

Tomaz Mikio Sasaki

**Controlador baseado em processador
digital de sinais para o ensino de controle
de robôs.**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia.

São Paulo
1997

OK

Tomaz Mikio Sasaki

**Controlador baseado em processador
digital de sinais para o ensino de controle
de robôs**

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia.

Área de Concentração:
Engenharia Mecânica

Orientador:
Jun Okamoto Jr.

Nas referências bibliográficas, faltou incluir a referência:

MATONE, R. Métodos numéricos para cinemática inversa de robôs manipuladores. São Paulo, 1991. 131p. Dissertação (Mestrado) – Escola Politécnica, Universidade de São Paulo.

Nas tabelas da página B-10, as massas e inércias apresentadas se referem aos **ligamentos** do manipulador, e não às **juntas**.

Na segunda tabela da página B-10, o título correto é “Tabela B.3”, e não “Tabela 3”.

O termo correto é “controle de estrutura variável”, e não “controle de estrutura variada”. Isto deve ser corrigido na página 38, linha 18; na página 40, linha 8; e na página 64, linha 1.

AUTOR: TOMAZ MIKIO SASAKI

TÍTULO: Controlador baseado em processador digital de sinais para o ensino de controle de robos.

DEDALUS - Acervo - EPMN



31600010146

Aos meus pais e à minha irmã.

AGRADECIMENTOS

Ao orientador Prof. Dr. Jun Okamoto Jr., pela motivação e pela orientação segura e paciente.

Ao Prof. Dr. Lucas Antonio Moscato, que me orientou no início deste trabalho.

Ao Prof. Dr. Paulo Eigi Miyagi, e a todos da Mecânica que se esforçam para incentivar e apoiar as atividades de pesquisa dentro deste departamento.

À minha namorada Myrna, pelo incentivo e compreensão.

Aos amigos Fabio, Song e Cristina, pelo estímulo e companheirismo.

Ao colega Nilson, pela sua preciosa ajuda na etapa de confecção da placa controladora.

Aos colegas Júlio Arakaki e Paulo Faggioni Filho, que colaboraram para que eu tivesse tempo para realizar este trabalho.

A todos que direta ou indiretamente colaboraram na execução deste trabalho.

SUMÁRIO

LISTA DE TABELAS	
LISTA DE FIGURAS	
RESUMO	
ABSTRACT	
INTRODUÇÃO	1
MODELAGEM E CONTROLE DE ROBÔS.....	5
2.1. MODELO CINEMÁTICO.....	6
2.1.1. <i>Problemas cinemáticos direto e inverso</i>	7
2.1.2. <i>Transformações homogêneas</i>	10
2.1.3. <i>A representação de Denavit-Hartenberg</i>	11
2.2. MODELO DINÂMICO	13
2.2.1. <i>Métodos para a obtenção do modelo dinâmico</i>	14
2.2.2. <i>Lagrange-Euler</i>	15
2.2.3. <i>Consideração dos atuadores no modelo dinâmico</i>	17
2.3. PLANEJAMENTO DE TRAJETÓRIA	20
2.3.1. <i>Trajетórias em coordenadas de junta</i>	21
2.3.2. <i>Trajетórias em coordenadas cartesianas</i>	22
2.4. CONTROLE DE POSIÇÃO	23
2.4.1. <i>Classificação dos algoritmos de controle</i>	24
2.4.2. <i>Controle convencional</i>	29
2.4.3. <i>Controle adaptativo</i>	34
2.4.4. <i>Controle de estrutura variável</i>	37
2.5. CRITÉRIOS PARA AVALIAÇÃO DO DESEMPENHO DE UM MANIPULADOR.....	43
PROCESSADORES DIGITAIS DE SINAIS EM APLICAÇÕES DE CONTROLE EM TEMPO REAL.....	47
3.1. SISTEMAS EM TEMPO REAL	47
3.1.1. <i>Hardware usado em aplicações em tempo real</i>	48
3.1.2. <i>Software em tempo real</i>	48
3.2. PROCESSADOR DIGITAL DE SINAIS.....	50
PROPOSTA DE UM SISTEMA DE CONTROLE PARA O ENSINO DE CONTROLE DE ROBÔS	54
DESENVOLVIMENTO DO TRABALHO.....	56
5.1. DESENVOLVIMENTO DO HARDWARE	56
5.2. DESENVOLVIMENTO DO SOFTWARE	58
5.2.1. <i>Software do microcontrolador</i>	58
5.2.2. <i>Software do DSP</i>	59
RESULTADOS E DISCUSSÃO.....	61
6.3. AVALIAÇÃO DA CAPACIDADE DE EXECUÇÃO DOS ALGORITMOS	61
6.4. EXEMPLOS DE MOVIMENTOS REALIZADOS SOB O CONTROLE DO SISTEMA DESENVOLVIDO	63
CONCLUSÃO	71

REFERÊNCIAS BIBLIOGRÁFICAS74

APÊNDICE A

APÊNDICE B

APÊNDICE C

LISTA DE TABELAS

TABELA 2.1. CONTROLADORES CONVENCIONAIS E SUAS RESPECTIVAS LEIS DE CONTROLE.	30
TABELA 2.2. EXEMPLOS DE CONTROLADORES ADAPTATIVOS.	37
TABELA 6.1. RESULTADOS DOS TESTES.	62

LISTA DE FIGURAS

FIGURA 1.1. MANIPULADOR.	1
FIGURA 1.2. DIAGRAMA DE BLOCOS DE UM ROBÔ.	2
FIGURA 2.1. UM MANIPULADOR E SUAS PARTES.	6
FIGURA 2.2. EXEMPLO DE UM MANIPULADOR COM 2 GRAUS DE LIBERDADE.	8
FIGURA 2.3. CINEMÁTICA DIRETA E INVERSA.	10
FIGURA 2.4. A POSIÇÃO DE UM PONTO EM RELAÇÃO A DOIS SISTEMAS DE COORDENADAS.	10
FIGURA 2.5. PARÂMETROS DE DENAVIT-HARTENBERG.	12
FIGURA 2.6. ENTRADAS E SAÍDAS DE UM MANIPULADOR.	13
FIGURA 2.7. SISTEMA FORMADO PELO CONJUNTO ATUADORES + MANIPULADOR.	18
FIGURA 2.8. DIAGRAMA N-S DO ALGORITMO QUE GERA AS TRAJETÓRIAS EM COORDENADAS DE JUNTA.	21
FIGURA 2.9. EXEMPLO DE UMA CURVA QUE INTERPOLA AS POSIÇÕES INICIAL E FINAL DE UMA JUNTA.	22
FIGURA 2.10. DIAGRAMA N-S DO ALGORITMO QUE GERA AS TRAJETÓRIAS EM COORDENADAS CARTESIANAS.	23
FIGURA 2.11. DIAGRAMA DE BLOCOS DO RESOLVED MOTION RATE CONTROL.	26
FIGURA 2.12. CONTROLE PD.	31
FIGURA 2.13. DIAGRAMA DE BLOCOS GENÉRICO DE UM CONTROLE ADAPTATIVO.	36
FIGURA 2.14. PLANO DE FASE PARA $\psi = \alpha$	38
FIGURA 2.15. PLANO DE FASE PARA $\psi = -\alpha$	39
FIGURA 2.16. PLANO DE FASE PARA ψ DESCONTÍNUA.	39
FIGURA 2.14. EXEMPLO DE APRESENTAÇÃO DE RESULTADOS PARA UMA ENTRADA DEGRAU.	44
FIGURA 2.15. EXEMPLO DE APRESENTAÇÃO DE RESULTADOS UTILIZANDO TRAJETÓRIAS DE REFERÊNCIA.	45
FIGURA 2.16. EXEMPLO DE MEDIDA DE DESEMPENHO UTILIZANDO A NORMA L^2 DO ERRO DE POSIÇÃO VERSUS A FREQUÊNCIA NOMINAL DA TRAJETÓRIA DE REFERÊNCIA (WHITCOMB; RIZZI; KODITSCHKEK, 1993).	46
FIGURA 2.17. EXEMPLO DE MEDIDA DE DESEMPENHO UTILIZANDO A MÉDIA E O DESVIO PADRÃO DA NORMA L^2 DO ERRO DE POSIÇÃO (10 REPETIÇÕES PARA CADA CONTROLADOR) (WHITCOMB; RIZZI; KODITSCHKEK, 1993).	46
FIGURA 3.1. DIAGRAMA DE FLUXO DE SINAL DE UM ALGORITMO PID NA FORMA DISCRETA.	52
FIGURA 3.2. EXEMPLO DE UM FILTRO IIR.	52
FIGURA 4.1. DIAGRAMA DE BLOCOS DE UM ROBÔ.	54
FIGURA 5.1. DIAGRAMA DE BLOCOS DO SISTEMA DE CONTROLE PROPOSTO.	56
FIGURA 5.2. A IMPLEMENTAÇÃO DO SISTEMA DE CONTROLE.	57
FIGURA 5.3. SISTEMA PARA ENSINO E TREINAMENTO EM ROBÓTICA.	58
FIGURA 5.4. UM CICLO DE CONTROLE DO DSP.	60
FIGURA 6.1. RESPOSTA DO MOTOR DE UMA DAS JUNTAS A UMA ENTRADA DEGRAU (CONTROLADOR P, COM $K_p=1.5$).	64
FIGURA 6.2. RESPOSTA DO MOTOR DE UMA DAS JUNTAS A UMA ENTRADA DEGRAU (CONTROLADOR P, COM $K_p=7$).	65
FIGURA 6.3. RESPOSTA DO MOTOR DE UMA DAS JUNTAS A UMA ENTRADA DEGRAU (CONTROLADOR P, COM $K_p=7$).	65
FIGURA 6.4. RESPOSTA DO MOTOR DA JUNTA 3 A UMA RAMPA (CONTROLADOR PD, COM $K_p=7$ E $K_d=1.5$).	66
FIGURA 6.5. RESPOSTA DO MOTOR DA JUNTA 3 A UMA RAMPA (CONTROLADOR PD, COM $K_p=10$ E $K_d=3.5$).	66
FIGURA 6.6. RESPOSTA DA JUNTA 3 A UMA ENTRADA RAMPA (CONTROLADOR PD, COM $K_p=7$ E $K_d=3.5$).	67
FIGURA 6.7. RESPOSTA DA JUNTA 3 A UMA RAMPA MAIS ACENTUADA (CONTROLADOR PD, COM $K_p=7$ E $K_d=1.5$).	68
FIGURA 6.8. RESPOSTA DA JUNTA 3 A UMA ENTRADA RAMPA (CONTROLADOR PD, COM $K_p=6$ E $K_d=3.5$).	68
FIGURA 6.9. RESPOSTA DA JUNTA 3 A UMA RAMPA (CONTROLADOR PD, COM $K_p=7$ E $K_d=1.5$).	69

FIGURA 6.10. RESPOSTA DA JUNTA 3 A UMA ENTRADA RAMPA (CONTROLADOR PD, COM $K_P=6$ E $K_D=3.5$).	
.....	69
FIGURA 6.11. RESPOSTA DA JUNTA 3 A UMA RAMPA (CONTROLADOR PD, COM $K_P=10.5$ E $K_D=3.5$).	70

RESUMO

Sistemas de controle de manipuladores utilizam-se de algoritmos em tempo real para realizar o planejamento de trajetória e o controle de posição das juntas do manipulador. A implementação de algumas técnicas de controle de posição que usam explicitamente o modelo dinâmico requer um grande número de cálculos em tempo real. No caso das trajetórias serem geradas no espaço cartesiano, a cinemática inversa também implica em um grande número de cálculos. A implementação destes algoritmos em um sistema real é difícil pelo fato da maior parte dos sistemas didáticos comerciais não permitirem a alteração dos seus algoritmos de controle. Além disso, é necessário um sistema com desempenho suficiente para realizar os cálculos envolvidos em algumas técnicas de controle. Este trabalho propõe o desenvolvimento de um sistema de controle com a flexibilidade e a capacidade numérica necessárias para a implementação de diversos algoritmos de controle de robôs. Um controlador baseado em um DSP (*Digital Signal Processor*) é proposto para ser usado para o estudo de técnicas de controle para manipuladores. Testes foram realizados para verificar a capacidade do controlador proposto para executar os algoritmos desejados. Exemplos de resultados que podem ser obtidos com o sistema proposto também são apresentados.

ABSTRACT

Manipulators control systems use real time algorithms to perform the trajectory planning and position control of each manipulator's joint. The implementation of some techniques of position control that uses dynamic model explicitly requires a great number of calculations in real time. In the case of trajectories generated in cartesian space, the inverse kinematics also involves a great number of calculations. The implementation of these algorithms is difficult because most of educational systems in market do not permit the alteration of control algorithms. Moreover, it is necessary a system with enough performance in order to perform the calculations involved in some control techniques. This work proposes the development of a control system with flexibility and capacity to implement several control algorithms of robots. A controller based in a DSP is proposed to study control techniques for manipulators. Tests were performed to verify the capacity of the controller to execute the desired algorithms. Examples of results that can be obtained with this system also are presented.

Capítulo 1

Introdução

Um manipulador mecânico pode ser modelado dinamicamente como um sistema que tem como entradas as forças (para juntas lineares) e os torques (para juntas rotacionais) aplicados às juntas, e como saídas as posições e velocidades das articulações (figura 1.1).

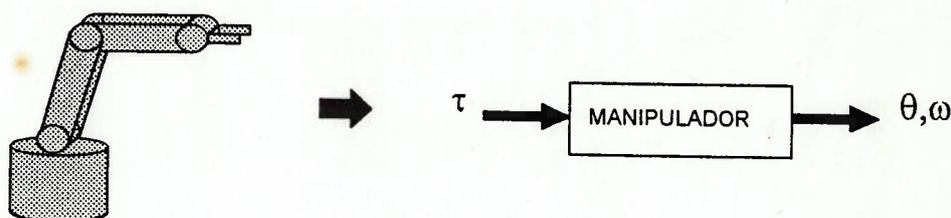


Figura 1.1. Manipulador.

As equações matemáticas que descrevem o modelo dinâmico do manipulador são bastante complexas, e caracterizam o manipulador como um sistema multivariável, com acoplamento entre as suas diversas entradas e saídas, e altamente não-linear. Os termos que compõem as suas equações matemáticas exigem um grande número de cálculos aritméticos e trigonométricos.

Do ponto de vista do controle dos movimentos do manipulador, um robô pode ser decomposto em outros sub-sistemas (figura 1.2).

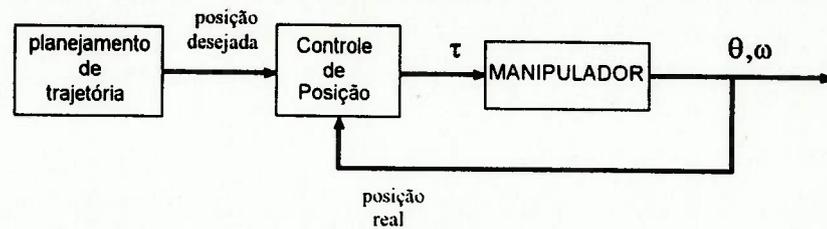


Figura 1.2. Diagrama de blocos de um robô.

O bloco responsável pelo planejamento de trajetória gera as posições desejadas para as juntas do manipulador a cada instante. Geralmente, estas posições são descritas utilizando as variáveis das juntas, já que normalmente os sensores de posição do robô estão nas juntas do manipulador. Outra opção seria realizar as trajetórias especificando as posições do punho no espaço cartesiano (por exemplo, para que a garra do robô descreva uma reta no espaço). No entanto, isto geralmente implica na necessidade de transformar cada ponto definido no espaço cartesiano para a posição equivalente definida nas variáveis das juntas. Esta operação é chamada de cinemática inversa.

O bloco de controle de posição, que tem como entradas as posições desejadas (geradas pelo planejamento de trajetória) e as posições reais das juntas do manipulador, é responsável por alimentar os atuadores de forma que o manipulador se movimente de acordo com as posições desejadas. Existem diversos algoritmos projetados para este fim. Devido à própria complexibilidade matemática do sistema a ser controlado (o manipulador), algumas técnicas de controle de posição envolvem um grande número de operações aritméticas e trigonométricas. Alguns algoritmos que apresentam características desejáveis, como a adaptação do sistema de controle a variações nos parâmetros do manipulador, necessitam ainda de um maior número de operações matemáticas.

Os algoritmos de planejamento de trajetória e controle de posição são implementados digitalmente; isto quer dizer que a leitura das saídas e a escrita nas entradas dos atuadores são feitas periodicamente, sendo este período chamado de *tempo de amostragem*. Dentro deste período de amostragem é necessário realizar uma série de operações matemáticas, sendo que o número destas operações depende dos algoritmos que estão sendo utilizados para gerar as trajetórias e controlar a posição do manipulador. Se, por exemplo, as trajetórias forem geradas no espaço cartesiano, será necessário realizar também a cinemática inversa, o que vai implicar num maior número de operações; da mesma forma, se for utilizado um controle de posição adaptativo (que será visto mais adiante), o número de operações necessárias dentro de um período de amostragem aumentará.

As principais razões que motivaram este trabalho são as seguintes:

- O estudo de uma determinada técnica de controle através de resultados obtidos experimentalmente permite uma apreciação mais completa que aquela feita através de simulações. Em um robô real, diversos fatores que não estão presentes nas simulações influenciam os resultados: dinâmicas não-modeladas, ruído, folgas (*backlash*), não-linearidade dos atuadores e saturação, por exemplo.
- Em geral, os sistemas didáticos comerciais para o ensino de robótica não permitem que o usuário interfira no algoritmo de controle de posição do sistema.
- A implementação do sistema de controle de movimentos de um robô geralmente exige um grande número de cálculos em tempo real, principalmente quando são necessárias algumas características desejáveis, como gerar a trajetória no espaço cartesiano ou rodar um algoritmo de controle de posição mais eficiente.

- O surgimento de microprocessadores cada vez mais velozes e baratos tem tornado cada vez mais viável a implementação de sistemas capazes de executar um grande número de cálculos em tempo real. Em particular, os DSP's (*Digital Signal Processor*) têm-se mostrado uma opção adequada para aplicações de controle (TEXAS INSTRUMENTS, 1990a).

Desta forma, a proposta deste trabalho é desenvolver um sistema de controle para robôs manipuladores que tenha a flexibilidade e a capacidade numérica necessárias para permitir a implementação de diversos algoritmos de controle de posição, sendo possível a realização de experiências com os mesmos. Para isto, é projetado um controlador utilizando um Processador Digital de Sinais.

Este trabalho está dividido da seguinte forma: os capítulos 2 e 3 apresentam a revisão bibliográfica e os preliminares teóricos dos tópicos “Modelagem e Controle de Robôs” e “Processadores Digitais de Sinais em Aplicações de Controle”; o capítulo 4 mostra a solução proposta; o capítulo 5 apresenta a implementação da proposta; o capítulo 6 mostra os testes realizados e discute os resultados obtidos; o capítulo 7 apresenta as conclusões e as sugestões para futuros trabalhos.

Capítulo 2

Modelagem e controle de robôs

O projeto de um sistema de controle geralmente envolve os seguintes passos:

- obter o modelo matemático do sistema que será controlado. Isto se faz necessário, pois a maioria das técnicas de controle necessita de um certo grau de conhecimento do modelo do sistema. No caso do manipulador, isto envolve a obtenção de seus modelos cinemático e dinâmico;
- escrever um algoritmo que produz as trajetórias que o manipulador deve seguir. Como já foi explicado anteriormente, as trajetórias podem ser descritas no espaço cartesiano ou no espaço de juntas;
- escolher a técnica de controle e obter as equações da lei de controle.

Há diversas referências que apresentam uma visão geral sobre estes três primeiros itens. Entre elas, destaca-se PAUL (1981), um livro clássico que devido ao tempo não incorpora informações mais recentes, o que é particularmente sentido no capítulo sobre controle de posição. Ainda assim, é um livro recomendado para uma introdução nos conceitos básicos da robótica. Outras referências são ASADA; SLOTINE (1986), FU et. al. (1987), SPONG; VIDYASAGAR (1989), e MARTÍNEZ (1991).

2.1. Modelo cinemático

Um braço articulado é formado por elos (*links*), juntas (*joints*), atuadores (*actuators*) de juntas, sensores de posição de juntas, punho, e alguma garra ou ferramenta (*end-effector*) (SALANT, 1991).

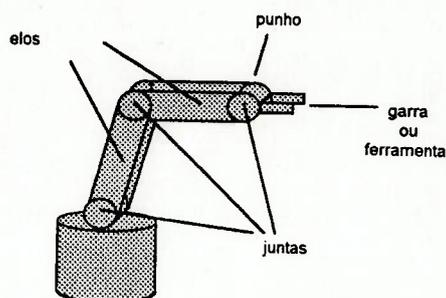


Figura 2.1. Um manipulador e suas partes.

