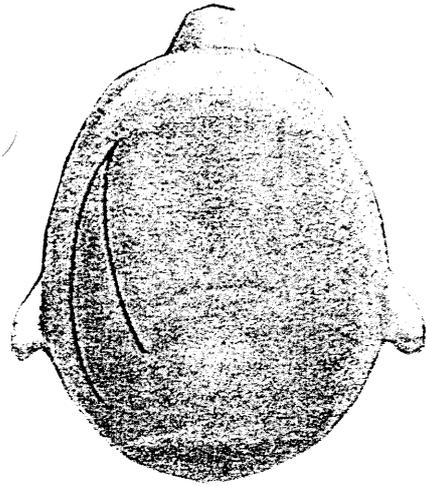
- Planeamento da trajectória no espaço livre;
- Planeamento da trajectória durante a fase de manuseamento

Planeamento do movimento do indicador direito

Correcto

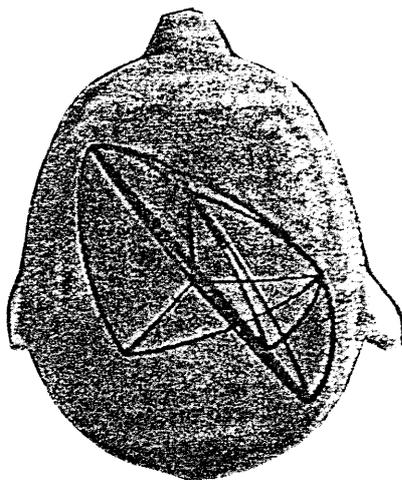


Incorrecto



Planeamento do movimento do indicador esquerdo

Correcto



Incorrecto

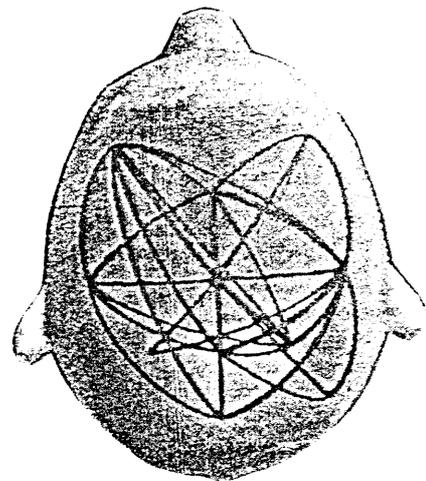


Fig. 6.1 "Mapas" cerebrais de seres humanos destros. De acordo com uma sucessão de dois estímulos, o indivíduo é levado a decidir (correcta ou incorrectamente) se deve accionar um botão com o indicador direito, ou alternativamente, com o esquerdo. Os "mapas" mostram a actividade cerebral antes da execução do movimento (i. e. mostram o planeamento do movimento). As linhas indicam "ligações cerebrais" de zonas com actividade semelhante (i. e. pontos onde o EEG apresenta covariância semelhante).

## Controlo da resposta do indicador

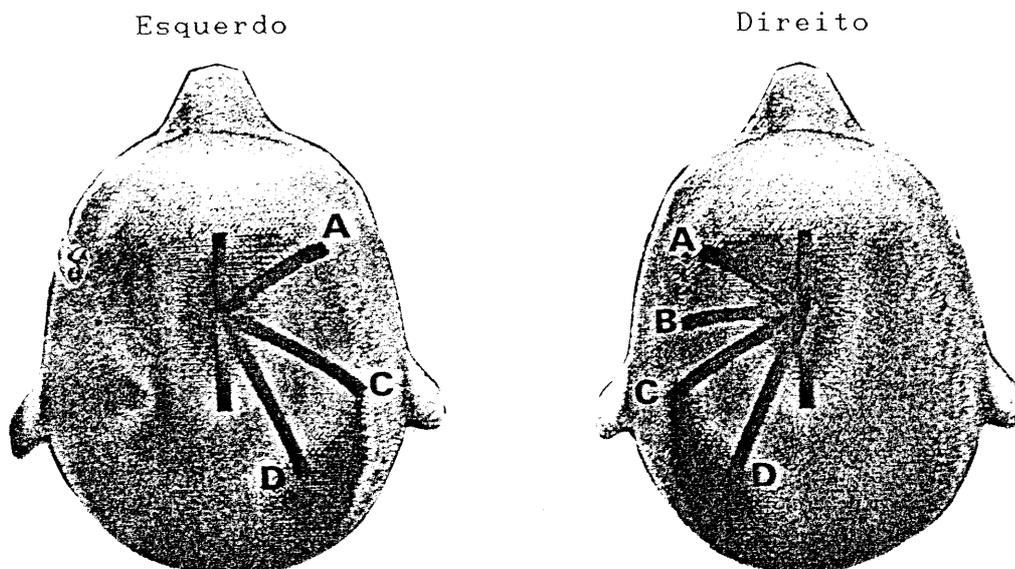


Fig. 6.2 "Mapas" cerebrais de indivíduos destros durante a fase de controlo do movimento. Note-se a simplicidade dos processos envolvidos face aqueles apresentados durante o (prévio) planeamento da tarefa (Fig. 6.1).

da folha de papel;

- Controlo das trajectórias prèviamente planeadas, e que consiste em dois estágios. No primeiro é efectuado um controlo de posição (e eventualmente velocidade), e no segundo é realizado um controlo que envolve não só posição mas também força/binário;

- A acção de controlo usa o conhecimento anteriormente adquirido sobre as características físicas da folha de papel, nomeadamente peso, espessura e tensão óptima a aplicar.