Os elos são as partes rígidas de um braço de robô. Juntas são as partes do braço que permitem uma conexão móvel entre dois elos. As juntas podem ser lineares ou rotacionais. Quando cada junta tem conexão com no máximo dois elos, o manipulador é denominado uma cadeia cinemática aberta.

A garra ou ferramenta que está conectada ao punho é o dispositivo utilizado pelo robô para cumprir sua tarefa (segurar peças, soldar, pintar, etc.). Neste texto, será utilizado genericamente o termo garra para fazer referência a este dispositivo.

Manipuladores são geralmente descritos como tendo um certo número de graus de liberdade. Em robótica, o número de graus de liberdade é o número de movimentos distintos que o braço pode realizar. Normalmente o número de graus de liberdade é igual ao número de juntas.

Supondo um sistema de coordenadas cartesiano fixo no ambiente de trabalho, pode-se observar que o movimento relativo nas juntas do manipulador altera as posições e orientações dos elos em relação ao sistema de coordenadas. Na maioria das aplicações

em robótica, o interesse maior recai sobre a descrição espacial da garra do manipulador em relação ao sistema de coordenadas fixo. No entanto, a informação da posição do manipulador geralmente é obtida através de sensores que estão ligados às suas juntas.

O modelo cinemático de um manipulador descreve analiticamente a sua geometria e os seus movimentos, sem considerar as forças e torques que causam estes movimentos. Este modelo permite, entre outras coisas, achar a relação entre as coordenadas de junta do manipulador e a posição e orientação da garra no espaço cartesiano.

PAUL (1981), ASADA; SLOTINE (1986), FU et. al. (1987), e SPONG; VIDYASAGAR (1989) dedicam pelo menos um capítulo para explicar os problemas cinemáticos direto e inverso e o método de Denavit-Hartenberg para a obtenção do modelo cinemático de uma cadeia cinemática aberta.

2.1.1. Problemas cinemáticos direto e inverso

Para entender os problemas cinemáticos direto e inverso será utilizado o exemplo a seguir. A figura 2.2 mostra um manipulador de dois graus de liberdade, com juntas rotacionais e elos com comprimentos l_1 e l_2 . É comum estabelecer um sistema de coordenadas fixo em relação ao qual todos os objetos, inclusive o manipulador, são referenciados. Neste caso, o sistema $O_0X_0Y_0$ é o sistema de coordenadas fixo.

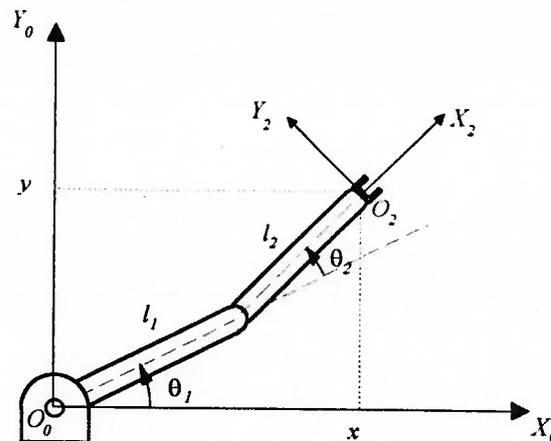


Figura 2.2. Exemplo de um manipulador com 2 graus de liberdade.

A posição da garra em relação a $O_0X_0Y_0$ é dada pelas coordenadas (x, y) . Para a orientação, é definido um outro sistema de coordenadas $O_2X_2Y_2$. Este novo sistema é fixo à garra, movendo-se juntamente com ela. Desta forma, a orientação da garra fica definida pela direção dos eixos X_2 e Y_2 em relação aos eixos X_0 e Y_0 .

Em geral os pontos no ambiente de trabalho são descritos em relação ao sistema de coordenadas fixo. Desta forma, para programar um movimento no robô, o modo mais imediato é definir as posições (x, y) pelas quais a garra deve passar. No entanto, o manipulador é capaz de informar sua própria posição através de sensores instalados nas juntas 1 e 2, podendo medir diretamente os ângulos θ_1 e θ_2 . Assim, torna-se necessário expressar as posições pelas quais a garra deve passar também em termos dos ângulos das juntas.

O problema cinemático direto consiste em, dada uma posição do manipulador definida pelos seus ângulos de junta θ_1 e θ_2 , obter a posição e a orientação da garra em relação ao sistema de coordenadas fixo. No exemplo, dados os ângulos θ_1 e θ_2 , a posição (x, y) da garra é dada pelas expressões

$$\begin{aligned}x &= l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \\y &= l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2)\end{aligned}\quad (2.1)$$

A orientação da garra é dada pela direção dos eixos X_2 e Y_2 em relação aos eixos X_0 e Y_0 , de acordo com as expressões

$$\begin{aligned}\mathbf{x}_2 &= \cos(\theta_1 + \theta_2)\mathbf{x}_0 + \sin(\theta_1 + \theta_2)\mathbf{y}_0 \\ \mathbf{y}_2 &= -\sin(\theta_1 + \theta_2)\mathbf{x}_0 + \cos(\theta_1 + \theta_2)\mathbf{y}_0\end{aligned}\quad (2.2)$$

onde $\mathbf{x}_0, \mathbf{y}_0, \mathbf{x}_2$ e \mathbf{y}_2 são os vetores unitários correspondentes aos eixos X_0, Y_0, X_2 e Y_2 , respectivamente. Escrevendo na forma matricial, define-se uma matriz de orientação.

$$\begin{bmatrix} \mathbf{x}_2 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \end{bmatrix} = \begin{bmatrix} n_x & s_x \\ n_y & s_y \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{y}_0 \end{bmatrix}\quad (2.3)$$

O problema cinemático inverso consiste no problema inverso: dada a posição e a orientação da garra em relação ao sistema de coordenadas fixo, obter os ângulos θ_1 e θ_2 .

A solução do problema neste exemplo é

$$\begin{aligned}\theta_1 &= \text{atan2}(y - l_2 n_y, x - l_2 n_x) \\ \theta_2 &= \text{atan2}(n_y, n_x) - \theta_1\end{aligned}\quad (2.4)$$

onde a função $\text{atan2}(y, x)$ é a função $\arctan\left(\frac{y}{x}\right)$, definida de maneira que

$$\text{atan2}(y, x) = \begin{cases} 0^\circ \leq \theta \leq 90^\circ & \text{para } x \geq 0, y \geq 0 \\ 90^\circ < \theta < 180^\circ & \text{para } x < 0, y \geq 0 \\ 180^\circ < \theta < 270^\circ & \text{para } x < 0, y < 0 \\ 270^\circ \leq \theta < 360^\circ & \text{para } x \geq 0, y < 0 \end{cases}\quad (2.5)$$

Desta forma ficam delineadas as duas questões principais de interesse na cinemática de manipuladores: a cinemática direta e a inversa (figura 2.3) (FU et. al., 1987).

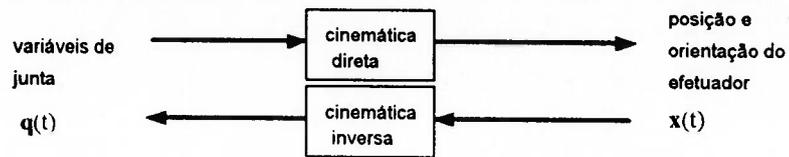


Figura 2.3. Cinemática direta e inversa.

2.1.2. Transformações homogêneas

Estes problemas cinemáticos (direto e inverso) tornam-se mais complexos com o aumento do número de graus de liberdade do manipulador. Para ajudar a escrever as equações de transformação de coordenadas utiliza-se um artifício matemático.

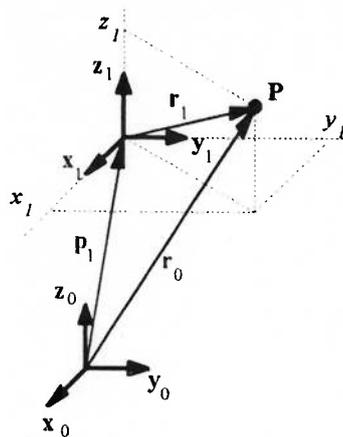


Figura 2.4. A posição de um ponto em relação a dois sistemas de coordenadas.

Na ilustração da figura 2.4, seja a posição do ponto P em relação aos sistemas de coordenadas ortonormais (x_0, y_0, z_0) e (x_1, y_1, z_1) dada respectivamente pelos seguintes vetores

$$r_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix}, \quad r_1 = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} \tag{2.6}$$

A posição da origem do sistema (x_1, y_1, z_1) em relação ao sistema (x_0, y_0, z_0) é dada pelo vetor p_1 .

$$\mathbf{p}_1 = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad (2.7)$$

Os eixos ortonormais (x_1, y_1, z_1) podem ser escritos como

$$\begin{aligned} x_1 &= n_x x_0 + n_y y_0 + n_z z_0 \\ y_1 &= s_x x_0 + s_y y_0 + s_z z_0 \\ z_1 &= a_x x_0 + a_y y_0 + a_z z_0 \end{aligned} \quad (2.8)$$

A matriz de transformação homogênea entre estes dois sistemas de coordenadas é dada por

$${}^0\mathbf{A}_1 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Esta matriz relaciona os vetores \mathbf{r}_0 e \mathbf{r}_1 da seguinte forma

$$\mathbf{r}_0 = {}^0\mathbf{A}_1 \mathbf{r}_1 \quad (2.10)$$

ou seja, dadas as coordenadas do ponto \mathbf{P} expressas no sistema 1, permite obter as coordenadas do mesmo ponto expressas no sistema 0.

2.1.3. A representação de Denavit-Hartenberg

A representação de Denavit-Hartenberg fornece um método sistemático para estabelecer um sistema de coordenadas em cada elo de uma cadeia articulada (FU et. al., 1987, cap. 2).

Uma vez que os sistemas de coordenadas tenham sido estabelecidos para cada elo segundo este método, cada uma das matrizes de transformação homogênea que relaciona o i -ésimo sistema de coordenadas ao $(i-1)$ -ésimo sistema de coordenadas pode

ser obtida. A posição relativa entre dois sistemas de coordenadas adjacentes fica determinada por 4 parâmetros. A figura 2.5 mostra o significado geométrico destes parâmetros.

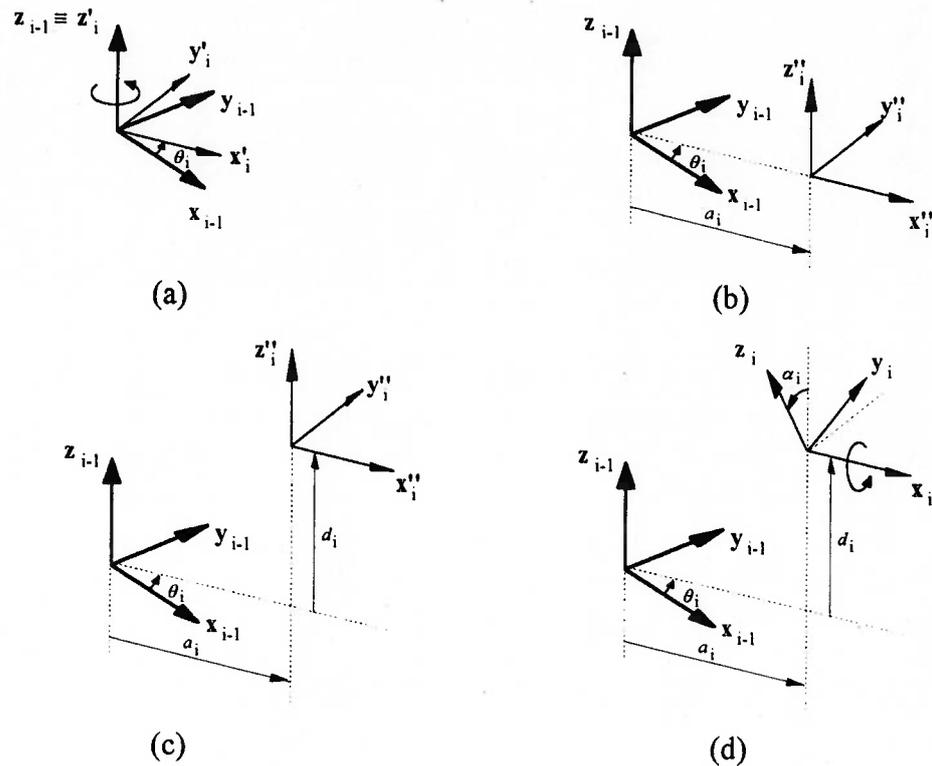


Figura 2.5. Parâmetros de Denavit-Hartenberg.

Utilizando o método de Denavit-Hartenberg, os parâmetros a_i e α_i resultam em constantes que dependem somente da geometria do manipulador. Quanto aos parâmetros d_i e θ_i , somente um deles varia com o movimento do manipulador; se a junta for prismática, d_i varia e θ_i é constante; se a junta for rotativa, θ_i varia e d_i é constante. Desta forma, para cada junta i , apenas um parâmetro é variável; este parâmetro é chamado de coordenada generalizada, que é representada por q_i .

Uma vez obtidos os parâmetros de Denavit-Hartenberg, as matrizes de transformação homogênea entre os sistemas de coordenadas números i e $(i-1)$ são dadas por

$${}^{i-1}A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

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

onde n é o número de graus de liberdade do manipulador.

As matrizes de transformação homogênea permitem a resolução da cinemática direta. Elas também podem ser usadas para montar as equações que permitem a resolução da cinemática inversa. Para a resolução desta última existem também métodos numéricos.

PAUL (1981), FU et. al. (1987), e SPONG; VIDYASAGAR (1989) buscam resolver a cinemática inversa analiticamente. MATONE (1991) fez um estudo da resolução deste problema utilizando métodos numéricos.

2.2. Modelo dinâmico

Um manipulador pode ser visto como um sistema cujas entradas são as forças (ou torques) aplicadas às juntas, e cujas saídas são as velocidades e posições das juntas.



Figura 2.6. Entradas e saídas de um manipulador.

As equações que relacionam as entradas e saídas de um manipulador podem ser escritas na forma matricial (MARTÍNEZ, 1991):

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (2.12)$$

onde:

q e \dot{q} são vetores $n \times 1$ com as posições e velocidades das juntas, respectivamente;

$\mathbf{H}(\mathbf{q})$ é uma matriz $n \times n$ denominada matriz de inércia;

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ é uma matriz $n \times n$, denominada matriz de forças centrífugas e de Coriolis;

$\mathbf{g}(\mathbf{q})$ é um vetor $n \times 1$ com as forças externas generalizadas.

2.2.1. Métodos para a obtenção do modelo dinâmico

Existem dois métodos básicos para o cálculo das equações dinâmicas de um manipulador: o método de Lagrange-Euler e o método de Newton-Euler (FU et. al., 1987).

O primeiro se baseia na aplicação direta da formulação de Lagrange-Euler para sistemas não-conservativos. Este método permite a obtenção explícita das matrizes $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ e $\mathbf{g}(\mathbf{q})$, que compõem o modelo dinâmico do manipulador. Neste caso, diz-se que as equações são obtidas na forma fechada. Isto é vantajoso, principalmente quando a lei de controle utiliza explicitamente o modelo matemático do sistema.

O segundo método, de Newton-Euler, apresenta a vantagem de exigir um número muito menor de operações aritméticas para obter as equações dinâmicas, mas possui a desvantagem de não obtê-las na forma fechada.

PAUL (1981), ASADA; SLOTTINE (1986), FU et. al. (1987), e SPONG; VIDYASAGAR (1989) apresentam estes dois métodos. FU et. al. (1987, cap.3) apresenta também uma variação do método de Lagrange-Euler, que são as equações generalizadas de d'Alembert. Assim como no método de Lagrange-Euler as equações dinâmicas são obtidas na forma fechada, mas é necessário um número menor de operações aritméticas para obtê-las.

O número de operações requeridas por estes métodos se torna crítico quando se utiliza um deles para calcular os termos das equações dinâmicas em tempo real, com a finalidade de serem usadas na lei de controle¹. Neste caso, a vantagem do método de Newton-Euler seria evidente (embora houvesse o inconveniente das equações não serem obtidas na forma fechada).

No entanto, as equações também podem ser introduzidas na lei de controle já na sua forma analítica. Neste caso, um destes métodos seria utilizado apenas uma vez para obter as equações; o número de operações do método utilizado já não seria crítico para o desempenho do sistema, e o método de Lagrange-Euler ofereceria a vantagem de apresentar as equações na forma fechada.

2.2.2. Lagrange-Euler

As equações dinâmicas de um manipulador podem ser obtidas através da aplicação direta da formulação de Lagrange-Euler a sistemas não-conservativos. A aplicação direta da formulação dinâmica lagrangiana, juntamente com a representação de Denavit-Hartenberg, resulta em um algoritmo conveniente para a descrição das equações de movimento de um manipulador. O algoritmo é baseado em operações matriciais e facilita a análise e a implementação computacional.

O algoritmo é dado pelos seguintes passos (FU et. al., 1987, cap.3):

- a) Obtenção das matrizes de transformação homogêneas pelo método de Denavit-Hartenberg.
- b) Obtenção das matrizes de inércia (ou pseudo-inércia) J_i de cada segmento em relação aos eixos nele alocados:

¹ Há leis de controle que utilizam explicitamente os termos das equações dinâmicas.

$$\mathbf{J}_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int y_i z_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (2.13)$$

- c) Obtenção das matrizes \mathbf{Q}_i , $i = 1, \dots, n$. Estas matrizes são definidas de acordo com a regra

$$\mathbf{Q}_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ se a junta } i \text{ for rotacional} \quad (2.14)$$

ou

$$\mathbf{Q}_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ se a junta } i \text{ for prismática}$$

- d) Obtenção das matrizes \mathbf{U}_{ij} e \mathbf{U}_{ijk} ($i, j, k = 1, 2, \dots, n$), que são definidas como

$$\mathbf{U}_{ij} = \begin{cases} {}^0 \mathbf{A}_{j-1} \mathbf{Q}_j^{j-1} \mathbf{A}_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases} \quad (2.15)$$

$$\mathbf{U}_{ijk} = \begin{cases} {}^0 \mathbf{A}_{j-1} \mathbf{Q}_j^{j-1} \mathbf{A}_{k-1} \mathbf{Q}_k^{k-1} \mathbf{A}_i & i \geq k \geq j \\ {}^0 \mathbf{A}_{k-1} \mathbf{Q}_k^{k-1} \mathbf{A}_{j-1} \mathbf{Q}_j^{j-1} \mathbf{A}_i & i \geq j \geq k \\ 0 & i < j \text{ ou } i < k \end{cases} \quad (2.16)$$

- e) Obtenção dos elementos das matrizes que compõem as equações dinâmicas.

As equações dinâmicas do manipulador podem ser escritas da seguinte forma:

$$\boldsymbol{\tau}(t) = \mathbf{H}(\mathbf{q}(t))\ddot{\mathbf{q}}(t) + \mathbf{C}(\mathbf{q}(t), \dot{\mathbf{q}}(t))\dot{\mathbf{q}}(t) + \mathbf{g}(\mathbf{q}(t)) \quad (2.17)$$

onde:

$\boldsymbol{\tau}(t)$ = um vetor $n \times 1$ do torque generalizado aplicado às juntas $i = 1, 2, \dots, n$; isto é

$$\boldsymbol{\tau}(t) = (\tau_1(t) \quad \tau_2(t) \quad \dots \quad \tau_n(t))^T \quad (2.18)$$

$\mathbf{q}(t)$ = um vetor $n \times 1$ das variáveis das juntas do robô; isto é

$$\mathbf{q}(t) = (q_1(t) \quad q_2(t) \quad \dots \quad q_n(t))^T \quad (2.19)$$

$\mathbf{H}(\mathbf{q})$ = uma matriz $n \times n$ denominada matriz de inércia. Seus elementos são obtidos da seguinte forma:

$$H_{ik} = \sum_{j=\max(i,k)}^n \text{Tr}(\mathbf{U}_{jk} \mathbf{J}_j \mathbf{U}_{ji}^T) \quad i, k = 1, 2, \dots, n \quad (2.20)$$

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ = uma matriz $n \times n$, cujos elementos são calculados conforme as seguintes expressões:

$$C_{ik} = \sum_{j=1}^n c_{ijk} \dot{q}_j \quad (2.21)$$

$$c_{ijk} = \sum_{w=\max(i,j,k)}^n \text{Tr}(\mathbf{U}_{wjk} \mathbf{J}_w \mathbf{U}_{wi}^T) \quad i, j, k = 1, 2, \dots, n \quad (2.22)$$

$\mathbf{g}(\mathbf{q})$ = um vetor $n \times 1$ com as forças externas generalizadas.

$$g_i = \sum_{j=i}^n (-m_j \mathbf{a} \mathbf{U}_{ji}^T \mathbf{r}_j) \quad i = 1, 2, \dots, n \quad (2.23)$$

$$\mathbf{a} = (0 \quad 0 \quad -|g| \quad 0) \quad \text{onde } g = 9.8062 \text{ m/s}^2 \quad (2.24)$$

${}^j \mathbf{r}_j$ é o vetor da posição do centro de gravidade do segmento j em relação ao sistema de coordenadas j (escrito neste mesmo sistema de coordenadas).