É interessante verificar que alguns destes procedimentos estão, ainda hoje, fora de qualquer enquadramento teórico e sistemático. Tão pouco existe conhecimento aprofundado sobre a metodologia associada à acção de "ligar/desligar" os múltiplos sensores (redundantes devido ao seu elevado número) envolvidos na passagem da primeira para a segunda fase da trajectória, o método

de integração dessa informação sensorial numa arquitectura de controlo, ou o algoritmo correspondente ao planeamento e controlo da actividade a partir de uma base de conhecimento com informação sobre o objecto a manipular (Figs. 6.1 e 6.2).

Na opinião do autor parte das dificuldades sentidas na mímica pretendida, passa pelo adopção única, senão mesmo quase obsessiva, de um só ponto de vista. A utilização de um controlador "clássico", seja ele linear, não-linear ou adaptativo, põe de imediato questões sobre a sua estabilidade. Sem deixar de tomar em atenção este ponto é, todavia, de considerar como pouco eficiente essa estratégia. Não será tão ou mais importante encontrar um processo que possibilite usar e actualizar (i. e. aprender) a informação armazenada numa base de conhecimento, de modo a melhorar o desempenho do controlador? E se com esta filosofia for encontrada uma estrutura de controlo que a prática mostre eficiente, mas com difícil, ou até impossível, demonstração da respectiva convergência? A verificar-se esta eventualidade, deverá ser abandonado este método em prol de outros quiçá com pior desempenho, mas que garantam teoricamente essa estabilidade?

Muito recentemente têm surgido alguns controladores obedecendo a esta filosofia, que passou a ser designada de "controlo com aprendizagem" (learning control). Alguns resultados preliminares apontam para desempenhos eficientes, embora, faltando em alguns casos, as referidas demonstrações (clássicas) de convergência. De facto, não existindo ainda uma estrutura lógica e sistemática (ou pelo menos estando-se ainda numa fase de pré-criação) capaz de tratar convenientemente informação "vaga" ou "difusa", e não tendo sido ainda descoberta uma maneira capaz de

aplicar a esta área científica a matemática predominante na actualidade, sobrevem uma falta de meios e conhecimentos para se conseguir uma "demonstração de convergência".

No fundo o problema reside num nível mais abstracto, nomeadamente no modelo que o cientista adopta para tratar o fenómeno em estudo. Grande parte da física contemporânea com implicações na engenharia, assenta em leis que jogam com os conceitos dos cálculos diferencial e integral, matricial e vectorial. Exemplos bem significativos desta afirmação são as leis de Newton e Maxwell. Como a teoria dos sistemas deriva destes conceitos base, é natural o seu envolvimento prioritário com as mesmas álgebras. Curiosamente alguns ramos da física têm sido estudados a partir de diferentes modelos. Tal é o caso da termodinâmica onde existem os pontos de vista da chamada termodinâmica clássica e da termodinâmica estatística. Assim, ao autor parece razoável questionar-se sobre a "excessiva" dependência da teoria dos sistemas dum tipo restrito de álgebra. No caso particular da análise e síntese de robots manipuladores, e adoptando esta linha de raciocínio, o autor tem vindo a desenvolver um modelo puramente baseado em conceitos estatísticos. Esse trabalho [6.8] demonstra que um tipo diferente de modelo tem vantagens para áreas específicas. Também o capítulo 5 aponta nessa direcção. Com efeito na sua concepção ideal (i. e. na forma que prescinde do uso da soma aritmética binária), a conversão de uma equação expressa na álgebra convencional, para uma descrição através da álgebra de Boole, pode ser encarada com o carácter de uma "transformada". Então, o uso da estatística, bem como a adopção da álgebra de Boole, permitiram o alargamento dos modelos