2.2.3 Consideração dos atuadores no modelo dinâmico

Os robôs possuem atuadores que exercem torques nas juntas dos manipuladores. Assim, na prática se tem acesso direto às entradas dos atuadores, e não aos torques nas juntas (figura 2.7).

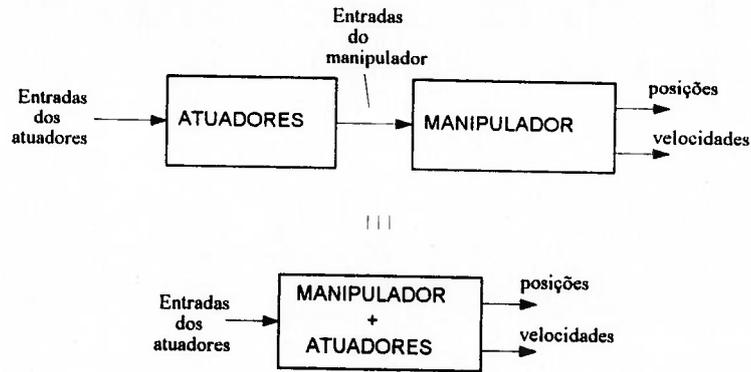


Figura 2.7. Sistema formado pelo conjunto atuadores + manipulador.

Os atuadores podem ser pneumáticos, hidráulicos ou elétricos. A seguir, apresenta-se o caso de particular interesse para este trabalho, onde os atuadores são motores de corrente contínua (CC).

O modelo dinâmico simplificado de um motor de corrente contínua com indutância de armadura desprezível é dado por (MARTÍNEZ, 1991, cap.3):

$$J_m \ddot{\theta}_m + f_m \dot{\theta}_m + \frac{K_a K_b}{R_a} \dot{\theta}_m + \tau_m = \frac{K_a}{R_a} v \quad (2.25)$$

onde

J_m é a inércia do rotor;

f_m é o coeficiente de atrito do rotor com os seus suportes;

θ_m é a posição angular do motor;

K_a é a constante de torque do motor;

K_b é a constante de força contra-eletromotriz;

R_a é a resistência de armadura;

τ_m é o torque aplicado à carga que está ligada ao motor;

v é a tensão de armadura.

Se os motores estiverem ligados às juntas simplesmente por meio de uma redução:

$$\begin{aligned} q_i &= \frac{\theta_{m_i}}{r_i} \quad i = 1 \dots n \\ \tau_i &= r_i \tau_{m_i} \end{aligned} \quad (2.26)$$

onde r_i é a redução da junta i (geralmente $r_i \gg 1$).

No entanto, como o manipulador pode incorporar outros tipos de mecanismos de acionamento (polias, correias etc.), as relações entre os torques e posições angulares dos motores e das juntas podem ser expressas de uma forma mais genérica (SPONG e VIDYASAGAR, 1989, cap.7):

$$\begin{aligned} q_i &= f\left(\frac{\theta_{m_1}}{r_1}, \dots, \frac{\theta_{m_n}}{r_n}\right) \quad i = 1 \dots n \\ \tau_i &= f(r_1 \tau_{m_1}, \dots, r_n \tau_{m_n}) \end{aligned} \quad (2.27)$$

Assim, o modelo dinâmico do manipulador e atuadores pode ser obtido com as equações (2.17), (2.25) e (2.27).

Se for considerado um robô onde as relações dadas por (2.26) são válidas, o modelo dinâmico completo do manipulador e seus atuadores pode ser escrito como (MARTÍNEZ, 1991, cap.3):

$$(\mathbf{RH}(\mathbf{q}) + \mathbf{J})\ddot{\mathbf{q}} + \mathbf{RC}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{Rg}(\mathbf{q}) + \mathbf{B}\dot{\mathbf{q}} = \mathbf{Kv} \quad (2.28)$$

onde

$$\begin{aligned}
\mathbf{J} &= \text{diag}\{J_{m_i}\} \\
\mathbf{B} &= \text{diag}\left\{f_{m_i} + \left(\frac{K_{a_i} K_{b_i}}{R_{a_i}}\right)\right\} \\
\mathbf{R} &= \text{diag}\left\{\frac{1}{r_i^2}\right\} \\
\mathbf{K} &= \text{diag}\left\{\left(\frac{K_{a_i}}{R_{a_i}}\right) \frac{1}{r_i}\right\}
\end{aligned} \tag{2.29}$$

Estas equações podem ser reescritas, obtendo

$$\mathbf{H}'(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}'(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}'(\mathbf{q}) = \mathbf{v} \tag{2.30}$$

onde

$$\begin{aligned}
\mathbf{H}'(\mathbf{q}) &= \mathbf{K}^{-1}(\mathbf{R}\mathbf{H}(\mathbf{q}) + \mathbf{J}) \\
\mathbf{C}'(\mathbf{q}) &= \mathbf{K}^{-1}(\mathbf{R}\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{B}) \\
\mathbf{g}'(\mathbf{q}) &= \mathbf{K}^{-1}\mathbf{R}\mathbf{g}(\mathbf{q})
\end{aligned} \tag{2.31}$$

As equações (2.30) possuem uma estrutura semelhante às equações (2.17)². Por isso, ao apresentar as técnicas de controle, continuaremos utilizando as equações (2.17), o que é usual também na literatura.

2.3. Planejamento de trajetória

O planejamento de trajetória é responsável por gerar as referências que o manipulador deve seguir para realizar o movimento desejado. Um movimento de um manipulador geralmente é especificado pelas posições inicial e final desejadas; as velocidades inicial e final desejadas geralmente são nulas. Se as posições forem dadas em relação às coordenadas das juntas, serão obtidas quatro condições de contorno para cada junta:

² MARTÍNEZ (1991, cap.3) mostra também que o modelo continua mantendo a propriedade de poder ser expresso linearmente em termos de um vetor dependente dos parâmetros dinâmicos. Esta propriedade é importante para a técnica de controle adaptativo.

$$\begin{aligned}
 q_i(0) &= q_{0_i} \\
 q_i(t_f) &= q_{f_i} \\
 \dot{q}_i &= 0 \\
 \dot{q}_i(t_f) &= 0 \\
 \text{para } i &= 1, \dots, n
 \end{aligned}
 \tag{2.32}$$

onde $q_i(t)$ é a posição da junta i no instante t , e t_f é o instante final da trajetória.

Há infinitas trajetórias que satisfazem estas condições de contorno, sendo que elas podem ser especificadas em coordenadas de junta ou em coordenadas cartesianas (FU et. al., 1987, cap.4).

2.3.1. Trajetórias em coordenadas de junta

Gerar as trajetórias em coordenadas de juntas é mais simples que gerá-las no espaço cartesiano, pois não é necessário calcular a cinemática inversa para cada ponto intermediário do movimento. No entanto, esta opção é adequada quando não é muito importante o caminho que o punho do manipulador segue no espaço para chegar ao ponto final; caso contrário (se houver obstáculos no ambiente, por exemplo) o problema torna-se um pouco mais complicado.

Em geral, o algoritmo básico para gerar os pontos da trajetória de junta apresenta a estrutura (Fu et. al., 1987, cap.4) da figura 2.8

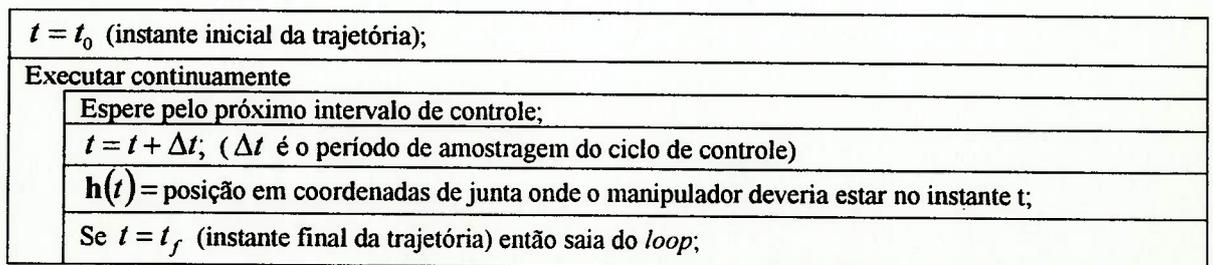


Figura 2.8. Diagrama N-S do algoritmo que gera as trajetórias em coordenadas de junta.

O procedimento para obter $h(t)$ é gerar para cada junta uma curva que interpole as condições de contorno dadas. Um exemplo de uma destas curvas está na figura 2.9.

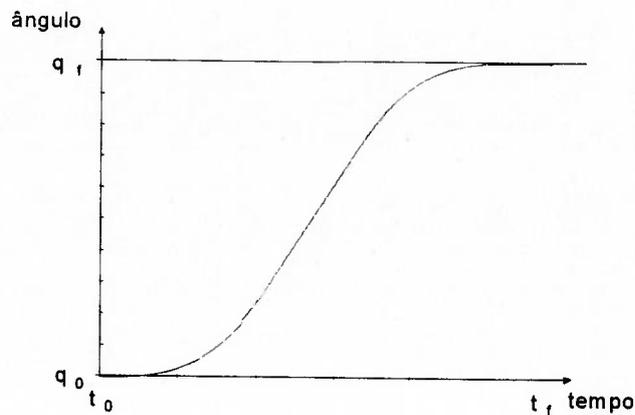


Figure 2.9. Exemplo de uma curva que interpola as posições inicial e final de uma junta.

Um método freqüente é utilizar polinômios para gerar diferentes trechos da trajetória. FU et. al. (1987, cap.4) descreve três destes métodos:

- **4-3-4:** Cada junta tem sua trajetória dividida em três segmentos: o primeiro segmento é gerado por um polinômio de quarto grau; o segundo segmento é gerado por um de terceiro grau; e o terceiro e último, por um do quarto grau.
- **3-5-3:** Semelhante ao método 4-3-4, mas utilizando polinômios de graus diferentes: um polinômio de terceiro grau para o primeiro segmento; um de quinto grau para o segundo segmento; e um de terceiro grau para o último segmento.
- **5-cúbica:** A trajetória é dividida em cinco segmentos, sendo que cada um deles é gerado por um polinômio de terceiro grau.

2.3.2. Trajetórias em coordenadas cartesianas

Para o controle em coordenadas cartesianas, o algoritmo apresenta a estrutura da figura

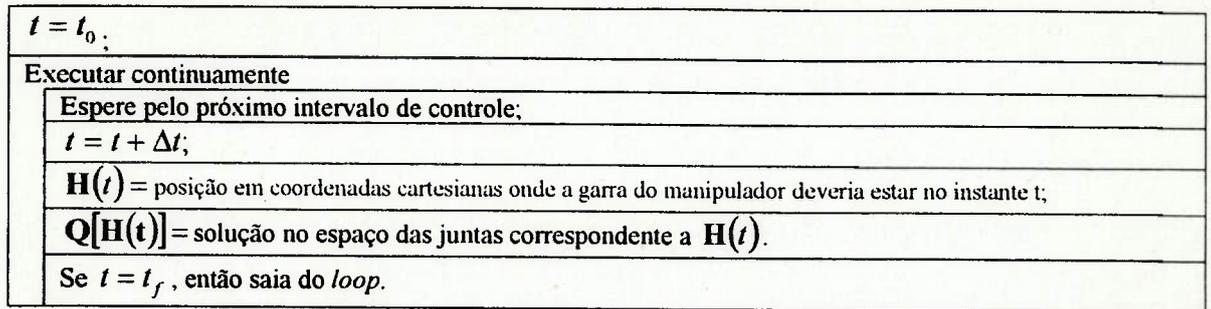


Figura 2.10. Diagrama N-S do algoritmo que gera as trajetórias em coordenadas cartesianas.

Neste caso, além de calcular a função $\mathbf{H}(t)$, que descreve a trajetória da garra, geralmente é necessário converter as posições cartesianas em suas correspondentes soluções no espaço das juntas ($\mathbf{Q}[\mathbf{H}(t)]$). A necessidade desta conversão deve-se ao fato da maioria dos algoritmos de controle trabalhar com base nas variáveis das juntas.

Esta transformação $\mathbf{Q}[\cdot]$ é a resolução da cinemática inversa, já citada anteriormente.

2.4. Controle de posição

O algoritmo de controle de posição é responsável por calcular as entradas que devem ser aplicadas aos atuadores do manipulador para que este se movimente de acordo com a trajetória desejada.

Para expressar isto matematicamente utiliza-se a equação do modelo matemático do manipulador

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (2.33)$$

onde $\boldsymbol{\tau}$ é o vetor de entrada do sistema.

Seja $\mathbf{q}_d(t)$ o vetor das posições angulares desejadas (ou de referência).

O objetivo de controle de movimento consiste em determinar τ de modo que (MARTÍNEZ, 1991, cap.4)

$$\lim_{t \rightarrow \infty} \tilde{\mathbf{q}}(t) = 0 \quad (2.34)$$

onde $\tilde{\mathbf{q}}(t) = \mathbf{q}_d(t) - \mathbf{q}(t)$.

Se a condição acima for satisfeita, o sistema será estável (MARTÍNEZ, 1991). Diz-se que o problema é de regulação quando se deseja satisfazer (2.34) para $\mathbf{q}_d(t)$ constante no tempo. O problema de *tracking* ocorre quando $\mathbf{q}_d(t)$ é variante no tempo. O problema é de otimização quando se deseja minimizar o valor de alguma função durante o movimento (o tempo que o sistema leva para ir de uma posição a outra, por exemplo).

A função utilizada para calcular o vetor τ de maneira a alcançar este objetivo é chamada lei de controle. Genericamente, ela pode ser expressa como

$$\tau = \tau(t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \mathbf{H}(\mathbf{q}), \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}), \mathbf{g}(\mathbf{q})) \quad (2.35)$$

2.4.1. Classificação dos algoritmos de controle

Os algoritmos de controle de posição de robôs podem ser classificados em alguns tipos básicos:

- controle convencional: este grupo inclui os algoritmos que pertencem ao método clássico do controle automático (controles PD ou PID). Quando o manipulador é tratado como n sistemas de uma entrada e uma saída, o controle é denominado independente por juntas (FU et. al., 1987, cap.5); quando o manipulador é tratado como um sistema de n entradas e n saídas acopladas, são utilizadas versões multivariáveis dos controles PD/PID (MARTÍNEZ, 1991, cap.5). Normalmente

adiciona-se a estes algoritmos uma alimentação direta (*feedforward*) com os termos da dinâmica inversa do manipulador.

- controle adaptativo (MARTÍNEZ, 1991, cap.6): são controladores adequados para a situação em que os parâmetros mecânicos do manipulador (massas e inércias, por exemplo) são desconhecidos ou sofrem variações (no caso do robô manipular diferentes cargas, por exemplo). Geralmente envolvem um algoritmo de identificação para estimar os parâmetros do manipulador.
- controle de estrutura variável (ASADA; SLOTINE, 1986, cap.6): são controladores cuja lei de controle é chaveada sempre que uma determinada região do espaço de estados é atravessada. Os algoritmos baseados nesta técnica possuem características desejáveis, como a rejeição de perturbações e a insensibilidade a variações nos parâmetros do sistema.

Além destes três grupos, há outras técnicas citadas em menor número na literatura:

- controle de força: utilizado quando o manipulador mantém contato com uma superfície e deve controlar a força aplicada a ela (ASADA e SLOTINE, 1986, cap.7).
- controle ótimo: busca minimizar um critério de desempenho ao longo de uma trajetória de referência (FU et. al., 1987, cap. 5);
- controle *resolved-motion*: de um modo bem resumido pode ser definido como um algoritmo que permite que os pontos de referência sejam dados nas coordenadas de trabalho da garra, ou seja, a posição desejada é escrita usando diretamente coordenadas do sistema cartesiano. A principal vantagem de realizar o controle no

espaço cartesiano é que, no controle no espaço de juntas, um pequeno erro em cada junta do manipulador pode causar um erro razoável na posição cartesiana da garra. Realizando o controle no espaço cartesiano, há um relaxamento nos erros de posição das juntas, mas os erros de posição e velocidade da garra são menores. Como exemplo, pode ser citado o *resolved motion rate control* (figura 2.11), que utiliza a matriz jacobiana J para realizar as conversões entre as velocidades expressas nas coordenadas de junta e nas coordenadas cartesianas (WHITNEY, 1969; WHITNEY, 1972).

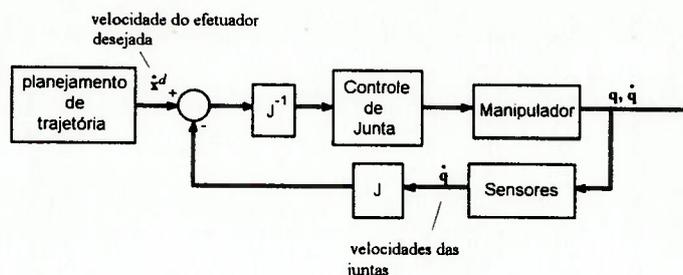


Figura 2.11. Diagrama de blocos do resolved motion rate control.

Os algoritmos de controle de robôs são aplicações da teoria de controle automático (HARASHIMA; DOTE; NAITOH, 1984). A seguir, são explicados alguns termos utilizados em controle automático que também são comuns na classificação de algoritmos de controle de robôs:

- a) controle clássico ou convencional: em controle automático, este termo se refere a técnicas que são aplicáveis somente a sistemas *SISO* (*single-input-single-output*), lineares e invariantes no tempo, sendo que o projeto do controlador é feito no domínio da frequência (HARASHIMA et. al., 1984).

Em controle de robôs, algumas leis de controle multivariáveis (versões PD e PID multivariáveis, por exemplo) também são classificadas como controle

convencional, apesar da lei de controle ser multivariável e a análise da estabilidade ser feita no domínio do tempo (MARTÍNEZ, 1991, cap.5), que são características do controle moderno. No entanto, será seguido aqui o que é usado na literatura sobre controle de robôs. Este modo de classificar é intuitivo, pois os controladores são versões multivariáveis de controladores clássicos.

Os controladores clássicos (do tipo PID independente por junta) têm sido usados em robôs desde a década de 60, mas estudos rigorosos de estabilidade, como os feitos em MARTÍNEZ (1991), foram realizados apenas na década de 80.

- b) controle moderno: este termo é normalmente utilizado em contraste com a teoria de controle clássico ou convencional.

O controle moderno abrange técnicas que são aplicáveis a sistemas multivariáveis - *MIMO (multi-input-multi-output)* - que podem ser lineares ou não-lineares e variantes no tempo. Esta teoria é essencialmente um tratamento no domínio do tempo baseado no conceito de estado. Esta abordagem permite que o projeto seja feito para uma série de entradas (ao invés de uma entrada específica, como a função impulso ou degrau), permitindo também a inclusão das condições iniciais (HARASHIMA et. al., 1984).

As técnicas de controle adaptativo, controle de estrutura variada e controle ótimo são exemplos de aplicações da teoria de controle moderno.

- c) controle robusto: robustez geralmente é entendida como a capacidade de um determinado algoritmo de controle manter alguma característica de desempenho (geralmente, a estabilidade assintótica global) ainda que ocorram variações ou distúrbios na dinâmica do sistema em relação ao modelo que foi usado inicialmente para projetar a lei de controle. Neste sentido, todo controlador

apresenta um certo grau de robustez; no entanto, costuma-se chamar de robustos aqueles algoritmos que apresentam uma boa insensibilidade à variação dos parâmetros do sistema dentro de certos limites pré-especificados no projeto (SPONG; VIDYASAGAR, 1989). Um resumo das principais técnicas de controle robusto aplicadas a manipuladores pode ser encontrado em ABDALLAH et. al. (1991).

- d) controle baseado no modelo: praticamente toda técnica de controle necessita de algum grau de conhecimento do modelo do sistema que será controlado; no entanto, este termo é utilizado na literatura de controle de robôs quando a lei de controle depende explicitamente das matrizes $H(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ e $\mathbf{g}(\mathbf{q})$. Como será visto adiante, diversas técnicas de controle convencional são baseadas no modelo; a maioria das técnicas de controle adaptativo também o são, pois necessitam do conhecimento prévio da estrutura das equações do sistema (os parâmetros numéricos são estimados). Um exemplo de um controlador que não é baseado no modelo (segundo a definição aqui usada) é o controle PD, pois nenhuma matriz do modelo é utilizada na lei de controle.

Alguns trabalhos e livros apresentam um apanhado de diversas técnicas de controle. MARTÍNEZ (1991) apresenta diversas técnicas de controle convencional (multivariável) e adaptativo e verifica, para cada controlador, a estabilidade do sistema usando o método direto de Lyapunov. BERGERMAN (1992) apresenta resultados obtidos em simulações utilizando diversas técnicas de controle (convencional, adaptativo e de estrutura variável), incluindo a maioria dos algoritmos apresentados em MARTÍNEZ (1991).