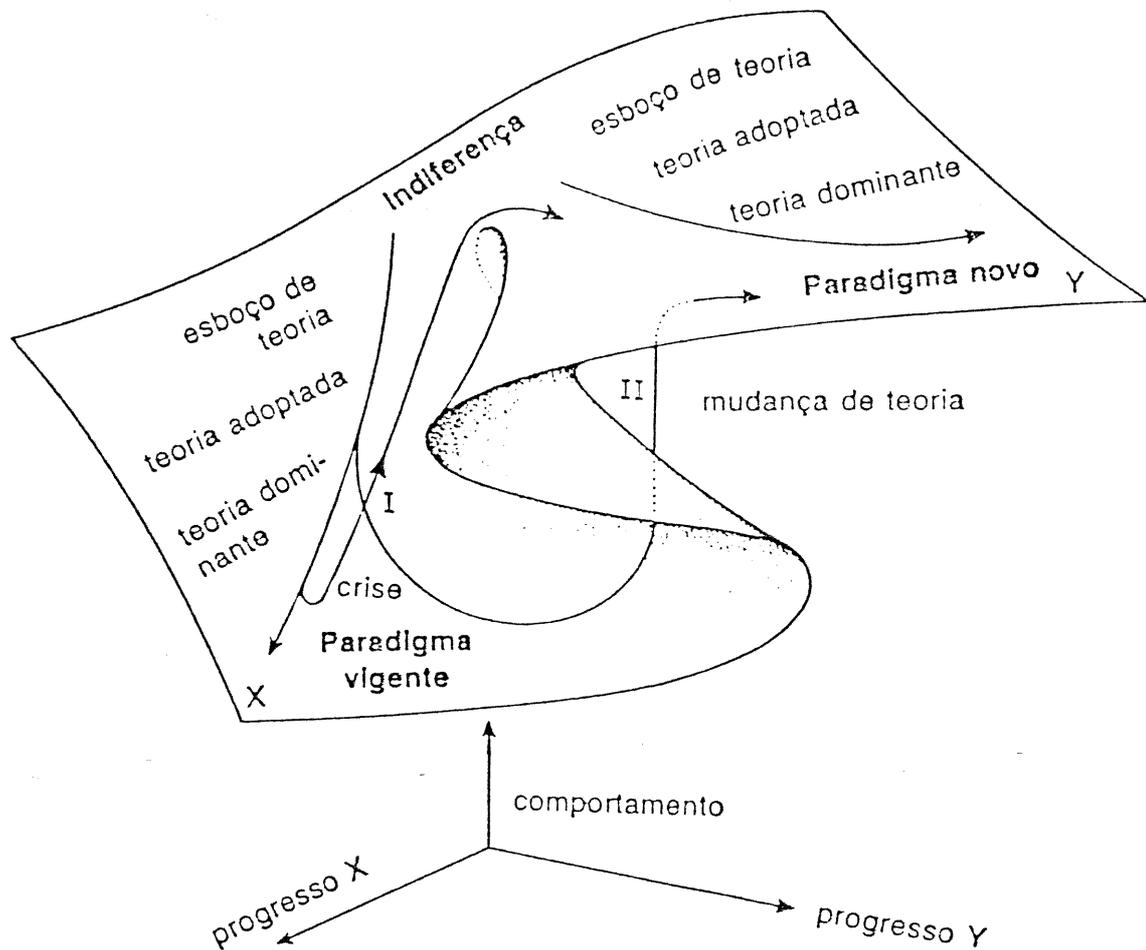


Fig. 6.3 Mudança de paradigmas numa perspectiva da teoria das catástrofes, segundo a topologia de uma cúspide.

possíveis para o sistema em estudo.

Esta perspectiva é bem ilustrada pela teoria das catástrofes [6.3] segundo a topologia de uma "cúspide" (Fig. 6.3). A passagem do paradigma vigente X, para um novo paradigma Y, pode ser conseguida lenta e suavemente (I), ou através de um procedimento descontínuo e abrupto (II), e conseqüentemente mais rápido. A modelização dos fenómenos físicos tem-se desenvolvido, profundamente, na direcção de X, tal que uma mudança na direcção de Y (isto é, a mudança da folha inferior para a folha superior da cúspide) é agora difícil, tanto mais que o paradigma X continua a dar os seus frutos.

Como é natural esta estratégia de abordagem no estudo dos robots manipuladores, não é a única ou a última resposta; contudo, é certamente uma perspectiva que para além dos resultados apresentados sugere diversas outras pistas de investigação. Por outras palavras, tendo em atenção as limitações da matemática predominante no tratamento dos fenómenos em estudo, podem-se conceber estratégias mais eficientes desde que se caminhe na direcção de uma mudança (ou pelo menos coexistência) de paradigmas. Quais os paradigmas mais adequados às "novas" realidades a investigar é, com certeza, uma pergunta ambiciosa, constituindo a presente obra um mero degrau no caminho a percorrer; todavia, ele é consequência da referida abertura científica a novos conceitos, os quais, à semelhança do que tem acontecido em muitas outras encruzilhadas da investigação científica, fomentaram algumas respostas e um muito maior número de questões. Assim, algumas conjecturas suscitadas pelo presente trabalho podem ser consideradas na seguinte lista.

- Capítulo 2

1. Desenho de novas estruturas mecânicas (incluindo melhores actuadores), mais eficientes quer no seu desempenho (em força, velocidade, etc), quer na facilidade de controlo.

- Capítulo 3

2. Uso de uma frequência de amostragem variável, em lugar de uma fixa, como meio do controlador se ajustar às necessidades do sistema

3. Uso de memória, na forma de uma base de conhecimento, de modo a proporcionar uma acção de antecipação (feedforward) que reduza o nível da realimentação, e melhore

assim o desempenho do sistema.

4. Investigação da influência de fenómenos dinâmicos, não considerados neste trabalho, tais como folgas (backlash), atritos não-lineares e flexibilidade.

- Capítulo 4

5. Considerações idênticas às referidas nos pontos 2, 3 e 4, e

6. Desenho de novas arquitecturas computacionais, mais adaptadas ao sistema a controlar.

7. Estudo das eventuais propriedades associadas à conversão álgebra clássica --> álgebra de Boole.

8. Em associação com o ponto anterior, investigação sobre as propriedades envolvidas com a descrição, a partir de dados experimentais, da dinâmica de um sistema, já que deste modo é possível "preencher" a tabela numérica que dá origem à conversão referida, sem ser necessário qualquer tipo de modelo matemático.

Para além destes pontos há ainda a referir, obviamente, o interesse da implementação prática dos métodos referidos nesta tese. Por outro lado, também o modelo estatístico, referido anteriormente, aponta quer para o ponto 1 quer para:

9. Estudo de novos algoritmos de planeamento de trajectória, que ao invés de assumirem, implicitamente, um determinado tipo de trajectória, a optimizem de acordo com as características da carga.

Detalhemos agora alguns destes pontos. Verificou-se no capítulo 2 que o uso de contrapesos melhora a estabilidade do robot manipulador, à custa de uma diminuição da sua rapidez. É assim lícito esperar, conforme sugerido no ponto 1, que outras

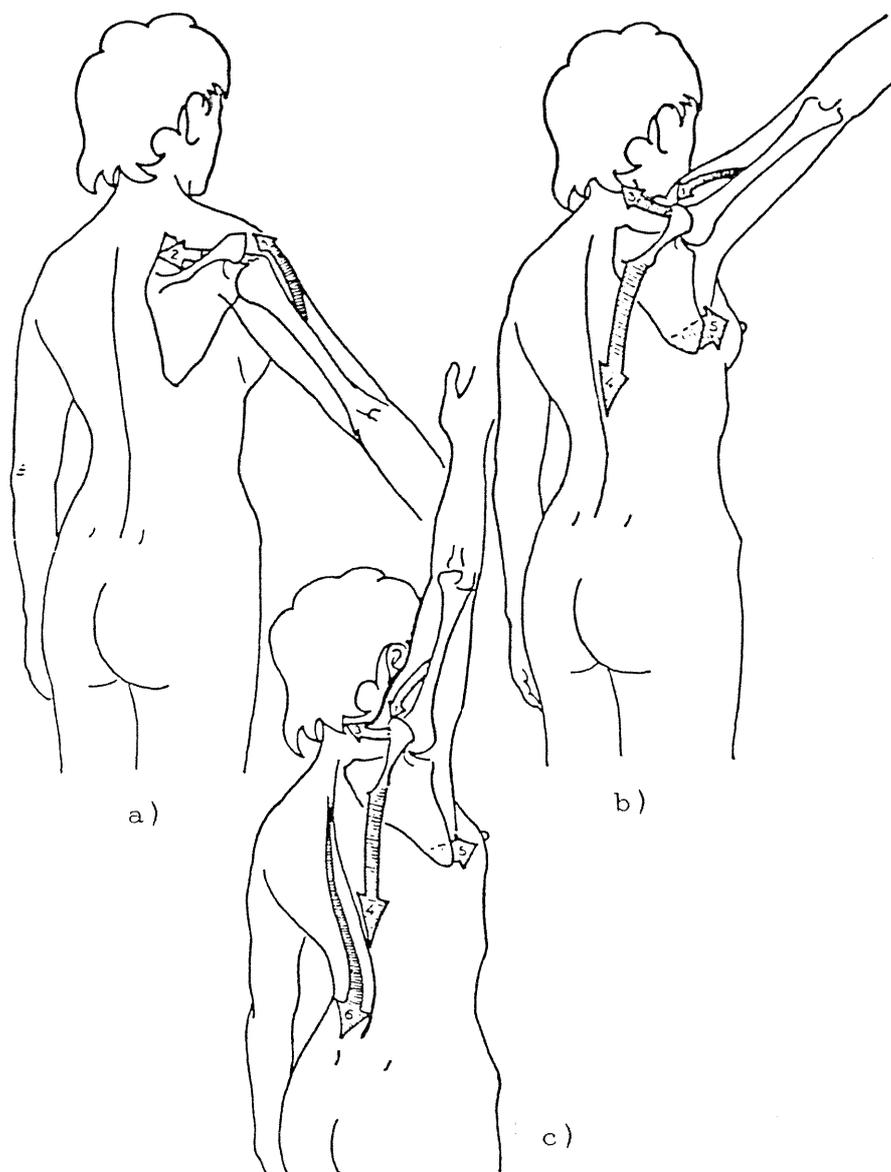


Fig. 6.4 Três tempos da abdução  
 (i. e. rotação do ombro no plano frontal)  
 Músculos: 1-deltóide, 2-supra-espinhoso, 3-trapézio,  
 4-grande-dentado, 5-espinhoso

a) Primeiro tempo: 0° a 90°  
 Rotação da articulação escapulo-humeral  
 Músculos motores: 1 e 2

b) Segundo tempo: 90° a 150°  
 Composto por dois sub-movimentos:  
 b1) Articulação esterno-costo-clavicular: 30°  
 b2) Articulação acrômio-clavicular: 30°  
 Músculos motores: 3 a 5  
 Músculos completamente retraídos: 1 e 2

c) Terceiro tempo: 150° a 180°  
 Participação do raquis no movimento  
 Músculos motores: 6  
 Músculos completamente retraídos: 1 a 5