PAUL (1981, cap.7) apresenta detalhadamente um controle PD independente por junta; com o acréscimo de alguns termos do modelo dinâmico para diminuir o acoplamento.

FU et. al. (1987, cap.5) também apresenta um projeto detalhado de um controle PD independente por junta, e utiliza-o como base para implementar um controle por torque computado, que praticamente utiliza todos os termos do modelo do manipulador. Outras técnicas de controle são apresentadas rapidamente: controle ótimo, de estrutura variada, *resolved motion*, adaptativo. Particularmente no caso do controle adaptativo, algumas das técnicas apresentadas pelo livro já não são consideradas adequadas, pois os algoritmos apresentados foram originalmente desenvolvidos para sistemas lineares, e as provas de estabilidade foram feitas somente com modelos simplificados.

ASADA; SLOTINE (1986, cap.6) também apresentam o projeto de um controle PD independente por junta, mas acrescenta a prova de estabilidade utilizando o método direto de Lyapunov. Apresentam também o controle por torque computado, o controle de estrutura variável e o controle de força (cap.7).

2.4.2. Controle convencional

A da tabela 2.1 apresenta as leis de controle de diversos algoritmos de controle convencional.

O primeiro algoritmo a ser considerado é o controle independente por junta. PAUL (1981, cap.7) e FU et. al. (1987, cap.5) apresentam inicialmente, de um modo bem detalhado, o projeto de um controlador PD independente por junta. As leis de controle são dadas por

$$\tau_i = K_p \tilde{q}_i + K_v \dot{\tilde{q}}_i \quad (2.36)$$

Tabela 2.1. Controladores convencionais e suas respectivas leis de controle.

Controlador	Lei de controle
PD	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}}$
Proporcional com realimentação de velocidade	$\tau = K_p \tilde{q} - K_v \dot{q}$
PID	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + K_I \int_0^t \tilde{q}(s) ds$
compensação pré-calculada	$\tau = H(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + g(q_d)$
PD com compensação de gravidade	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q)$
PD com compensação pré-calculada de gravidade	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q_d)$
PD com compensação pré-calculada	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + H(q_d) \ddot{q}_d + C(q_d, \dot{q}_d) \dot{q}_d + g(q_d)$
PD+	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + H(q) \ddot{q}_d + C(q, \dot{q}) \dot{q}_d + g(q)$
PD com compensação	$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + H(q) (\ddot{q}_d + \Lambda \dot{\tilde{q}}) + C(q, \dot{q}) (\dot{q}_d + \Lambda \tilde{q}) + g(q)$ $\Lambda = K_v^{-1} K_p$
torque calculado	$\tau = H(q) [\ddot{q}_d + K_p \tilde{q} + K_v \dot{\tilde{q}}] + C(q, \dot{q}) \dot{q} + g(q)$
Torque calculado +	$\tau = H(q) [\ddot{q}_d + K_p \tilde{q} + K_v \dot{\tilde{q}}] + C(q, \dot{q}) \dot{q} + g(q) - C(q, \dot{q}) v$ $v = -\frac{p}{p+\lambda} \dot{\tilde{q}} - \frac{1}{p+\lambda} (K_p \tilde{q} + K_v \dot{\tilde{q}})$ $p = \frac{d}{dt}, \quad \lambda > 0$

Estes projetos (para a obtenção dos parâmetros K_p e K_v) são feitos no domínio da frequência. É feita a consideração que o problema se resume ao controle de n motores CC com carga, e assim são obtidos os parâmetros K_p e K_v . O controlador obtido desta forma só funcionaria de modo aceitável se as reduções do robô fossem grandes o suficiente para que a dinâmica do manipulador possa ser considerada como um distúrbio. Assim, os autores propõem adições ao controle PD projetado para eliminar a dinâmica do manipulador.

Algoritmos de controle convencional multivariáveis são apresentados em MARTÍNEZ (1991). O controle PD multivariável é dado pela seguinte lei de controle:

$$\tau = \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}} \quad (2.37)$$

onde \mathbf{K}_p e \mathbf{K}_v são matrizes $n \times n$ definidas positivas. O diagrama de blocos deste controlador é mostrado na figura 2.12.

Pode-se observar que o controle PD independente por junta é um caso particular do PD multivariável; ele corresponde ao caso em que as matrizes \mathbf{K}_p e \mathbf{K}_v são diagonais.

Outros algoritmos que não são baseados no modelo do manipulador podem ser vistos nas primeiras linhas da tabela 2.1:

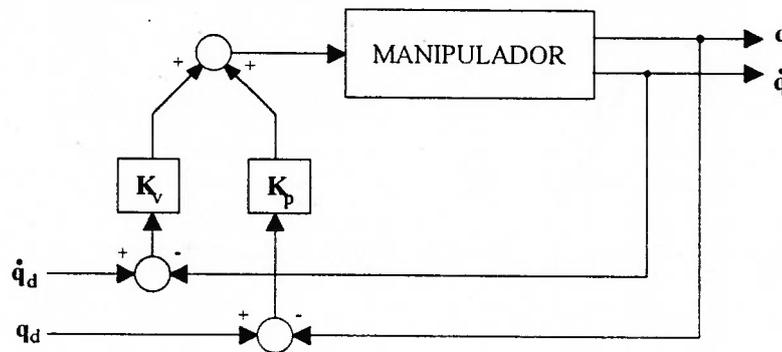


Figura 2.12. Controle PD.

O controle proporcional com realimentação de velocidade é um estágio anterior do controle PD; e é bastante usado no controle da posição angular de motores de corrente contínua. O controle PID é o controle PD adicionado a uma componente integral, que é responsável por eliminar o erro do sistema em estado estacionário.

As leis de controle restantes da tabela 2.1 são técnicas baseadas no modelo do manipulador. O controle com compensação pré-calculada é um controle em malha

aberta, ou seja, sem realimentação; a partir das referências e do modelo do manipulador são calculados os torques/forças que devem ser aplicados em cada junta. Para que este algoritmo funcione bem, é necessário conhecer exatamente as equações dinâmicas do manipulador.

As técnicas de controle seguintes da tabela são controladores PD combinados parcial ou totalmente com o modelo dinâmico do manipulador.

O controle PD com compensação da gravidade (MARTÍNEZ, 1991, cap.5) é o algoritmo de controle proporcional-derivativo adicionado à componente gravitacional do modelo do robô ($\mathbf{g}(\mathbf{q})$). Esta componente $\mathbf{g}(\mathbf{q})$ é responsável pelo erro estático em regime estacionário que surge quando um controle PD puro é utilizado; esta é a motivação para que esta componente seja adicionada à lei de controle. Além do mais, o vetor $\mathbf{g}(\mathbf{q})$ é relativamente fácil de ser calculado, pois depende apenas do cálculo da energia potencial do robô. Se o modelo matemático do manipulador for exato (ou seja, corresponde exatamente à dinâmica do manipulador) o erro estacionário em regime permanente pode ser eliminado através deste método.

O controle PD com compensação pré-calculada de gravidade (MARTÍNEZ, 1991, cap.5) é estruturalmente igual ao controlador anterior, sendo que a única diferença é a utilização de $\mathbf{g}(\mathbf{q}_d)$ em lugar de $\mathbf{g}(\mathbf{q})$ (ou seja, o vetor é calculado para as posições de referência, e não para as posições reais do sistema). Isto permite que o termo de compensação gravitacional seja calculado antes que o movimento seja executado, o que diminui o número de cálculos em tempo real.

O controle PD com pré-compensação (MARTÍNEZ, 1991, cap.5) é uma combinação do controle PD com o controle com pré-compensação. Este algoritmo pode

ser visto como uma generalização do PD com compensação pré-calculada de gravidade, com a diferença que o objetivo agora é eliminar não apenas o erro estacionário, mas também o erro que ocorre durante o movimento devido aos outros termos dinâmicos do manipulador. O cálculo prévio, utilizando as posições de referência, tem o objetivo de diminuir o número de cálculos em tempo real.

O controle PD+ é semelhante ao PD com pré-compensação (MARTÍNEZ, 1991, cap.5), com a diferença que \mathbf{H} , \mathbf{C} e \mathbf{g} são calculados utilizando as posições e velocidades reais do manipulador durante o movimento; os vetores de aceleração e velocidade que multiplicam as matrizes \mathbf{H} e \mathbf{C} , respectivamente, continuam sendo calculados utilizando a referência. Pode-se observar que, quando a referência é constante ($\dot{\mathbf{q}}_d = \mathbf{0}$; $\ddot{\mathbf{q}}_d = \mathbf{0}$), o algoritmo se reduz ao controle PD com compensação de gravidade. Este controlador foi proposto por PADEN; PANJA (1988).

O controle PD com compensação (MARTÍNEZ, 1991, cap.5) é uma variação do PD+. Como no controle PD, as matrizes \mathbf{K}_p e \mathbf{K}_v devem ser definidas positivas; para o controle PD com compensação, elas também devem fazer com que a matriz $\Lambda = \mathbf{K}_v^{-1}\mathbf{K}_p$ seja definida positiva.

O controle por torque calculado (MARTÍNEZ, 1991, cap.5) combina um controlador PD e o modelo do manipulador de maneira que, se o modelo for exato, o comportamento do sistema fica reduzido ao de um sistema do segundo grau.

O controle por torque calculado+ (MARTÍNEZ, 1991, cap.5) é o controle por torque calculado com um termo adicional. Pode-se notar que a única diferença com o controle por torque calculado é o termo adicional $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v}$, sendo que o vetor $\mathbf{v}(t)$ é obtido pela filtragem dos erros de posição e de velocidade. Devido à presença deste

vetor, esta lei de controle é dinâmica, ou seja, não depende apenas dos valores atuais de \mathbf{q} , $\dot{\mathbf{q}}$, \mathbf{q}_d , $\dot{\mathbf{q}}_d$ e $\ddot{\mathbf{q}}_d$, mas também de seus valores passados.

2.4.3. Controle adaptativo

O projeto de um algoritmo de controle convencional na maioria das vezes depende do modelo dinâmico do sistema que se quer controlar. O modelo pode ser necessário por aparecer explicitamente na lei de controle ou para ajustar os parâmetros do controlador. A estrutura (as operações de cálculo que relacionam a entrada com a saída) do modelo matemático de um manipulador pode ser conhecida com uma certa precisão, mas os seus parâmetros (massas e inércias dos segmentos, por exemplo) normalmente não são conhecidos com exatidão. Mesmo que sejam conhecidos com uma precisão aceitável, alguns destes parâmetros podem mudar durante a operação do sistema (por exemplo, se o robô passar a manipular uma carga diferente). As técnicas de controle adaptativo em geral possuem um algoritmo que estima os parâmetros do sistema, e estes, por sua vez, são usados para ajustar a lei de controle.

Historicamente, dentro da teoria de sistemas, o controle adaptativo teve o seu foco inicialmente voltado para sistemas lineares. Apesar das plantas industriais geralmente serem não-lineares, é normal linearizar os sistemas em torno do ponto de operação e realizar o projeto de controle, pois geralmente o objetivo de controle desejado era a regulação.

Muitos dos projetos iniciais de controle adaptativo para manipuladores propostos na literatura realizam suposições ou aproximações, tais como linearização, invariância no tempo ou dinâmica desacoplada, para garantir a convergência no *tracking*. Os trabalhos de KOIVO; GUO (1983) e DUBOWSKY; DesFORGES (1979), por exemplo, procuram aplicar aos manipuladores técnicas utilizadas anteriormente em sistemas

lineares (*self-tuning* e *Model Reference Adaptive Control*, respectivamente). Nestes artigos a estabilidade é provada somente para o modelo simplificado do manipulador, e não para o modelo completo. A partir de 1986, começaram a surgir os primeiros controladores adaptativos globalmente convergentes³, que não fazem estas aproximações. Um tutorial sobre as técnicas de controle globalmente convergentes é apresentado em ORTEGA; SPONG (1989). WHITCOMB, RIZZI; KODITSCHKEK (1993) contém um comparativo entre três técnicas adaptativas e quatro técnicas convencionais.

As técnicas de controle mais recentes (a partir de 1986) baseiam-se em uma propriedade do modelo dinâmico do manipulador, segundo a qual as equações dinâmicas podem ser reescritas na forma (MARTÍNEZ, 1991, cap.3)

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} \quad (2.38)$$

onde $\boldsymbol{\phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ é uma matriz $n \times r$ de funções, e $\boldsymbol{\theta}$ é um vetor de dimensão r dependente dos parâmetros dinâmicos do manipulador (massas, inércias etc.).

Assume-se que a matriz $\boldsymbol{\phi}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ é conhecida, enquanto que os parâmetros do vetor $\boldsymbol{\theta}$ são incertos ou desconhecidos.

Em geral, um controlador adaptativo é formado por duas partes: a lei de adaptação e a lei de controle. A lei de adaptação permite determinar uma estimativa dos parâmetros que compõem o vetor $\boldsymbol{\theta}$, a partir dos valores das entradas e das saídas do sistema ao longo do tempo. O vetor com os valores estimados normalmente é denominado $\hat{\boldsymbol{\theta}}$.

³ Controladores globalmente convergentes são aqueles que garantem, para todas as trajetórias e condições iniciais, o seguimento assintótico da trajetória mantendo limitados todos os seus sinais internos (ORTEGA e SPONG, 1989).

$$\hat{\theta} = \hat{\theta}(t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \hat{\theta}(0)) \quad (2.39)$$

Quando as matrizes \mathbf{H} , \mathbf{C} e \mathbf{g} são calculadas utilizando os parâmetros estimados, costuma-se representá-las por $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$ e $\hat{\mathbf{g}}$, respectivamente.

A lei de controle continua sendo definida como em (2.35), mas agora o modelo do manipulador depende dos parâmetros estimados.

$$\tau = \tau(t, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d, \hat{\theta}) \quad (2.40)$$

Um diagrama de blocos genérico tem então a forma da figura 2.13.

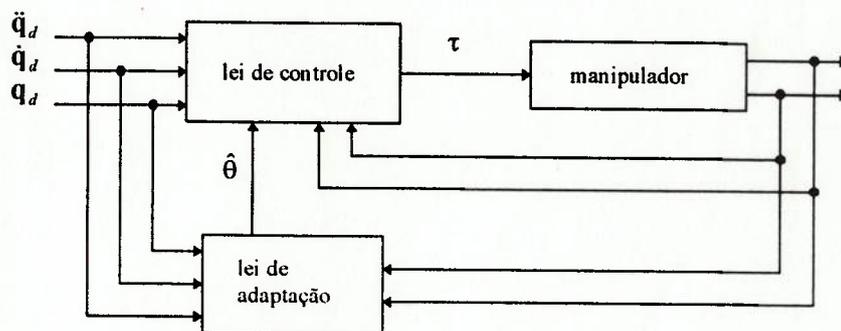


Figura 2.13. Diagrama de blocos genérico de um controle adaptativo.

O controle PD com compensação adaptativo e o controle por torque calculado+ adaptativo (MARTÍNEZ, 1991, cap.6) são versões adaptativas do controle PD com compensação e do controle por torque calculado+, respectivamente.

O controle adaptativo com dinâmica inversa (WHITCOMB; RIZZI; KODITSCHKEK, 1993) é outra variação de controle adaptativo.

Estas técnicas são globalmente assintoticamente estáveis em relação ao erro de posição, mas são somente globalmente estáveis em relação ao erro nos parâmetros; isto quer dizer que os parâmetros não necessariamente convergem para os valores reais do sistema.

Tabela 2.2. Exemplos de controladores adaptativos.

<p>PD com compensação adaptativo ou adaptativo com dinâmica inversa e "criticamente amortecido"</p>	$\tau = \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}} + \phi \hat{\theta}$ $\phi \hat{\theta} = \hat{\mathbf{H}}(\mathbf{q})(\ddot{\mathbf{q}}_d + \Lambda \dot{\tilde{\mathbf{q}}}) + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})(\dot{\mathbf{q}}_d + \Lambda \tilde{\mathbf{q}}) + \hat{\mathbf{g}}(\mathbf{q})$ $\hat{\theta} = \Gamma \int_0^t \phi^T (\dot{\mathbf{q}} + \Lambda \tilde{\mathbf{q}}) dt + \hat{\theta}(0)$ $\Lambda = \mathbf{K}_v^{-1} \mathbf{K}_p$
<p>Torque calculado+ adaptativo</p>	$\tau = \phi \hat{\theta}$ $\phi \hat{\theta} = \hat{\mathbf{H}}(\mathbf{q})[\ddot{\mathbf{q}}_d + \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}}] + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) - \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v}$ $\hat{\theta} = -\Gamma \int_0^t \phi^T \mathbf{v} dt + \hat{\theta}(0)$ $\mathbf{v} = -\frac{p}{p+\lambda} \dot{\tilde{\mathbf{q}}} - \frac{1}{p+\lambda} (\mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}})$ $p = \frac{d}{dt}, \quad \lambda > 0$
<p>adaptativo com dinâmica inversa</p>	$\tau = \mathbf{K}_p \tilde{\mathbf{q}} + \mathbf{K}_v \dot{\tilde{\mathbf{q}}} + \phi \hat{\theta}$ $\phi \hat{\theta} = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_d + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \hat{\mathbf{g}}(\mathbf{q})$ $\hat{\theta} = \Gamma \int_0^t \phi^T (\dot{\mathbf{q}} + \varepsilon(\tilde{\mathbf{q}})\tilde{\mathbf{q}}) dt$ $\varepsilon(\tilde{\mathbf{q}}) = \varepsilon_0 (1 + \ \tilde{\mathbf{q}}\)^{-1}$

2.4.4. Controle de estrutura variável

Conforme o próprio nome diz, a lei de controle obtida com esta técnica possui uma estrutura que varia de acordo com regras pré-especificadas. Para entender os conceitos básicos envolvidos nesta técnica de controle, pode-se recorrer a um exemplo simples dado por YOUNG (1978). Seja um sistema de segunda ordem linear e invariante no tempo:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= ax_2 - bu \\ \mathbf{x}^T &= (x_1, x_2) \\ a, b &> 0 \end{aligned} \tag{2.41}$$

Uma lei de controle de estrutura variável para este sistema é da forma

$$u = \psi x_1 \quad (2.42)$$

onde ψ é uma função descontínua dada por

$$\psi = \begin{cases} \alpha, & \text{se } x_1 s > 0 \\ -\alpha, & \text{se } x_1 s < 0 \end{cases}$$

$$\alpha > 0$$

$$s = cx_1 + x_2, \quad c > 0 \quad (2.43)$$

sendo que $s=0$ define o que será chamado aqui de linha de chaveamento.

O sistema resultante é uma combinação de dois sistemas, cujas equações e planos de fase estão nas figuras 2.14, 2.15 e 2.16.

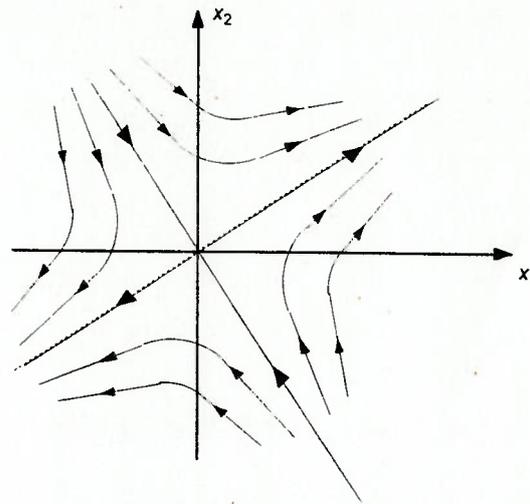


Figura 2.14. Plano de fase para $\psi = \alpha$.

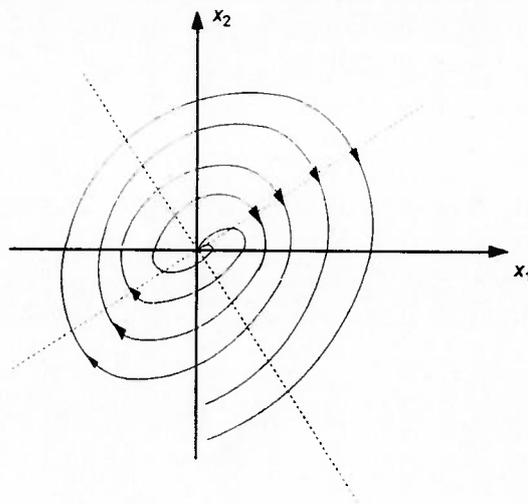


Figura 2.15. Plano de fase para $\psi = -\alpha$.

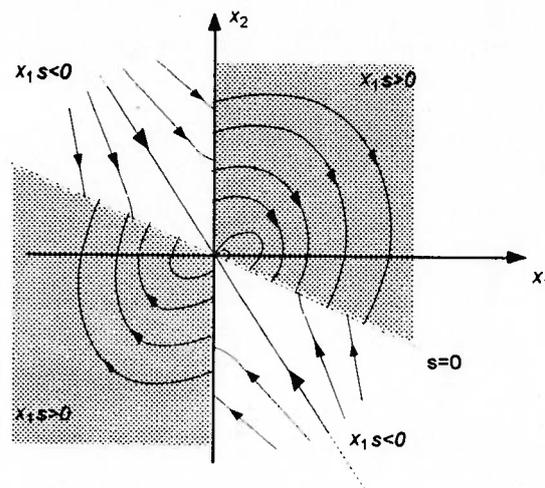


Figura 2.16. Plano de fase para ψ descontínua.