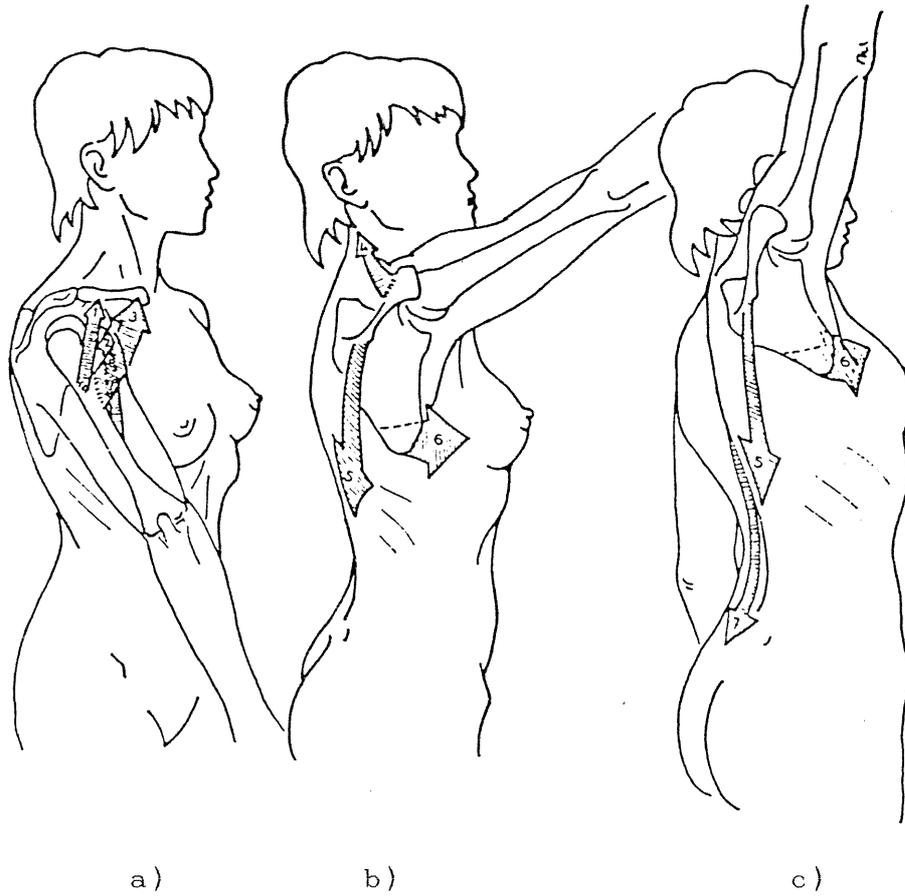


Fig. 6.5 Três tempos da antepulsão-flexão (i. e. rotação do ombro no plano sagital)  
 Músculos: 1-deltóide (feixe anterior clavicular), 2-coraco-braqueal, 3-grande-peitoral, 4 e 5-trapézio, 6-grande-dentado, 7-músculos lombares

a) Primeiro tempo: 0° a 60°  
 Rotação da articulação escapulo-humeral  
 Músculos motores: 1 a 3

b) Segundo tempo: 60° a 120°  
 Composto por dois sub-movimentos:  
     b1) Articulação esterno-costo-clavicular: 30°  
     b2) Articulação acrômio-clavicular: 30°  
 Músculos motores: 4 a 6  
 Músculos completamente retraídos: 1 a 3

c) Terceiro tempo: 120° a 180°  
 Participação do raquis no movimento  
 Músculos motores: 7  
 Músculos completamente retraídos: 1 a 6

estruturas mais elaboradas tenham intrinsecamente melhores características dinâmicas. O braço humano é uma excelente prova desta suposição. Uma observação das estruturas óssea e muscular revela que:

- Existem vários pares de músculos, sendo cada par (flexor e extensor) responsável por um raio de acção limitado, onde a sua resposta é óptima, sendo desactivado em favor de outro par para um raio de acção consecutivo (Figs. 6.4 a 6.6). Refira-se que se entende por raio de acção não sòmente o intervalo angular mas também os limites de exercício de movimentos em força ou em precisão.

- Estes músculos são "máquinas" de deslocamento limitado e de grande capacidade em regimes transitórios, ao contrário dos actuadores normalmente usados, que são meras adaptações de máquinas vocacionadas para regimes permanentes a altas velocidades.

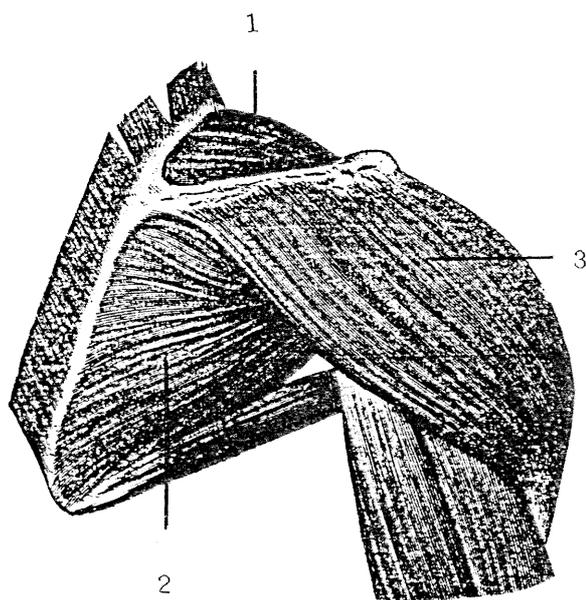


Fig. 6.6 Músculos do ombro vistos pela face posterior.  
1-supra-espinhoso, 2-infra-espinhoso, 3-deltóides

- A estrutura óssea é leve e resistente devido à sua estrutura reticulada (Fig. 6.7). Por outro lado a sua forma permite reduzir problemas de atravancamento devido aos actuadores, bem como desempenhos cinemáticos e dinâmicos extremamente eficientes.

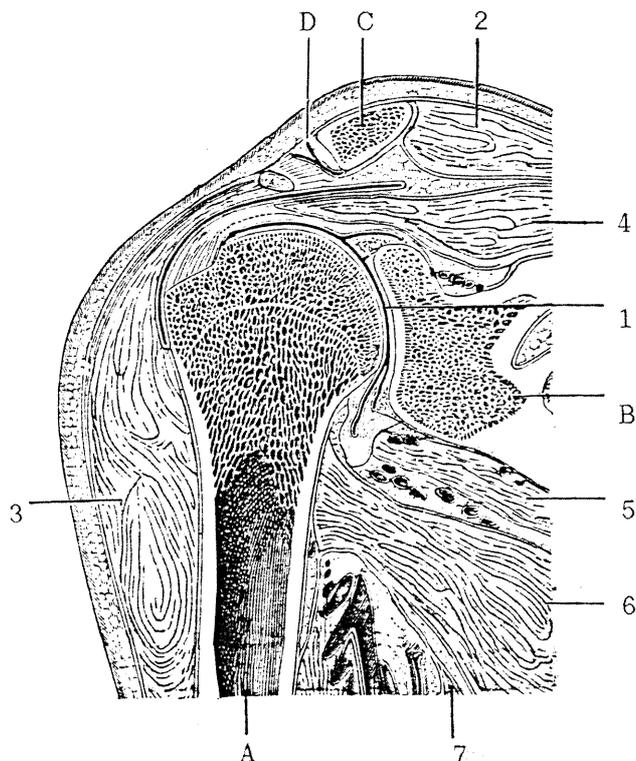


Fig. 6.7 Corte vertical do ombro direito.  
 A-húmero, B-escápula, C-clavícula,  
 D-articulação acrômio-clavicular  
 1-articulação escapulo-humeral, 2-trapézio,  
 3-deltóides, 4-supra-espinhoso, 5-infra-escapular,  
 6-grande redondo, 7-grande dorsal.

No capítulo 3 (equação 3.5) verificou-se que um controlador pelo método do binário calculado dá origem, entre outras, à equação

$$J_c(q)(\ddot{e} + K_1 \dot{e} + K_2 e) = \delta T \quad (6.1)$$

onde  $\delta T$  corresponde ao erro de modelização. Esta expressão foi

interpretada como espelhando o comportamento da malha de realimentação  $J_c(q)$ ,  $K_1$ ,  $K_2$ , que fornece o binário necessário à compensação do erro  $\delta T$ . Desta maneira provou-se ser mais "inteligente" uma arquitectura de controlo possuindo uma maior (e fixa) frequência de amostragem na malha  $J_c(q)$ ,  $K_1$ ,  $K_2$ , ficando as componentes  $C_c(q, \dot{q}) + G_c(q)$  relegadas para um cálculo a frequência inferior. É contudo natural (ponto 2), conceber-se uma estratégia possivelmente mais eficiente, segundo a qual a malha  $J_c(q)$ ,  $K_1$ ,  $K_2$ , "ajustaria" a sua frequência de cálculo consoante as necessidades - variantes no tempo - do sistema. Impondo à equação (6.1) uma condição dinâmica do tipo

$$\ddot{e} + K_1 \dot{e} + K_2 e = J_c(q)^{-1} \delta T \leq \delta_{MAX} \quad (6.2)$$

onde  $\delta_{MAX}$  é uma função a partir da qual o máximo valor possível para o período de amostragem no instante  $j+1$  ( $h_{PREV}(j+1)$ ), então este pode ser previsto tendo por base o valor óptimo no instante  $j$  ( $h_{OPT}(j)$ ), ou seja

$$h_{PREV}(j+1) = \Phi[h_{OPT}(j), \delta_{MAX}(j)] \quad (6.3)$$

Deste modo o controlador não só ajustará a sua carga computacional às necessidades manifestadas pelo sistema, mas também possibilitará a eventual execução de outras tarefas num modo de tempo repartido. Como é obvio, alguns factores adicionais devem ser tidos em consideração:

- A função  $\delta_{MAX}$  deverá estar bem adaptada ao sistema, de modo a não indicar especificações imprecisas.

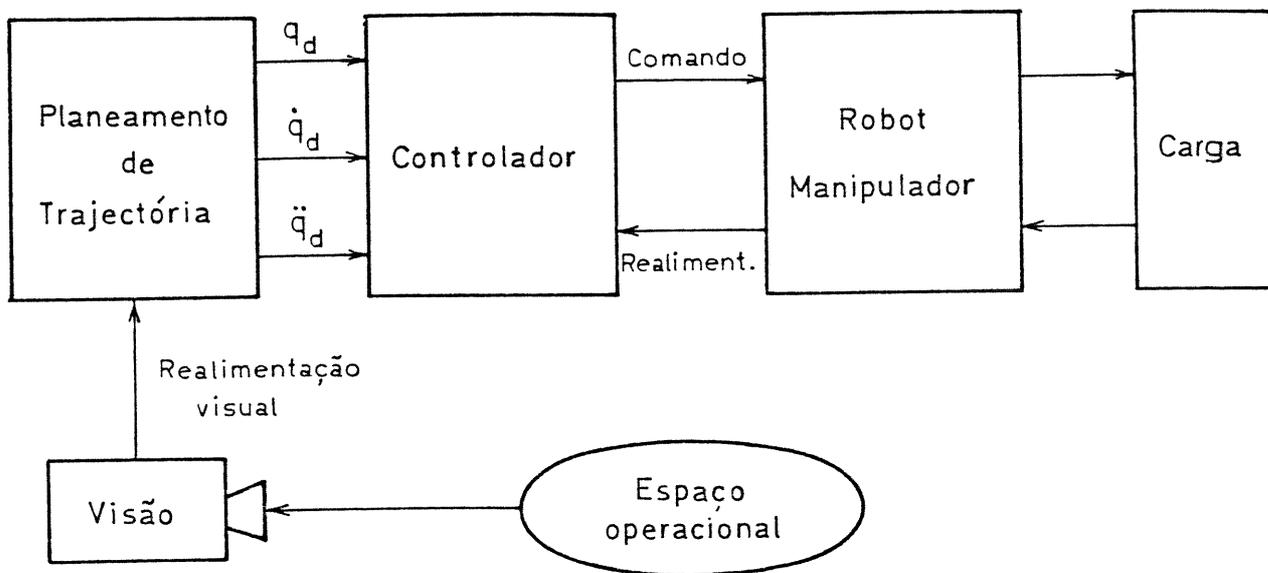
- O cálculo de  $h_{PREV}$ , através das expressões (6.2) e (6.3), introduz um peso adicional que pode acarretar efeitos indesejáveis

veis, já que reduz a capacidade computacional disponível para a malha  $J_c(q)$ ,  $K_1$ ,  $K_2$ . Para evitar este inconveniente as funções  $\delta_{MAX}$  e  $\Phi$  deverão apresentar um baixo peso computacional.