O plano de fase final mostra que todas as trajetórias acabam levando o sistema para a reta $s = 0$; ao longo desta linha ocorre o chaveamento da lei de controle, fazendo com que o sistema se mantenha na reta, seguindo para a origem. Este é o chamado modo de escorregamento (*sliding mode*).

Quando o sistema está no modo de escorregamento, suas trajetórias satisfazem $s = cx_1(t) + x_2(t) = 0$. Neste caso, o comportamento do sistema é governado pela equação

$$\dot{x}_1 = -cx_1 \quad (2.44)$$

Pode-se observar que o comportamento do sistema no modo de escorregamento depende apenas de c , que é um parâmetro de projeto, e independe dos parâmetros do sistema a e b (dentro da suposição que são positivos). Isto caracteriza este controlador como insensível às variações nos parâmetros do sistema.

Há ainda duas questões importantes para serem tratadas com respeito a esta técnica de controle. Quando o modo de escorregamento ocorre, teoricamente o controle de estrutura variável faz o chaveamento a uma frequência infinita. No entanto, na prática o chaveamento ocorre a uma frequência finita, o que ocasiona uma “trepidação” (*chattering*) do sistema em torno da linha de chaveamento. Este sinal de alta frequência pode trazer efeitos indesejáveis (excitando dinâmicas não modeladas, por exemplo), o que levou à idéia de utilizar uma função “menos abrupta” para realizar o chaveamento (ASADA; SLOTINE, 1986, cap.6). Seguindo esta solução, ψ poderia ser adotado como sendo dado por

$$\psi = bu(x_1, s) \quad (2.45)$$

$$u(x_1, s) = \begin{cases} \text{sgn}(x_1, s) & \text{se } |s| \geq \varepsilon \\ \frac{1}{\varepsilon} |s| \text{sgn}(x_1, s) & \text{se } |s| < \varepsilon \end{cases}$$

ou seja, quando $|s| \geq \varepsilon$, ψ é exatamente igual ao que foi estabelecido inicialmente; mas quando $|s| < \varepsilon$, o valor de ψ passa a variar de uma maneira proporcional ao módulo de s . Deste modo, consegue-se eliminar (ou pelo menos diminuir) efeitos indesejáveis devidos à “trepidação”.

Outra questão importante é a condição para que as trajetórias do sistema atinjam a linha de chaveamento. Para garantir isto, a condição $\dot{s}s < 0$ deve ser satisfeita e os valores limites dos parâmetros devem ser conhecidos.

BERGERMAN (1992) apresenta três métodos de controle de estrutura variável aplicados a manipuladores.

Como um exemplo deste tipo de controle, é apresentado a seguir o algoritmo proposto por ASADA; SLOTINE (1986, cap.6).

Seja o modelo disponível (não o real) do manipulador dado pelas equações

$$\tau = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{h}} \quad (2.46)$$

Em geral, supõe-se que a incerteza do modelo está nos seus parâmetros, e não na sua estrutura.

A lei de controle é dada por

$$\tau = \hat{\mathbf{H}}\mathbf{u} + \hat{\mathbf{h}} \quad (2.47)$$

onde as componentes do vetor \mathbf{u} são definidas como

$$\begin{aligned} u_i &= G_i(\mathbf{q})[\hat{u}_i - \bar{k}_i(\mathbf{q}, \dot{\mathbf{q}})sat(s_i/\phi_i)] \\ \hat{u}_i &= \ddot{q}_{d_i} + 2\lambda\tilde{q}_i + \lambda^2\tilde{q}_i \\ s_i &= -\tilde{q}_i - 2\lambda\tilde{q}_i - \lambda^2 \int^T \tilde{q}_i(T)dT \\ sat(s_i/\phi_i) &= \begin{cases} s_i/\phi_i & \text{se } |s_i| \leq |\phi_i| \\ \text{sgn}(s_i/\phi_i) & \text{em outros casos} \end{cases} \\ \tilde{q}_i &= q_{d_i} - q_i \end{aligned} \quad (2.48)$$

λ é a largura de banda desejada, e deve ser escolhida pequena o suficiente para não excitar modos estruturais não modelados ou interferir com atrasos de tempo que foram desconsiderados na modelagem.

Para obter $G_i(\mathbf{q})$, $\bar{k}_i(\mathbf{q}, \dot{\mathbf{q}})$ e ϕ_i , deve-se fazer uma avaliação prática das incertezas nos parâmetros. Seja

$$\begin{aligned} \mathbf{H}^{-1} &= [\mathbf{L}_1 \dots \mathbf{L}_n] \\ \hat{\mathbf{H}} &= [\hat{\mathbf{H}}_1 \dots \hat{\mathbf{H}}_n] \\ \hat{\mathbf{H}} - \mathbf{H} &= [\Delta\mathbf{H}_1 \dots \Delta\mathbf{H}_n] \\ \Delta\mathbf{h} &= \hat{\mathbf{h}} - \mathbf{h} \end{aligned} \quad (2.49)$$

onde \mathbf{H} e \mathbf{h} compõem as equações reais do sistema:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h} \quad (2.50)$$

Seja

$$\mathbf{L}_i^T \cdot \hat{\mathbf{H}}_i > 0 \quad i = 1, \dots, n \quad (2.51)$$

Os valores exatos dos parâmetros dinâmicos do manipulador não são conhecidos, mas é importante saber os limites dentro dos quais estes parâmetros podem variar. Se a condição acima não puder ser garantida para toda a faixa de incerteza dos parâmetros, uma solução é adotar $\hat{\mathbf{H}}$ como uma matriz diagonal definida positiva (composta, por exemplo, dos elementos diagonais da matriz $\hat{\mathbf{H}}$ original).

O conhecimento dos limites em que se encontram os parâmetros permite a obtenção de escalares $\beta_i^{\min} = \beta_i^{\min}(\mathbf{q})$ e $\beta_i^{\max} = \beta_i^{\max}(\mathbf{q})$ que satisfaçam

$$0 \leq \beta_i^{\min} \leq \mathbf{L}_i^T \cdot \hat{\mathbf{H}}_i \leq \beta_i^{\max} \quad i = 1, \dots, n \quad (2.52)$$

Assim, define-se $G_i = G_i(\mathbf{q})$

$$G_i = (\beta_i^{\min} \beta_i^{\max})^{-1/2} \quad (2.53)$$

A margem de ganho é definida como

$$\beta_i = (\beta_i^{\max} / \beta_i^{\min})^{-1/2} \quad (2.54)$$

e $k_i(\mathbf{q}, \dot{\mathbf{q}})$ deve satisfazer a seguinte condição

$$k_i(\mathbf{q}, \dot{\mathbf{q}}) \geq \beta_i \left[(1 - \beta_i^{-1}) |\hat{u}_i| + (\mathbf{L}_i^+)^T \Delta \mathbf{h}^+ + \sum_{j \neq i} G_j |\ddot{\mathbf{q}}_d| \cdot |\mathbf{L}_i^T \cdot \hat{\mathbf{H}}_j| + \eta_i \right] \quad (2.55)$$

η é uma constante positiva, inversamente proporcional ao tempo que o sistema leva para atingir a superfície de escorregamento.

$\Delta \mathbf{h}^+$ e \mathbf{L}_i^+ são os vetores com os valores absolutos dos componentes de $\Delta \mathbf{h}$ e \mathbf{L}_i .

O valor de ϕ_i é obtido pelas seguintes equações diferenciais

$$\begin{aligned} k_i(\mathbf{q}_d, \dot{\mathbf{q}}_d) &\geq \frac{\lambda \phi_i}{\beta_i(\mathbf{q}_d)} \Rightarrow \dot{\phi}_i + \lambda \phi_i = \beta_i(\mathbf{q}_d) k_i(\mathbf{q}_d, \dot{\mathbf{q}}_d) \\ k_i(\mathbf{q}_d, \dot{\mathbf{q}}_d) &\leq \frac{\lambda \phi_i}{\beta_i(\mathbf{q}_d)} \Rightarrow \dot{\phi}_i + \frac{\lambda \phi_i}{\beta_i^2(\mathbf{q}_d)} = \frac{k_i(\mathbf{q}_d, \dot{\mathbf{q}}_d)}{\beta_i(\mathbf{q}_d)} \end{aligned} \quad (2.56)$$

Assumindo as condições iniciais como

$$\phi(0) = \beta_i(\mathbf{q}_d(0)) k_i(\mathbf{q}_d(0), \dot{\mathbf{q}}_d(0)) / \lambda \quad (2.57)$$

Apesar das vantagens do controle de estrutura variável que podem ser percebidas numa análise teórica, na prática a estabilidade assintótica do erro geralmente é sacrificada para evitar a “trepidação” (ABDALLAH et. al., 1991).

2.5. Critérios para avaliação do desempenho de um manipulador

Para avaliar o desempenho de um manipulador com um determinado algoritmo de controle é comum o uso de simulações; outros trabalhos apresentam os resultados obtidos em uma implementação real. Os resultados (das simulações ou das implementações práticas) geralmente são obtidos aplicando as seguintes referências nas entradas do manipulador:

- a) degrau unitário: este tipo de entrada geralmente é utilizado em simulações. É particularmente útil para ajudar no ajuste dos parâmetros de controladores convencionais (PD, PID). Bergerman (1991, cap.4) utiliza este tipo de entrada em simulações e apresenta os resultados comparativos entre vários algoritmos na forma de gráficos e de tabelas. Estas tabelas mostram os valores de sobressinal (M_p), tempo de subida (t_r) e erro de regime (e_{ss}) obtidos com cada algoritmo. Esta experiência permite ter idéia da estabilidade, seguimento e regulação do sistema para os diferentes casos. É útil para apresentar resultados comparativos entre diferentes algoritmos, mas não para atribuir uma nota geral de desempenho, já que o manipulador é um sistema não-linear, e por isso degraus de amplitudes diferentes resultariam em resultados diferentes. Haveria complicações em utilizar este tipo de entrada em uma implementação real devido aos esforços que este tipo de entrada poderia causar no manipulador.

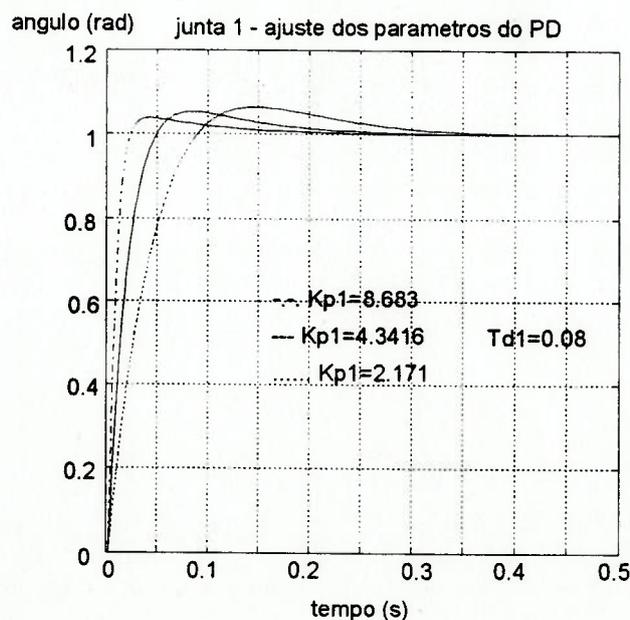


Figura 2.14. Exemplo de apresentação de resultados para uma entrada degrau.

- b) trajetórias de referência: estas trajetórias podem ser no espaço de juntas (3-5-3, 4-3-4) ou no espaço cartesiano (retas, círculos, espirais), e estão mais próximas das condições normais de uso de um manipulador. Em geral, a apresentação de resultados é feita através de gráficos mostrando o erro de posição de cada junta ao longo da trajetória. Este é o caso dos resultados apresentados em FU et. al. (1986, cap.5), e em BERGERMAN (1991, cap.4), por exemplo. Este último também acrescentou tabelas com a medida do erro máximo (e_m) de cada junta ao longo das trajetórias. É comum também encontrar trabalhos que apresentam os gráficos da trajetória descrita pelo manipulador em comparação com as trajetórias de referência.

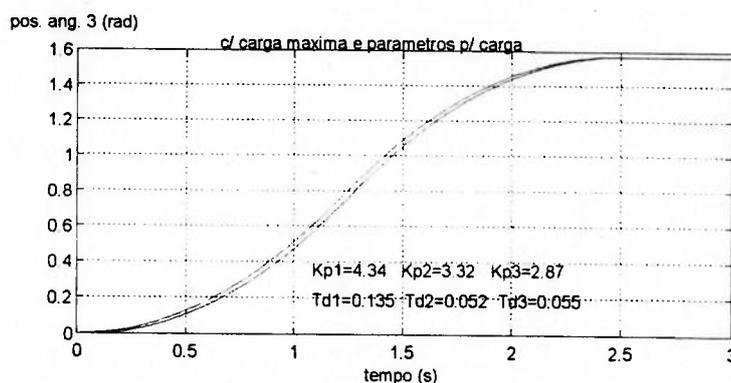


Figura 2.15. Exemplo de apresentação de resultados utilizando trajetórias de referência.

Estas são as formas geralmente adotadas para avaliar e apresentar resultados do desempenho de um manipulador com um determinado algoritmo. WHITCOMB; RIZZI; KODITSCHKEK (1993) propõem em seu trabalho uma medida mais objetiva do desempenho de seguimento de uma trajetória. Para isto, emprega-se a norma L^2 do erro ao longo da trajetória como uma medida numérica objetiva de performance de seguimento, sendo que a norma L^2 é dada por $L^2[e(t)] = \left((1/t) \int_{t_0}^t \|e(t)\|^2 dt \right)^{1/2}$ onde $e(t)$ é

um escalar ou um vetor do erro de seguimento. Para confirmar a confiabilidade dos resultados, cada teste é executado dez vezes (figuras 2.16 e 2.17).

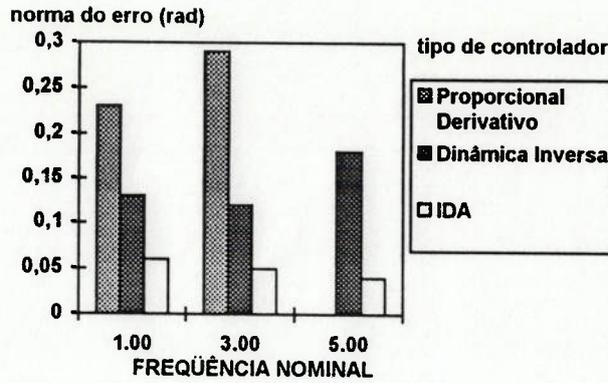


Figura 2.16. Exemplo de medida de desempenho utilizando a norma L^2 do erro de posição versus a frequência nominal da trajetória de referência (WHITCOMB; RIZZI; KODITSCHKEK, 1993).

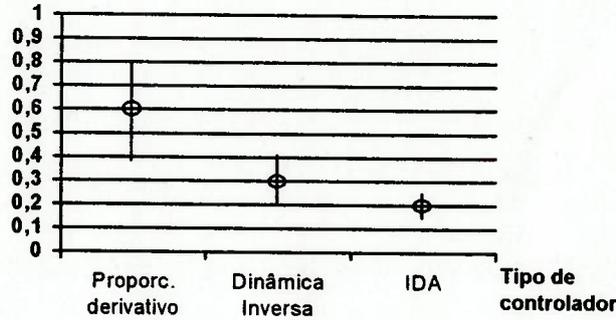


Figura 2.17. Exemplo de medida de desempenho utilizando a média e o desvio padrão da norma L^2 do erro de posição (10 repetições para cada controlador) (WHITCOMB; RIZZI; KODITSCHKEK, 1993).

Capítulo 3

Processadores digitais de sinais em aplicações de controle em tempo real

Após as etapas de modelagem do manipulador e projeto dos algoritmos de trajetória e de controle de posição, a implementação efetiva do sistema de controle envolve ainda os seguintes itens:

- o projeto de um circuito controlador, que pode utilizar um ou mais processadores.
- o projeto do programa que é executado no controlador, permitindo o controle do sistema em tempo real.

A seguir, são apresentados os tópicos relevantes para a implementação de um sistema de controle em tempo real e a utilização de Processadores Digitais de Sinais em aplicações de controle.

3.1. Sistemas em tempo real

Aplicações em robótica são essencialmente uma forma de controle de processos (SAVITZKY, 1985). Numa implementação real, os algoritmos de planejamento de trajetória e de controle de posição devem ser executados dentro de certos limites de tempo que são impostos pelo próprio processo que se deseja controlar (no caso, o manipulador). O sistema de controle de um robô deve, portanto, ser capaz de realizar o controle do sistema em tempo real.

Há algumas definições para a terminologia sistemas em tempo real. A definição padrão é:

“Um sistema em tempo real é aquele em que a exatidão de um sistema computacional não depende apenas da exatidão lógica da computação, mas também do tempo em que o resultado é produzido. Se os limites de tempo do sistema não são satisfeitos, diz-se que ocorreu uma falha no sistema.” (LINIMON, 1994)

Este termo é utilizado muitas vezes quando se quer dizer “rápido”, principalmente em controle de processos. No entanto, tempo real não é necessariamente sinônimo de “rápido”, mas significa que o tempo que o sistema leva para responder é suficiente para que o processo em questão seja controlado.

3.1.1. Hardware usado em aplicações em tempo real

Pouca coisa distingue o *hardware* usado em aplicações em tempo real de um microcomputador comum. Um *hardware* usado em aplicações em tempo real tem uma CPU, memória, periféricos, e assim por diante. Todavia, há razões para escolher um determinado tipo de CPU, periférico, e outras partes constituintes do sistema; esta escolha é feita de acordo com o uso a que o sistema será submetido, os custos do projeto e da implementação etc.

A implementação em tempo real de estratégias de controle cujo algoritmo seja complexo pode requerer uma grande capacidade computacional. Da mesma forma se altas taxas de amostragem forem necessárias.

3.1.2. Software em tempo real

Um dos modos de classificar sistemas de tempo real é de acordo com a estrutura de seu *software* (SAVITZKY, 1985). A seguir, estão descritos alguns termos utilizados para descrever as estruturas de *software*.

Ciclo de “escolha” (*Polling loop*). O programa examina cada uma de suas entradas de maneira a determinar se um evento ocorreu; em caso positivo, a resposta requerida por este evento deve ser apresentada antes de prosseguir para a próxima entrada.

Sistemas direcionados a eventos. Um sistema que responde diretamente a eventos externos. Esta designação se aplica a três tipos principais de sistemas:

- a) *Foreground/Background*. Um ciclo de “escolha” (*polling loop*) é executado em *background*, enquanto eventos em *foreground* causam interrupções no processador. Quando ocorre uma interrupção, o controle é temporariamente transferido a uma rotina que trata deste evento, e então retorna ao processo em *background*.

Em um caso limite, o processo em *background* é simplesmente um *loop* que não faz nada, e todo o processamento de eventos é feito nas rotinas de tratamento de interrupção. O mais freqüente é que as interrupções tratem daqueles eventos que requerem resposta rápida, movendo dados entre dispositivos de entrada/saída e *buffers* na memória. Neste caso, o processo em *background* é então habilitado a prosseguir quando um *buffer* for enchido ou esvaziado.

- b) *Multitarefa (Multitasking)*. Quando vários processos mais ou menos independentes estão ocorrendo ao mesmo tempo, é conveniente manusear cada um com um programa independente. Quando todos estes programas estão sendo executados no mesmo processador, o resultado é chamado multitarefa (*multitasking*).
- c) *Multiprocessador*. Quando há muito trabalho para apenas um processador fazer, adiciona-se outro(s) processador(es), o que constitui um sistema

multiprocessador. Como os microprocessadores estão cada vez menores e com custo menor, este tipo de sistema tem se tornado mais comum.

3.2. Processador Digital de Sinais

O processamento digital de sinais é um campo de estudo que se ocupa com o processamento de informação representada na forma digital (ROBERTS; CLIFFORD, 1987).

No processamento analógico de sinais, o sinal é processado continuamente no tempo com níveis de amplitude contínuos. No processamento digital de sinais, o sinal analógico é amostrado em intervalos de tempo discretos. Sua amplitude é convertida em valores digitais com um conversor A/D. A representação digital do sinal é então processada no domínio digital, como uma seqüência de números. Este processamento pode ser usado para analisar ou sintetizar sinais (FFT's, síntese de voz), para modificar a resposta em frequência ou a amplitude de sinais (filtros, controladores), ou para estimação de parâmetros de sistemas (filtros de Kalman, filtros e controladores adaptativos). Após o processamento, os dados são convertidos de volta à forma analógica com o uso de um conversor D/A.