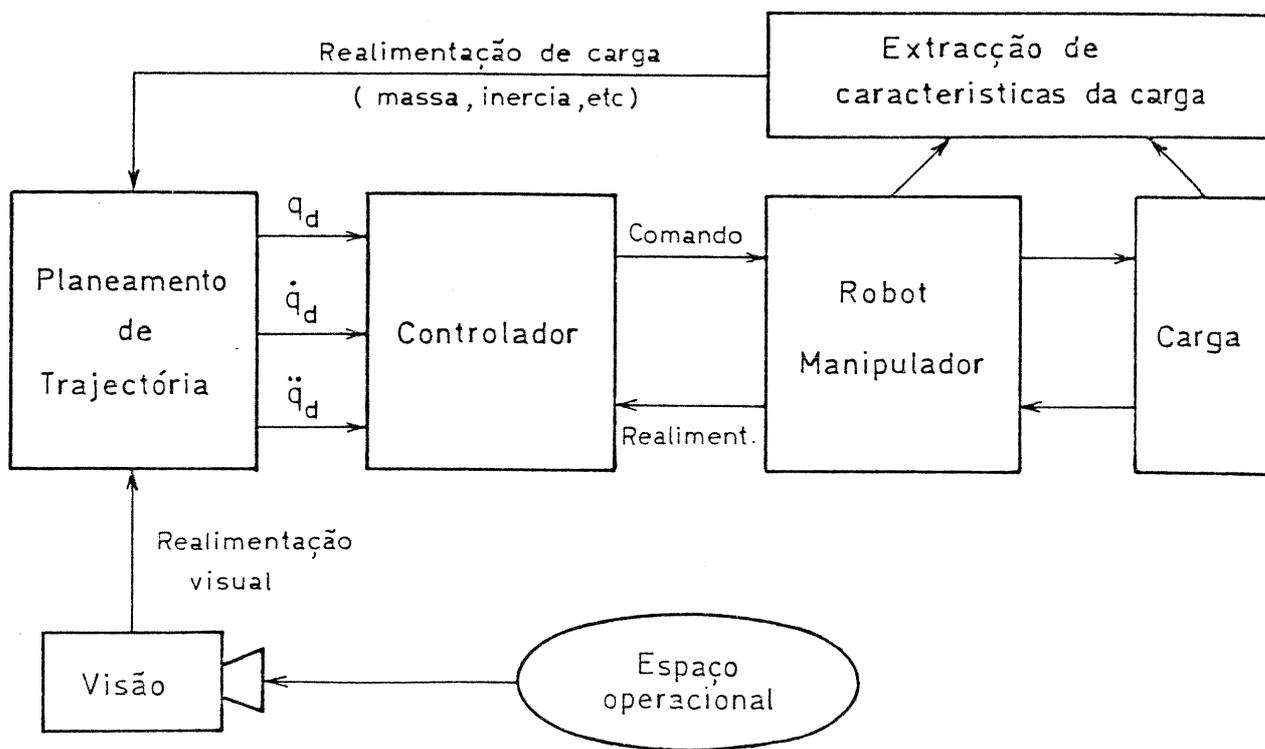
- Os valores máximo e mínimo de  $h_{PREV}$  devem estar bem especificados, de modo a saber-se de antemão se a capacidade de cálculo instalada no sistema de controlo, é suficiente nas situações de  $h_{PREV}$  ter o valor mínimo ou se, havendo capacidade de cálculo disponível, esta é compensadora pela complexidade adicional introduzida por este esquema.

Os pontos 3 a 8 são consequências directas das considerações apresentadas, pelo que dispensam a apresentação de novos argumentos. Já o ponto 9 apresenta novas considerações, a saber:

- O planeamento de trajectórias normalmente proposto na literatura científica, assenta numa primeira fase de realimentação visual, a partir da qual são detectadas zonas de impossibilidade de movimentação devido à ocorrência de colisões (obstacle avoidance), seguida de uma fase onde são consideradas trajectórias no espaço operacional que satisfazem as restrições anteriores. Contudo, este planeamento de trajectórias é "cego" relativamente às características da carga, isto é, opta-se por uma dada trajectória independentemente do tipo de carga, pelo que nesse aspecto funciona em malha aberta (Fig. 6.8a). De acordo com os resultados evidenciados pela adopção do modelo estatístico, anteriormente referido, é interessante considerar um algoritmo de planeamento de trajectórias com uma estrutura em malha fechada relativamente à carga a manipular (Fig. 6.8b). Enquanto no capítulo 4 se melhorou o desempenho do sistema, a partir de um enriquecimento do conjunto de sinais de realimentação com um de maior largura de banda (i. e. usando a aceleração para além da



a)



b)

Fig. 6.8 Planeamento de trajectórias sugerido pelo modelo estatístico.