O processamento digital de sinais já existe como disciplina a algum tempo. As primeiras implementações em computadores grandes não eram muito práticas, mas a disponibilidade de processadores dedicados com custo cada vez menor nos últimos anos está revolucionando o campo. Estes processadores são chamados de processadores digitais de sinais, representados pela sigla DSP (*Digital Signal Processor*).

O primeiro DSP foi lançado pela Texas Instruments em 1982, com a identificação TMS32010. Desde então, os DSP's têm se tornado ferramentas comuns no processamento de voz, filtragem, telecomunicações, e aplicações de análise de sinais.

Eles estão sendo usados em periféricos de computadores tais como *plotters* e acionadores de discos flexíveis, e em sistemas automotivos de controle, como suspensões ativas e freios ABS. Em controle industrial, eles estão sendo usados em robôs, máquinas ferramentas e controle de motores.

A operação mais comumente utilizada em processamento digital de sinais é uma função “multiplica-e-acumula”. Os DSP's implementam esta função em hardware, tal que ela leva apenas um ciclo de máquina. Em um microprocessador de 32 bits, como o 68020, esta função leva até 36 ciclos de máquina. A função “multiplica-acumula” é tão comum em processamento digital de sinais que é freqüente a declaração que ela é a propriedade que define se um processador é um DSP. Em geral, pode ser dito que a capacidade de realizar a operação “multiplica-acumula” em alta velocidade é a distinção principal entre DSP's e outros processadores.

Assim, a arquitetura do DSP torna-o capaz de realizar algumas operações aritméticas rapidamente. Esta característica faz do DSP uma opção adequada para aplicações de controle (AHMED; MESHKAT, 1988), onde os cálculos têm de ser realizados em um intervalo de tempo limitado. Além do mais, alguns algoritmos de controle em sua forma discreta são muito similares a filtros digitais IIR (*Infinite Impulse Response*). As figuras 3.1 e 3.2 mostram como o algoritmo PID (*Proportional-Integral-Derivative*), em sua forma discreta, é muito similar a um filtro IIR, que é normalmente usado em aplicações de processamento digital de sinais.

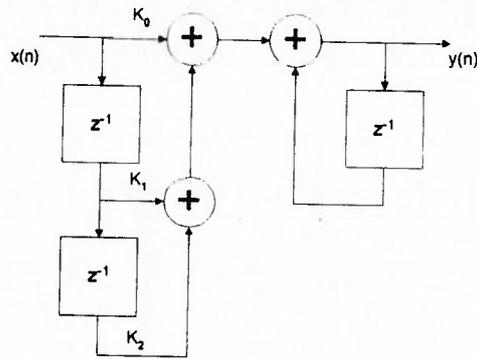


Figura 3.1. Diagrama de fluxo de sinal de um algoritmo PID na forma discreta.

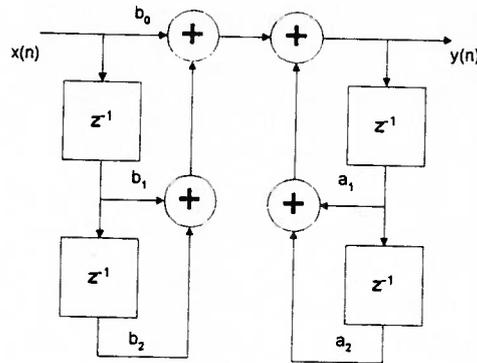


Figura 3.2. Exemplo de um filtro IIR.

SLIVINSKY; BORNINSKI (1988) apresenta um texto didático sobre a implementação de algoritmos de controle nos DSP's da Texas.

Como exemplo da utilização de DSP's em aplicações de controle, pode-se citar HEMERLY (1993), que implementa um sistema utilizando dois DSP's TMS320C30. Este sistema é usado para executar técnicas de controle adaptativas e H^∞ para foguete e para controle de temperatura, e os tempos de execução são comparados com os obtidos em um PC/AT 286.

Um exemplo de aplicação de DSP's na área de controle de robôs pode ser encontrado em DOTE; SHINOJIMA; YOSHIMURA (1987), que implementa um

controle de estrutura variada para um manipulador empregando dois sistemas: um com um microprocessador Intel 8086 e outro com um DSP Texas TMS320C20.

Capítulo 4

Proposta de um sistema de controle para o ensino de controle de robôs

Um sistema de controle de robôs envolve os sub-sistemas responsáveis pelo planejamento de trajetória e controle de posição (figura 4.1)

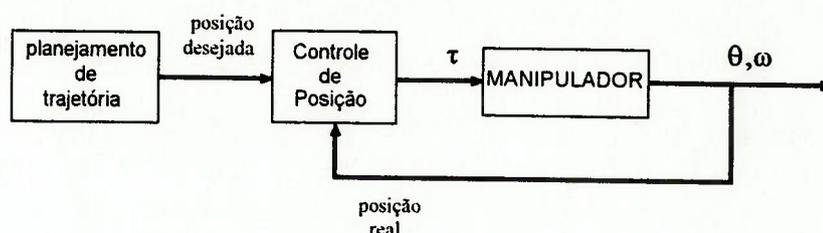


Figura 4.1. Diagrama de blocos de um robô.

Há diversos algoritmos de planejamento de trajetória e controle de posição, sendo que a implementação de alguns deles requer um grande número de operações matemáticas dentro de um período limitado de amostragem.

A proposta deste trabalho foi elaborada tendo em vista os seguintes fatores:

- O interesse em permitir a realização de experiências práticas com diversos algoritmos de controle de robôs.
- A dificuldade em realizar estes experimentos com sistemas didáticos comerciais devido ao fato da maior parte deles possuir uma arquitetura fechada (que não permite modificar o algoritmo de controle do robô).

- A necessidade de um sistema de controle capaz de executar em tempo real várias operações aritméticas e trigonométricas para tornar possível a implementação experimental de alguns dos algoritmos de controle existentes na literatura.

Desta forma, este trabalho propõe um sistema de controle com a capacidade e flexibilidade necessárias para que o estudante possa realizar a implementação do algoritmo de seu interesse e verificar o seu desempenho. Na implementação deste sistema, o uso de um DSP é proposto pelo fato deste processador apresentar características adequadas para aplicações de controle em tempo real mostrando-se veloz na execução de determinados tipos de operação (a operação “multiplica-acumula”, por exemplo), conforme já foi apresentado anteriormente.

Para que este sistema possa ser utilizado no ensino de controle de robôs, ele deve ter a flexibilidade necessária para que os algoritmos responsáveis pelo planejamento da trajetória e controle de posição possam ser trocados com facilidade pelo estudante, sem que ele precise ter conhecimento de todos os detalhes do sistema. Outra característica interessante é permitir a modificação de parâmetros de controle de maneira interativa, de modo que o estudante possa ver a influência de determinados parâmetros do controlador no comportamento do robô.

Capítulo 5

Desenvolvimento do trabalho

5.1. Desenvolvimento do hardware

A solução proposta (figura 5.1) utiliza uma arquitetura com dois processadores: um microcontrolador Intel 8751 e um DSP TMS320C25. Esta arquitetura com dois processadores permite a divisão das funções necessárias ao funcionamento do sistema de controle. Desta forma, o DSP fica responsável apenas pelas tarefas que estão diretamente relacionadas com o movimento do robô, enquanto o microcontrolador cuida de outras tarefas periféricas (comunicação com um microcomputador, por exemplo).

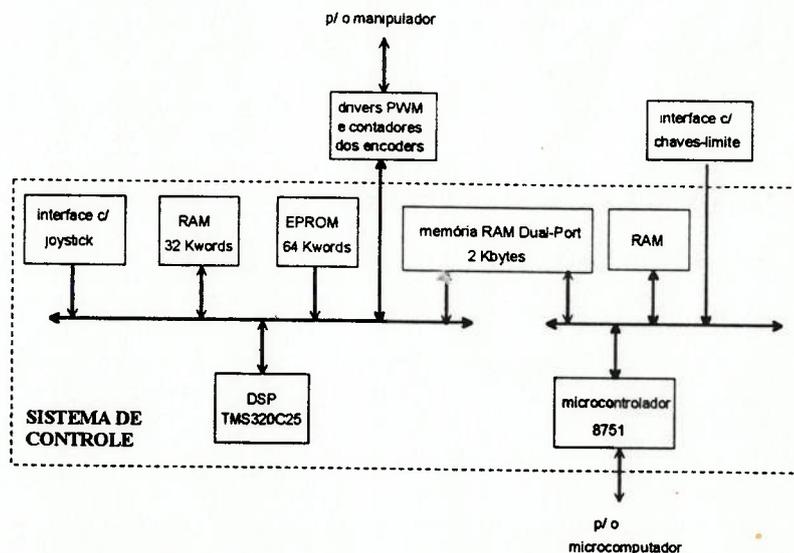


Figura 5.1. Diagrama de blocos do sistema de controle proposto.

Para a comunicação entre o microcontrolador e o DSP, foi projetada uma memória RAM *dual-port*, que é um espaço de memória comum aos dois processadores.

A figura 5.1 mostra uma foto do sistema de controle implementado.

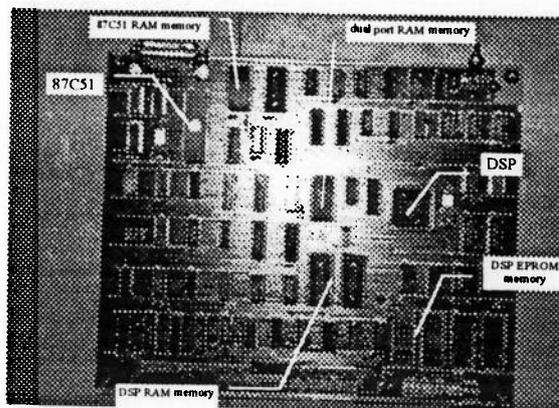


Figura 5.2. A implementação do sistema de controle.

A divisão das tarefas é feita da seguinte forma:

O microcontrolador 8751 realiza a comunicação serial com o microcomputador, decodifica comandos, envia dados e monitora as chaves de fim de curso do manipulador.

O DSP é responsável por gerar as trajetórias de movimento do manipulador, executar o algoritmo de controle de posição, amostrar as posições das juntas, acionar os *drivers* PWM (*Pulse Width Modulation*) e permitir o movimento do manipulador através de um joystick.

Este sistema de controle foi integrado com um robô manipulador desenvolvido no Departamento de Engenharia Mecânica da Escola Politécnica da USP (figura 5.3).

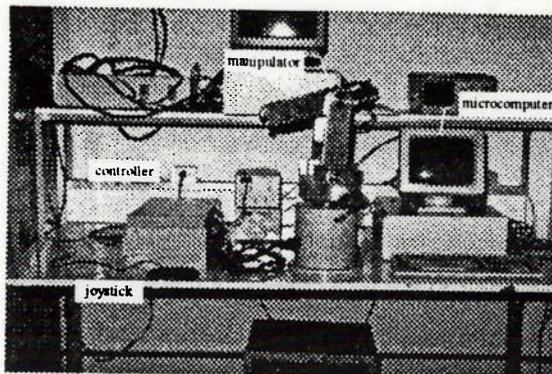


Figura 5.3. Sistema para ensino e treinamento em robótica

O manipulador deste sistema tem a sua estrutura mecânica em alumínio e possui 5 graus de liberdade, sendo cada grau acionado por um motor CC, com realimentação de posição feita através de codificadores ópticos incrementais. Os modelos cinemático e dinâmico deste manipulador, incluindo a matriz de inércia, estão nos Apêndices A, B e C.

5.2. Desenvolvimento do software

Os programas dos dois processadores têm uma estrutura *foreground/background* (SAVITZKY, 1985), ou seja, ficam executando um *polling loop* em *background*, enquanto interrupções são executadas em *foreground*.

5.2.1. Software do microcontrolador

O microcontrolador possui três interrupções, que estão as relacionadas com o canal serial, com a memória RAM *dual port* e com as chaves de fim de curso do manipulador.

As interrupções são acionadas na troca de informações com o microcomputador (no caso do canal serial) e com o DSP (no caso da RAM *dual port*), e no caso de alguma chave de fim de curso ser acionada.

Quando o microcontrolador não está atendendo a nenhuma interrupção, ele executa a decodificação dos comandos recebidos do computador, e controla o fluxo de dados enviados para o mesmo. Além disso, realiza a manipulação de alguns parâmetros antes de enviá-los para o DSP.

Os programas do microcontrolador são escritos na linguagem *assembly* do processador. Para a compilação do programa, é utilizado o compilador *cross-assembler* de domínio público chamado CAS.

5.2.2. *Software* do DSP

O DSP possui uma interrupção principal (de maior prioridade) acionada por um temporizador, que interrompe o microprocessador a cada 22 ms (que é o tempo de amostragem do sistema). Logo que ocorre o pedido de interrupção, o microprocessador realiza a leitura dos codificadores, calcula a próxima posição desejada e executa o algoritmo de controle para as cinco juntas. Estas tarefas são diretamente responsáveis pelo movimento do robô e devem ser completadas num intervalo de tempo menor que o intervalo de amostragem. No tempo restante (até que o relógio de tempo real do sistema interrompa novamente o processador), o DSP cuida de outras funções (o envio e recebimento de parâmetros através da memória RAM *dual port*, por exemplo). A figura 5.4 mostra a divisão do tempo de um ciclo de controle do DSP.

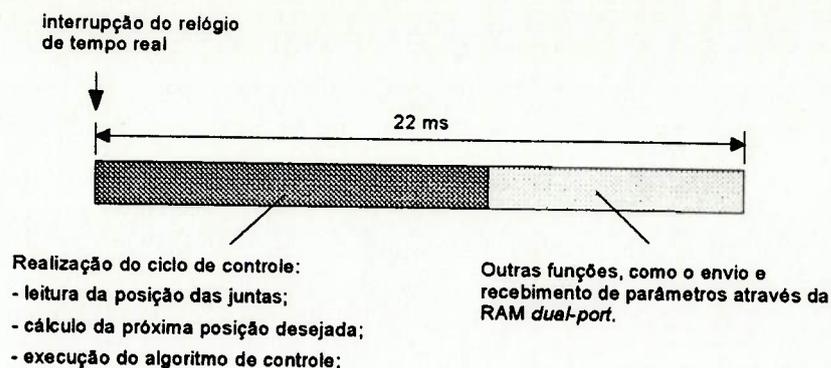


Figura 5.4. Um ciclo de controle do DSP.

O programa do DSP tem uma estrutura modularizada, de maneira que o estudante pode escrever apenas as funções relacionadas aos algoritmos de planejamento de trajetória e controle de posição, integrando-as com as funções já existentes que cuidam das operações básicas do sistema.

O programa está desenvolvido nas linguagens *assembly* e C (TEXAS INSTRUMENTS, 1990b; TEXAS INSTRUMENTS, 1990c). Os algoritmos de planejamento de trajetória e controle de posição podem ser escritos inteiramente na linguagem C.

Capítulo 6

Resultados e discussão

A implementação de alguns algoritmos de controle de robôs podem requerer um grande número de cálculos dentro do intervalo de amostragem do sistema. O primeiro bloco de testes foi realizado antes da implementação do sistema para verificar se o DSP teria a capacidade necessária para executar cálculos em tempo real. O segundo bloco de testes mostra um exemplo dos dados que podem ser obtidos com o sistema desenvolvido.

6.3. Avaliação da capacidade de execução dos algoritmos

Foram realizados alguns testes (SASAKI, 1992) com o objetivo de verificar se o DSP consegue executar os algoritmos de controle de interesse num tempo suficientemente pequeno para que o manipulador possa ser controlado.

Estes testes foram realizados por meio do *TMS320C2x SWDS (Software Development System)*, que é uma ferramenta residente no PC que permite desenvolver e depurar programas dedicados para os DSP's *TMS320C25* e *TMS320C26*. Esta ferramenta se constitui das seguintes partes (TEXAS INSTRUMENTS, 1990d; TEXAS INSTRUMENTS, 1990e):

- uma placa (instalada num *slot* do PC) que permite o acesso ao espaço de I/O em tempo real, a simulação de software e o suporte para o desenvolvimento de aplicações em software;
- um pacote de software que provê um ambiente para compilar, combinar arquivos-objeto em um módulo executável, carregar, executar programas na placa, e

depurar os programas no PC. A depuração pode ser feita acompanhando a execução do programa nos códigos em linguagens C e assembler.

Na primeira parte destas simulações, foram implementados no DSP algoritmos que exigem um menor número de cálculos (trajetória 3-5-3, controle PD). Para efeito de comparação, os resultados foram comparados com os obtidos com por um controlador anterior do sistema, que utiliza um processador 8088.

Na segunda parte, foram implementados cálculos que requerem um maior número de operações matemáticas por iteração, e são comumente utilizados dentro dos algoritmos de planejamento de trajetória e controle de posição de robôs: a cinemática inversa e o cálculo dos termos do modelo dinâmico do manipulador. A cinemática inversa é usada quando a trajetória é gerada no espaço cartesiano e os pontos gerados precisam ser transformados para coordenadas de juntas; os termos do modelo dinâmico do manipulador são necessários para os algoritmos baseados no modelo (vide capítulo anterior).

Na tabela 6.1 estão os resultados obtidos nos testes (SASAKI, 1992).

Tabela 6.1. Resultados dos testes.

Teste	tempo de uma iteração (3 juntas)	
	8088	DSP
1	11ms	0.3ms
2	-	5.9ms
3	-	6.7ms

- teste 1 - geração de trajetórias no espaço de juntas usando tabelas pré-calculadas, controle de posição PD.
- teste 2 - cinemática inversa para as 3 primeiras juntas do manipulador.
- teste 3 - cálculo de todos os termos do modelo dinâmico disponível do manipulador.

Pode-se observar pelo primeiro teste que o DSP apresentou um desempenho bem melhor que o 8088 (foi aproximadamente 30 vezes mais rápido).

As implementações 2 e 3 não foram realizadas para o 8088, pois não há no departamento um *kit* análogo ao *SWDS* do DSP. No entanto, a partir do primeiro resultado, pode-se esperar nos casos 2 e 3 um tempo bem maior que o período de amostragem necessário para controlar o sistema, que é de 22 ms (YAMAGUCHI; AMARAL, 1992). Este resultado impossibilita a implementação destes algoritmos em tempo real utilizando o 8088.

Estes testes mostram que o DSP é capaz de executar a cinemática inversa (que normalmente é utilizada para gerar as trajetórias no espaço cartesiano) e calcular todos os termos do modelo dinâmico do manipulador (que normalmente são utilizados nos algoritmos de controle baseados no modelo) em aproximadamente 13 ms⁴ (a soma dos tempos gastos nos testes), que é menor que o período de amostragem utilizado no robô (aproximadamente 22ms). Desta forma, pode-se afirmar que o controlador desenvolvido é capaz de executar os algoritmos que fazem uso destes cálculos, o que o torna adequado para implementar vários algoritmos dentre aqueles apresentados no capítulo 2.

6.4. Exemplos de movimentos realizados sob o controle do sistema desenvolvido

Além da capacidade de processamento, uma característica desejada para este sistema é que ele tenha algumas funções básicas implementadas, de modo que o estudante possa se limitar à implementação dos algoritmos de controle e planejamento da trajetória. O sistema implementado possui as funções básicas que permitem a monitoração do

⁴ Deve-se observar que estes tempos estão superestimados, pois por conveniência as simulações foram feitas usando ponto flutuante.

movimento do robô. Desta forma, é possível gravar em um arquivo de dados as posições do robô durante um movimento. As posições do robô são obtidas através dos codificadores ópticos incrementais do sistema. É possível também alterar os parâmetros do controlador, o que permite verificar a mudança do comportamento do robô com os parâmetros. As figuras 6.1 a 6.11 mostram alguns dados obtidos para diferentes entradas e parâmetros de controle.

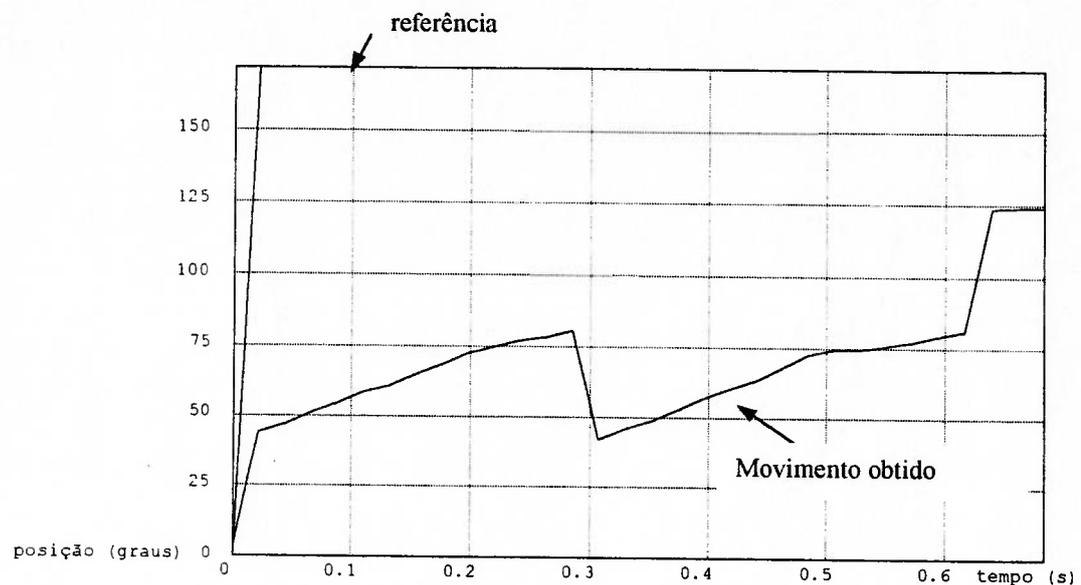


Figura 6.1. Resposta do motor de uma das juntas a uma entrada degrau (controlador P, com $K_p=1.5$).

As figuras 6.1 a 6.3 mostram o comportamento do motor de uma das juntas submetido a uma entrada degrau para diferentes valores no parâmetro K_p de um controlador P (Proporcional). Pode-se observar que, para um maior valor de K_p , o erro estático foi menor, o que é um resultado esperado.

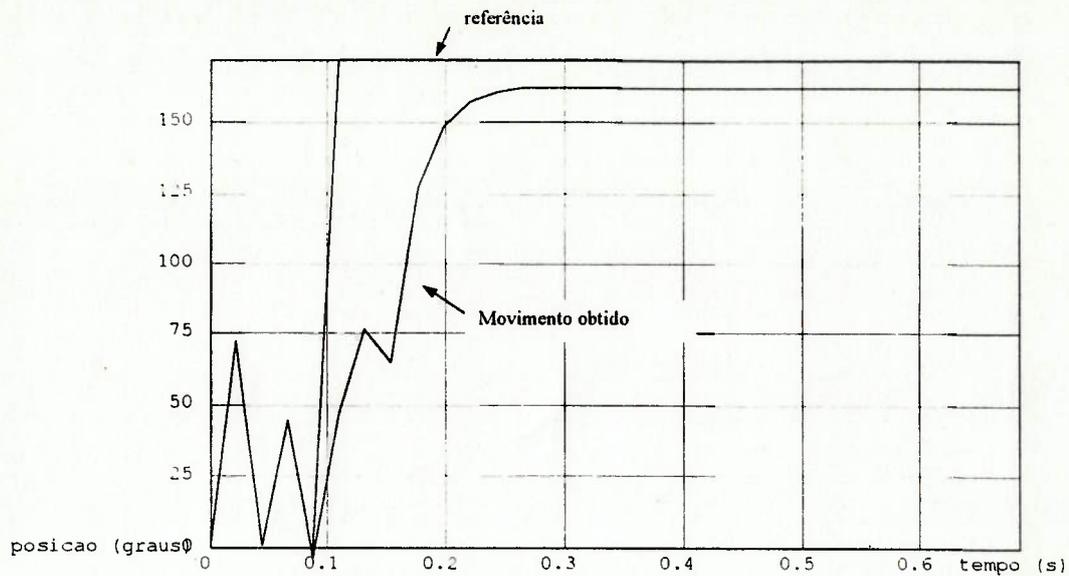


Figura 6.2. Resposta do motor de uma das juntas a uma entrada degrau (controlador P, com $K_p=7$).

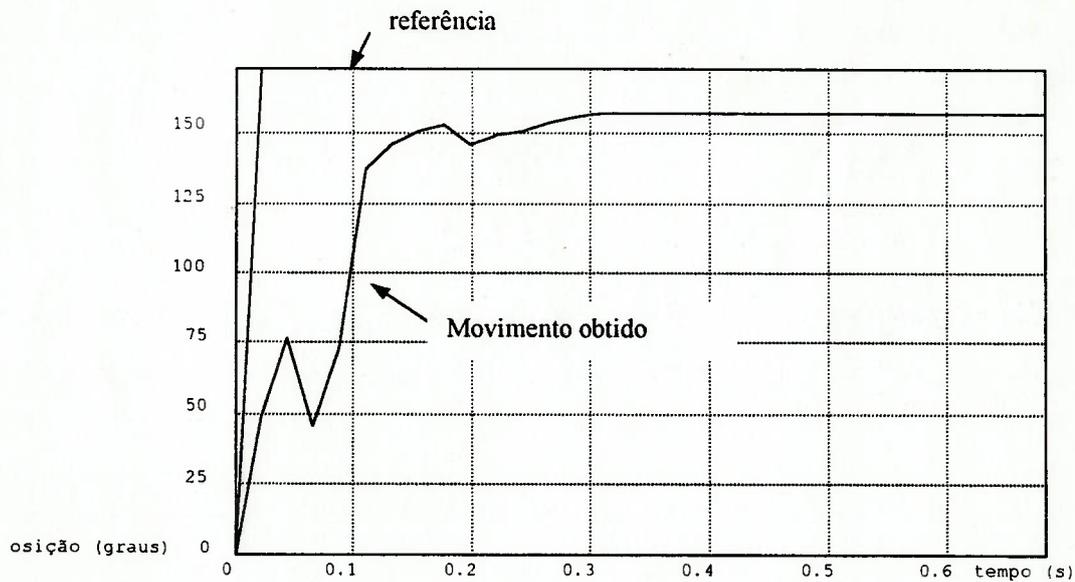


Figura 6.3. Resposta do motor de uma das juntas a uma entrada degrau (controlador P, com $K_p=7$).

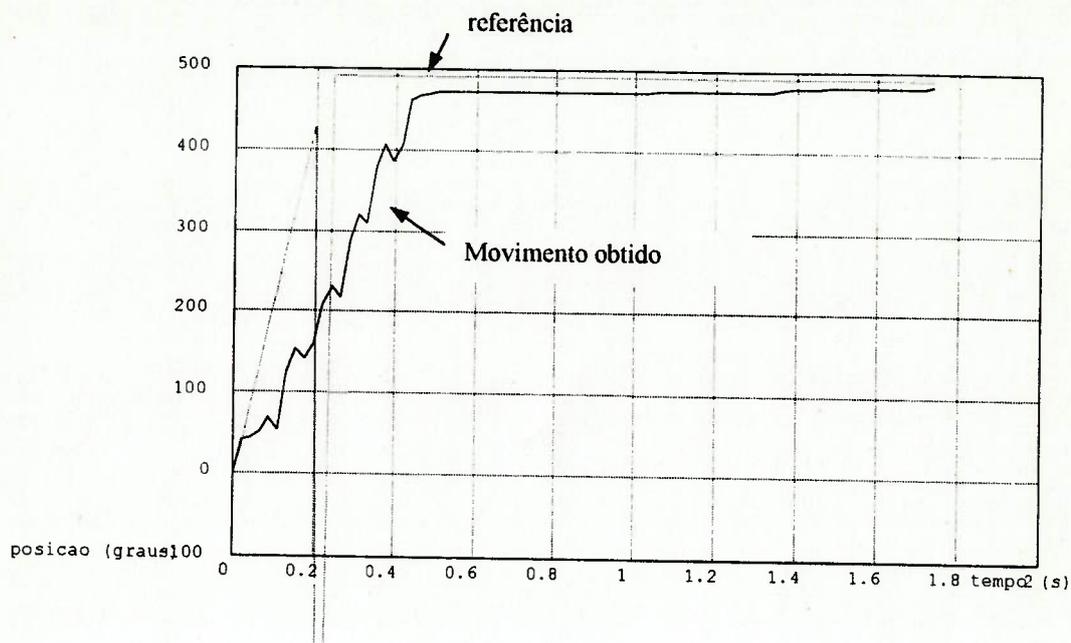


Figura 6.4. Resposta do motor da junta 3 a uma rampa (controlador PD, com $K_p=7$ e $K_d=1.5$).

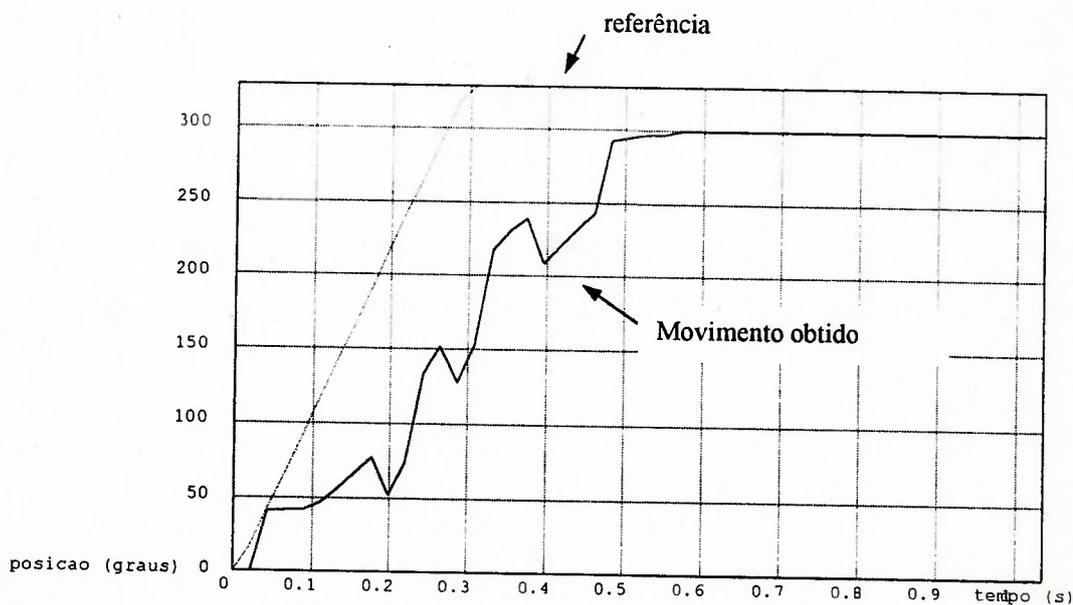


Figura 6.5. Resposta do motor da junta 3 a uma rampa (controlador PD, com $K_p=10$ e $K_d=3.5$).

As figuras 6.4 e 6.5 mostram a resposta do motor da junta 3 a uma entrada rampa. Estes testes foram feitos com o motor conectado às engrenagens de transmissão, mas

desconectado da junta. Desta forma, os valores de deslocamento correspondem ao deslocamento que a junta 3 teria se estivesse conectada ao motor (daí os valores tão altos de deslocamento que aparecem nestes gráficos).

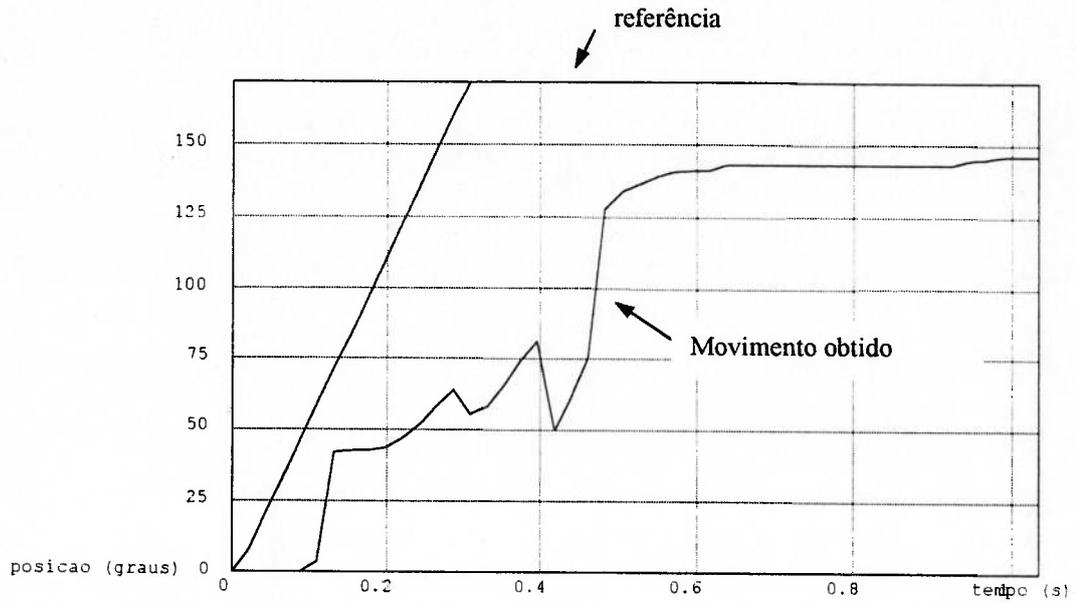


Figura 6.6. Resposta da junta 3 a uma entrada rampa (controlador PD, com $K_p=7$ e $K_d=3.5$).

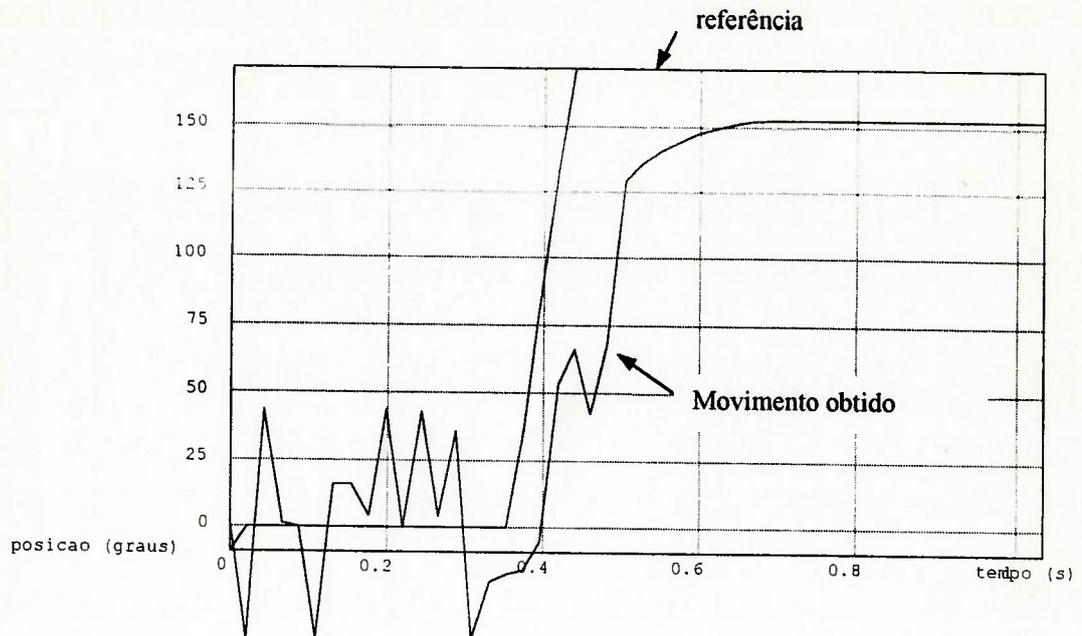


Figura 6.7. Resposta da junta 3 a uma rampa mais acentuada (controlador PD, com $K_p=7$ e $K_d=1.5$).

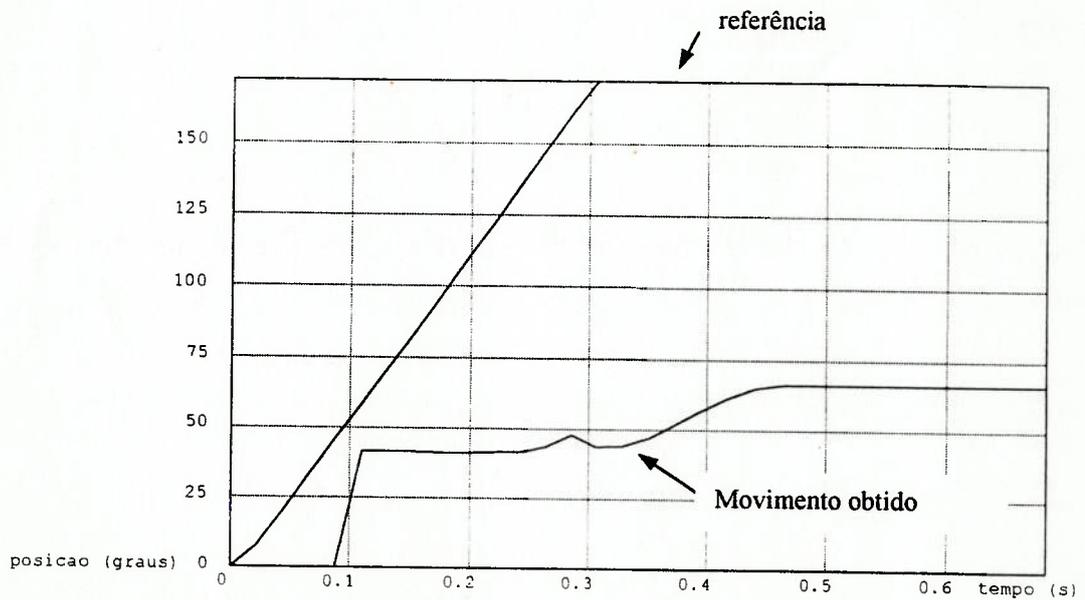


Figura 6.8. Resposta da junta 3 a uma entrada rampa (controlador PD, com $K_p=6$ e $K_d=3.5$).

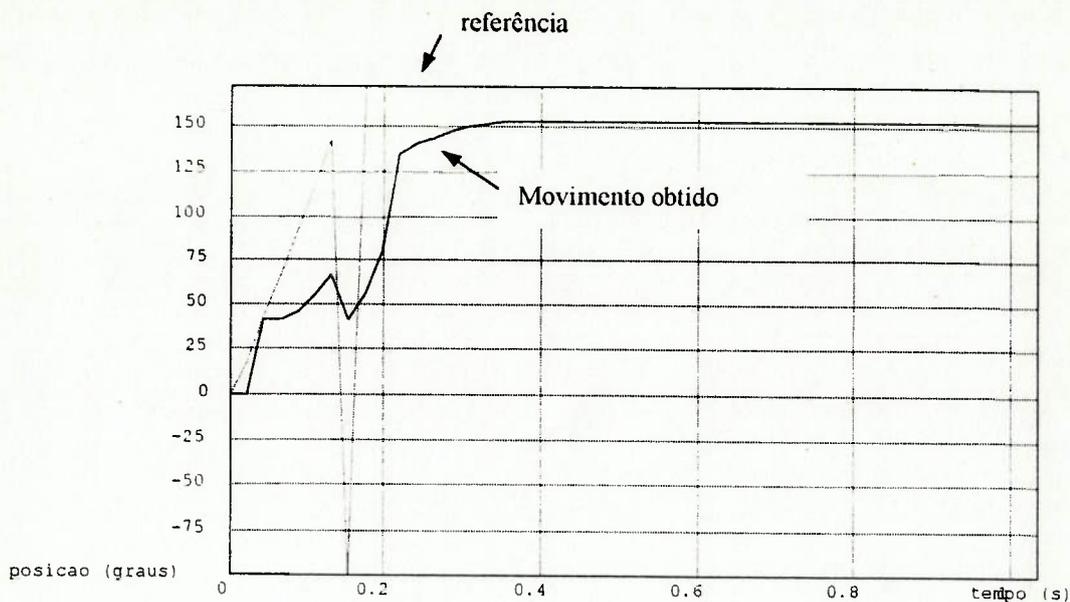


Figura 6.9. Resposta da junta 3 a uma rampa (controlador PD, com $K_p=7$ e $K_d=1.5$).

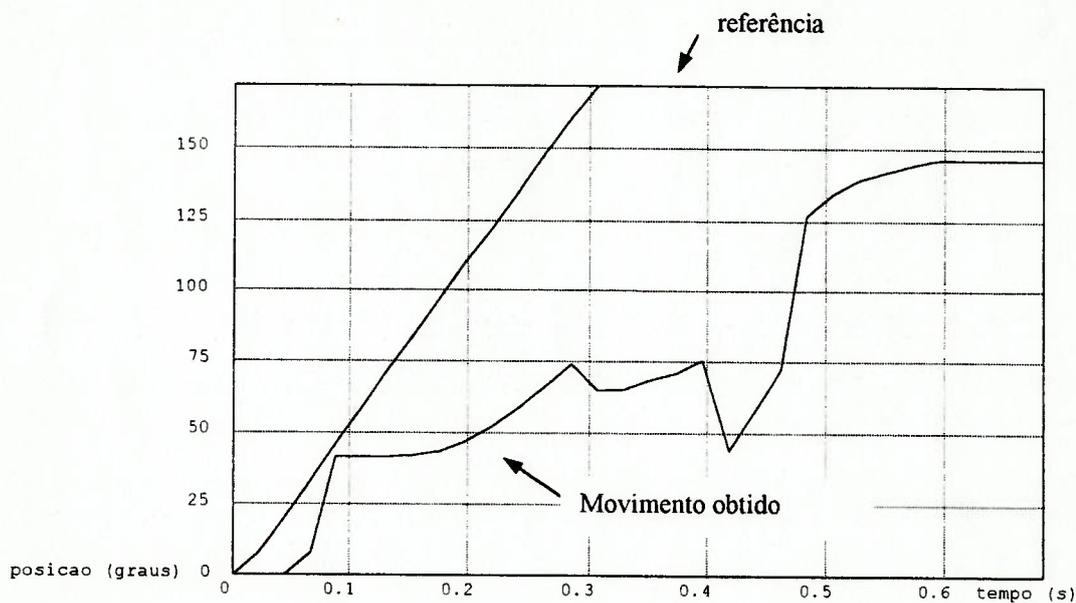


Figura 6.10. Resposta da junta 3 a uma entrada rampa (controlador PD, com $K_p=6$ e $K_d=3.5$).