- a) Sem realimentação relativamente à carga
- b) Com realimentação relativamente à carga.

posição e velocidade), agora um desempenho mais eficiente passa pelo enriquecimento do conjunto de sinais de realimentação, agora obtido caminhando na direcção oposta à utilizada no capítulo 4, isto é, através de uma realimentação baseada nas características da carga. Se, como na grande maioria dos casos, a carga se mantiver constante os seus parâmetros (peso, inercia, etc), então a "realimentação de carga" corresponde a uma amostragem de baixa frequência, já que só é renovada a informação quando à "mão" do robot é ordenado "AGARRAR CARGA" ou "LARGAR CARGA". Obviamente que no caso mais complexo (mas menos frequente) da variação dos parâmetros da carga, então a frequência desta malha de realimentação deverá ser consideravelmente superior.

Por outro lado o modelo estatístico apoia também o ponto 1, já que mostra as insuficiências dos actuadores correntemente usados num sistema requerendo regimes altamente transitórios intrínsecos aos robots manipuladores.

Ao terminar este trabalho aponta-se ainda como uma via de investigação o desenvolvimento de robots manipuladores, com apoio em dados biológicos, perspectiva algo "esquecida" quiçá devido à sua complexidade, mas que certamente trará compensações, pois dos melhores Mestres se obtêm os melhores ensinamentos. De facto, como se pode ler no livro do Génesis (Caps. 1 a 3), Deus criou o homem "à sua imagem e semelhança", colocando-o na terra para que ele a transformasse e dominasse, colaborando assim na obra divina de aperfeiçoamento do mundo.

Será que na nossa evolução sócio-cultural e científica, não nos caberá também a missão de criar instrumentos à nossa "imagem e semelhança", capazes de contribuirem para a construção de um mundo melhor?

## REFERÊNCIAS

- [6.1] Jacqueline Renaud  
L'Activité du Cerveau Mise en Cartes  
Science & Vie, pg. 28-35, Juillet, 1987.
- [6.2] Michael Guillen  
Pontes para o Infinito: O Lado Humano das Matemáticas  
Gradiva, 1987.
- [6.3] Sebastião J. Formosinho  
Nos Bastidores da Ciência: Resistencia dos Cientistas à  
Inovação Científica  
Gradiva, 1988.
- [6.4] Y. Suganuma, M. Ito  
Knowledge-Based Learning Control  
Control-Theory and Advanced Technology, pg. 367-383,  
vol. 2, Sept., 1986.
- [6.5] George A. Bekey, Rajko Tomovic  
Robot Control by Reflex Actions  
IEEE Int. Conf. Rob. Automat., San Francisco,  
California, 1986.
- [6.6] Suguru Arimoto, Sadao Kawamura, Fumio Miyazaki  
Bettering Operation of Robots by Learning  
J. Robotic Systems, vol. 1, pg. 123-140, Summer, 1984.
- [6.7] W. Thomas Miller III  
Sensor-Based Control of Robotic Manipulators Using a  
General Learning Algorithm  
IEEE J. Robot., Automat., vol RA-3, pg. 157-165, April  
1987.
- [6.8] J. A. Tenreiro Machado, J. L. Martins de Carvalho  
A Statistical Approach to the Analysis and Design of  
Robot Manipulators  
IEEE Int. Workshop on Intelligent Robots and Systems,  
Tokyo, Japan, 1988.
- [6.9] M. Alonso, E. J. Finn  
Fundamental University Physics: Quantum and Statistical  
Physics  
Addison-Wesley Publishing Company, USA, 1968.
- [6.10] I. A. Kapandji  
Physiologie Articulaire  
Librairie Maloine S.A., 1974, France.
- [6.11] L. Testut, A. Latarjet  
Tratado de Anatomia Humana  
Salvat Editores, S. A., 1975.

- [6.12] Neville Hogan  
Mechanical Impedance Control in Assistive Devices and Manipulators  
Proc. Joint Automatic Control Conference, San Francisco, USA, 1980.
- [6.13] Anders Wallenborg, Karl Johan Astrom  
Limit Cycle Oscillations in High Performance Robot Drives  
IEE Control '88, Oxford, UK, 1988.
- [6.14] L. Stark  
The Biological Model of Redundancy  
NATO ARW on Robots With Redudancy: Design, Sensing and Control, Salo, Italy, 1988.
- [6.15] A. Pedotti, V. V. Krishnan, L. Stark  
Optimization of Muscle-Force Sequencing in Human Locomotion  
Mathematical Biosciences, vol. 38, pg. 57-76, 1978.
- [6.16] Y. C. Tsai, A. H. Soni  
Accessible Region and Synthesis of Robot Arms  
ASME J. of Mechanical Design, vol. 103, pg. 803-811, Oct., 1981.
- [6.17] Y. Stepanenko, T. S. Sankar  
Vibro-Impact Analysis of Control Systems With Mechanical Clearance and Its Application to Robotic Actuators  
ASME J. Dynamic Syst., Meas., Contr., vol. 108, pg. 9-16, March, 1986.
- [6.18] Yoichi Tatara  
Mechanochemical Actuators  
Advanced Robotics, vol. 2, pg. 69-85, 1987.
- [6.19] D. G. Caldwell, P. M. Taylor  
Artificial Muscles as Robotic Actuators  
IFAC Symp. on Robot Control, Karlsruhe, FRG, 1988.