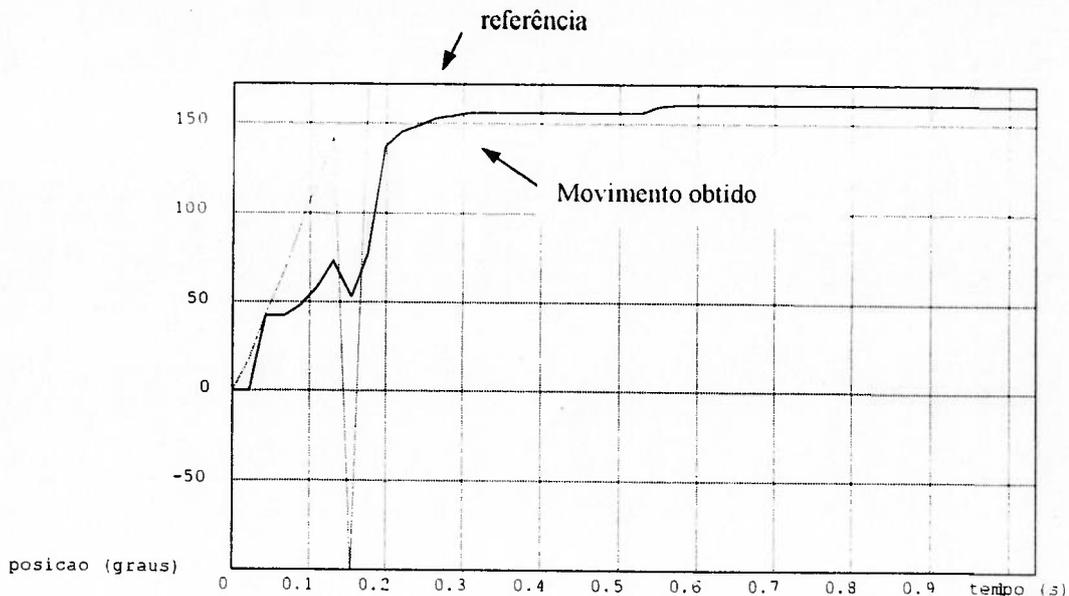


Figura 6.11. Resposta da junta 3 a uma rampa (controlador PD, com $K_p=10.5$ e $K_d=3.5$).

As figuras 6.6 a 6.11 mostram o comportamento da junta 3 submetido a uma entrada rampa, utilizando diferentes parâmetros em um controlador PD. Estes experimentos foram realizados medindo a posição angular do elo 3 em relação à horizontal, mantendo o elo 2 perpendicular à horizontal. Devido à configuração do braço, seria esperado que o erro de posição resultasse em outro sentido (com a posição real maior que a posição de referência) devido à componente gravitacional. No entanto, como a massa deste elo é pequena e a redução utilizada nesta junta é grande, é provável que outros fatores (como o atrito, por exemplo) tiveram uma maior influência neste caso.

Estes resultados são um exemplo da utilização das funções básicas do sistema, que permitem a monitoração do movimento do robô e a alteração dos parâmetros do controlador. Desta forma, o estudante pode se limitar à implementação do seu algoritmo de controle ou planejamento da trajetória e utilizar as funções já existentes no sistema de controle para obter os resultados.

Capítulo 7

Conclusão

O controle de movimento de um robô envolve o planejamento de sua trajetória (que gera as trajetórias desejadas) e o controle de sua posição (que calcula as entradas que devem ser aplicadas nos atuadores para que o robô siga a trajetória desejada).

Há diversas técnicas de controle de posição para manipuladores. O estudo de uma determinada técnica pode ser feita através de simulações, mas a realização de experimentos com um sistema real permite observar o comportamento do sistema diante de fatores que não são modelados (ou são difíceis de modelar), como ruído e folgas, por exemplo. Em geral, os sistemas didáticos comerciais não permitem que o usuário interfira no algoritmo de controle de posição do sistema.

A implementação de algumas destas técnicas de controle em um sistema real pode implicar em um grande número de cálculos aritméticos e trigonométricos em um intervalo de tempo limitado.

A proposta deste trabalho foi desenvolver um sistema de controle para robôs manipuladores com a capacidade numérica necessária para implementar diversos algoritmos de controle de posição presentes na literatura (FU et. al., 1987; MARTÍNEZ, 1991; ASADA; SLOTINE, 1986) e com a flexibilidade de permitir ao estudante implementar os algoritmos e observar os seus comportamentos sem necessitar de um conhecimento profundo do *hardware* do sistema.

O sistema de controle proposto foi desenvolvido utilizando um processador digital de sinais (DSP – *Digital Signal Processor*), que é um processador que tem sido utilizado em aplicações de controle devido à sua capacidade de executar algumas operações aritméticas com uma velocidade maior que em outros processadores (a função “multiplica-acumula”, por exemplo). A solução proposta envolveu ainda um microcontrolador Intel 8751, com quem o DSP divide as funções do sistema de controle. A divisão de funções permite que o DSP fique responsável pelas funções que envolvem diretamente o controle do manipulador. Outra arquitetura envolvendo envolvendo estes dois processadores poderia ter sido (deixar o DSP responsável apenas pelos cálculos, por exemplo, enquanto o 8751 interage com todos os periféricos do sistema). No caso deste trabalho, a arquitetura escolhida foi limitada pelo sistema didático com o qual o sistema de controle desenvolvido teria que ser integrado.

Alguns testes foram realizados para verificar se o DSP poderia executar alguns cálculos que demandam muitas operações matemáticas, e que estão presentes em vários algoritmos de planejamento de trajetória e controle de posição. Os cálculos utilizados para realizar esta avaliação foram a cinemática inversa e os termos do modelo dinâmico do manipulador. Estes testes mostraram que o DSP consegue realizar estes cálculos em um tempo suficientemente curto para que o manipulador disponível fosse controlado.

O *software* do sistema implementa várias funções básicas, como as funções de acesso aos periféricos (motores, codificadores etc.), as funções que enviam os dados das posições das juntas durante o movimento de um robô, e as funções que permitem a alteração de parâmetros do controlador e da trajetória. Alguns resultados foram obtidos através destes recursos, e foram apresentados na forma gráfica.

Desta forma, o sistema implementado apresenta a capacidade numérica necessária para a implementação de diversos algoritmos de controle presentes na literatura, e tem a flexibilidade (arquitetura aberta, funções de acesso aos periféricos do sistema e funções de monitoração do movimento já implementadas) que o torna adequado para o estudo experimental e ensino das técnicas de controle de robôs.

Referências Bibliográficas

- AHMED, I.; MESHKAT, S. Using DSPs in control. **Control Engineering**, pp. 46-47, Feb. 1988.
- ABDALLAH, C.; DAWSON, D.; DORATO, P.; JAMSHIDI, M. Survey of robust control for rigid robots. **IEEE Control Systems**, Feb. 1991.
- ASADA, H.; SLOTINE, J. J. **Robot analysis and control**. New York, John Wiley, 1986.
- BERGERMAN, M. **Controle robusto de posição de manipuladores mecânicos**. São Paulo, 1992. 175p. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo.
- DOE, Y.; SHINOJIMA, M.; YOSHIMURA, H. Microprocessor-based novel variable structure control fo robot manipulator. **IFAC 10th Triennial World Congress**. Munich, pp.283-287, 1987.
- DUBOWSKY, S.; DesFORGES, D. T. The application of model-reference adaptive control to robotic manipulators. **Journal of Dynamic Systems, Measurement, and Control**, v.101, p.193-200, Sept. 1979.
- FU, K. S.; GONZALEZ, R. C.; LEE, C. S. G. **Robotics: control, sensing, vision, and intelligence**. Singapura, McGraw-Hill, 1987. (McGraw-Hill series in CAD/CAM robotics and computer vision)
- HARASHIMA, F.; DOE, Y.; NAITOH, H. Status review-applications of modern control theory. **IECON**. 1984.
- HEMERLY, E.M. Aplicações em controle de uma arquitetura paralela baseada em processadores digital de sinal. **SBA: Controle & Automação**, vol.4, n.1, pp.38-46, Agosto-Setembro 1993.
- KOIVO, A. J.; GUO, T. H. Adaptive linear controller for robotic manipulators. **IEEE Transactions on Automatic Control**, v.AC-28, n.1, p.162-171, 1983.
- LINIMON, M. (1994) **Comp.realtime: Frequently Asked Questions**. Usenet news.answers. Available via anonymous ftp from rtfm.mit.edu in pub/usenet/comp.realtime. 15+ pages
- MARTÍNEZ, R. K. **Control de movimiento de robots manipuladores**. Centro de Sistemas de Manufactura, Instituto Tecnológico y de Estudios Superiores de Monterrey, 1991.
- ORTEGA, R.; SPONG, M. Adaptive motion control of rigid robots: a tutorial. **Automatica**, v.25, n.6, p.877-888, 1989.
- PADEN, B.; PANJA, R. Globally asymptotically stable PD+ controller for robot manipulators, **International Journal of Control**, vol.47, no.6, pp.1697-1712, 1988.
- PAUL, R. P. **Robot manipulators: mathematics, programming and control**. MIT Press, 1981.
- ROBERTS, R.A.; MULLIS, C.T. **Digital signal processing**. Addison-Wesley, 1987.
- SALANT, M. A. **Introdução à robótica**. São Paulo, McGraw-Hill e Makron Books, 1991.

- SASAKI, T.M. **Estudos sobre o uso de um processador digital de sinais para geração de trajetórias de robôs.** Relatório Interno do Departamento de Engenharia Mecânica da EPUSP, São Paulo, 1992.
- SAVITZKY, S. **Real time microprocessor systems.** New York, Van Nostrand Reinhold Company, 1985.
- SLIVINSKY, C. **...** compensation and implementation with the **...** ons with the TMS320 family. 1988.
- SPONG, M. W. **...** nics and control. New York, John Wiley, 1989.
- TEXAS INSTR **...** ide. cap. 6, 1990a.
- TEXAS INSTRUMENTS, **TMS320 TACU-p...** SP assembly language tools. 1990d.
- TEXAS INSTRUMENTS, **TMS320C2x/C5x optimizing C compiler.** 1990c.
- TEXAS INSTRUMENTS, **TMS320C2x SWDS.** 1990d.
- TEXAS INSTRUMENTS, **TMS320C2x C debugger.** 1990e.
- WHITCOMB, L.L.; RIZZI, A.A.; KODITSCHKEK, D.E. Comparative Experiments with a new adaptive controller for robot arms. **IEEE Transactions on Robotics and Automation**, vol.9, n.1, pp.59-70, Feb. 1993.
- WHITNEY, D.E. Resolved motion rate control of manipulators and human prostheses. **IEEE Transactions on Man-Machine System**, vol. MMS-10, pp 47-53, Jun. 1969.
- WHITNEY, D.E. The mathematics of coordinated control of prosthetic arms and manipulators. **Journal of Dynamic Systems, Measurement, and Control**, pp 303-309, Dec. 1972.
- YAMAGUCHI, M., AMARAL, J.A.A. **Algoritmo de Controle em Tempo Real para Robô Didático.** Trabalho de Formatura do Departamento de Engenharia Mecânica da Escola Politécnica da USP, sob a orientação do Prof. Jun Okamoto Jr., São Paulo, 1992.

Apêndice A

Modelo cinemático

Conforme o método de Denavit-Hartenberg (FU et. al., 1987, cap.2), foram alocados os sistemas de coordenadas nos elos do robô. A tabela abaixo mostra os parâmetros obtidos para cada junta.

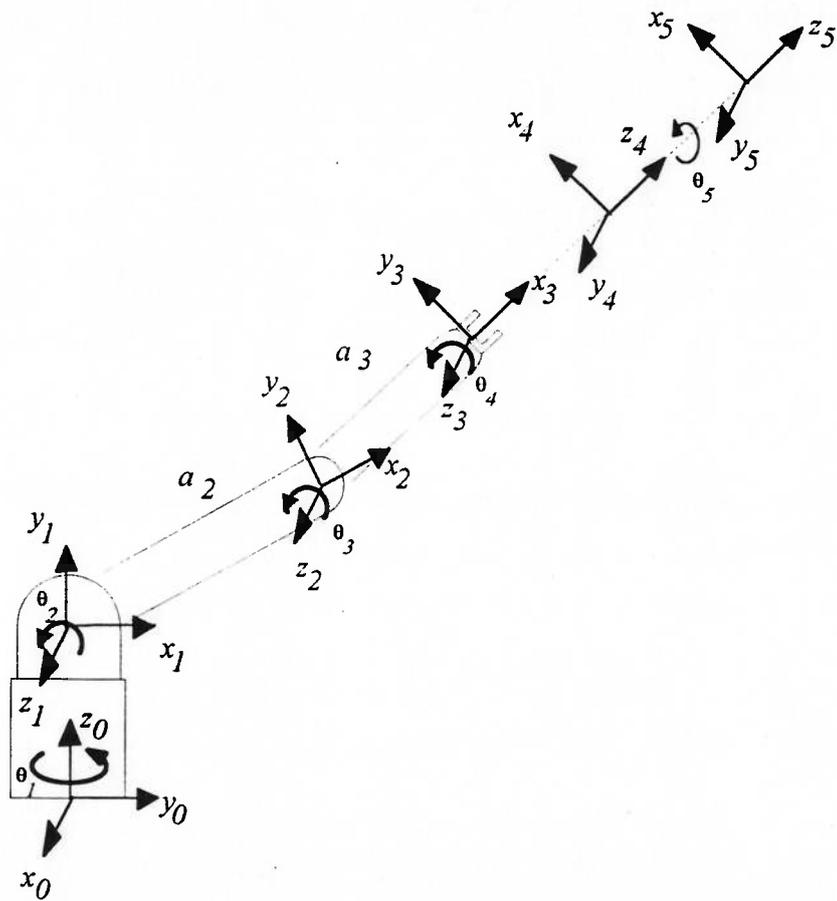


Figura A.1. Estabelecimento dos sistemas de coordenadas.

Tabela A.1. Parâmetros de Denavit-Hartenberg.

junta i	θ_i	α_i	a_i (cm)	d_i (cm)
1	q_1	90°	0	d_1
2	q_2	0°	26	0.5
3	q_3	0°	22	0
4	q_4	90°	0	0
5	q_5	0°	0	0

As matrizes de transformação homogênea são dadas por:

$${}^0\mathbf{A}_1 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.1})$$

$${}^1\mathbf{A}_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 26\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & 26\sin\theta_2 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.2})$$

$${}^2\mathbf{A}_3 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 22\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & 22\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3})$$

$${}^3\mathbf{A}_4 = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & 0 \\ \sin\theta_4 & 0 & -\cos\theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.4})$$

$${}^4\mathbf{A}_5 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.5})$$

Apêndice B

Obtenção da matriz de inércia

Para a obtenção da matriz de inércia, foi utilizado o seguinte procedimento:

- para cada segmento do manipulador, foi calculada a inércia de cada peça em relação ao sistema de coordenadas alocado no referido segmento;
- a soma das inércias de todas as peças do segmento resultou na matriz de inércia do segmento.

B.1. Segmento 1

Peças que compõem o segmento:

1. motor + redutor;
2. fundo da base + suporte;
3. engrenagem;
4. disco para suporte;
5. laterais da base;
6. rolamentos;
7. mancais.

Motor + redutor

$$\begin{aligned}
 {}^1m_1 &= 1.0 \text{ kg} \\
 {}^1J_{x_1x_1} &= 183 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{y_1y_1} &= 201 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{z_1z_1} &= 55 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{x_1y_1} &= 26.1 \text{ kg} \cdot \text{cm}^2 \\
 ({}^1\bar{x}_1, {}^1\bar{y}_1, {}^1\bar{z}_1) &= (-5.8, -4.5, 0) \text{ cm}
 \end{aligned}
 \tag{B.6}$$

Fundo da base + suporte

$$\begin{aligned}
 {}^2m_1 &= 0.95 \text{ kg} \\
 {}^2J_{x_1x_1} &= 74 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{y_1y_1} &= 48 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{z_1z_1} &= 97 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{x_1y_1} &= 30.4 \text{ kg} \cdot \text{cm}^2 \\
 ({}^2\bar{x}_1, {}^2\bar{y}_1, {}^2\bar{z}_1) &= (-4, -8, 0) \text{ cm}
 \end{aligned}
 \tag{B.7}$$

Engrenagem

$$\begin{aligned}
 {}^3m_1 &= 0.08 \text{ kg} \\
 {}^3J_{x_1x_1} &= 1 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{y_1y_1} &= 1 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{z_1z_1} &= 0.33 \text{ kg} \cdot \text{cm}^2 \\
 ({}^3\bar{x}_1, {}^3\bar{y}_1, {}^3\bar{z}_1) &= (0, 0, -3) \text{ cm}
 \end{aligned}
 \tag{B.8}$$

Disco para suporte

$$\begin{aligned}
 {}^4m_1 &= 0.5 \text{ kg} \\
 {}^4J_{x_1x_1} &= 35 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{y_1y_1} &= 5 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{z_1z_1} &= 35 \text{ kg} \cdot \text{cm}^2 \\
 ({}^4\bar{x}_1, {}^4\bar{y}_1, {}^4\bar{z}_1) &= (0, -8, 0) \text{ cm}
 \end{aligned}
 \tag{B.9}$$

Laterais da base

$$\begin{aligned}
 {}^5m_1 &= 0.4 \text{ kg} \\
 {}^5J_{x_1x_1} &= 24 \text{ kg} \cdot \text{cm}^2 \\
 {}^5J_{y_1y_1} &= 30 \text{ kg} \cdot \text{cm}^2 \\
 {}^5J_{z_1z_1} &= 20 \text{ kg} \cdot \text{cm}^2 \\
 {}^5J_{x_1y_1} &= 1.98 \text{ kg} \cdot \text{cm}^2 \\
 ({}^5\bar{x}_1, {}^5\bar{y}_1, {}^5\bar{z}_1) &= (-3.3, -1.5, 0) \text{ cm}
 \end{aligned} \tag{B.10}$$

Rolamentos

$$\begin{aligned}
 {}^6m_1 &= 0.08 \text{ kg} \\
 {}^6J_{x_1x_1} &= 3.8 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{y_1y_1} &= 3.8 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{z_1z_1} &= 0.037 \text{ kg} \cdot \text{cm}^2 \\
 ({}^6\bar{x}_1, {}^6\bar{y}_1, {}^6\bar{z}_1) &= (0, 0, 0) \text{ cm}
 \end{aligned} \tag{B.11}$$

Mancais

$$\begin{aligned}
 {}^7m_1 &= 0.06 \text{ kg} \\
 {}^7J_{x_1x_1} &= 2.93 \text{ kg} \cdot \text{cm}^2 \\
 {}^7J_{y_1y_1} &= 2.93 \text{ kg} \cdot \text{cm}^2 \\
 {}^7J_{z_1z_1} &= 0.144 \text{ kg} \cdot \text{cm}^2 \\
 ({}^7\bar{x}_1, {}^7\bar{y}_1, {}^7\bar{z}_1) &= (0, 0, 0) \text{ cm}
 \end{aligned} \tag{B.12}$$

B.2. Segmento 2

Peças que compõem o segmento:

1. laterais;
2. tampas superior e inferior;
3. engrenagem;
4. eixo;
5. rolamentos;
6. mancais.

Laterais

$$\begin{aligned}
 {}^1m_2 &= 0.36 \text{ kg} \\
 {}^1J_{x_2x_2} &= 11.3 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{y_2y_2} &= 95 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{z_2z_2} &= 89 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{x_2z_2} &= -2.14 \text{ kg} \cdot \text{cm}^2 \\
 ({}^1\bar{x}_2, {}^1\bar{y}_2, {}^1\bar{z}_2) &= (-13.2, 0, -0.45) \text{ cm}
 \end{aligned} \tag{B.13}$$

Tampas superior e inferior

$$\begin{aligned}
 {}^2m_2 &= 0.16 \text{ kg} \\
 {}^2J_{x_2x_2} &= 6 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{y_2y_2} &= 68 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{z_2z_2} &= 70 \text{ kg} \cdot \text{cm}^2 \\
 ({}^2\bar{x}_2, {}^2\bar{y}_2, {}^2\bar{z}_2) &= (-20, 0, 0) \text{ cm}
 \end{aligned} \tag{B.14}$$

Engrenagem

$$\begin{aligned}
 {}^3m_2 &= 0.08 \text{ kg} \\
 {}^3J_{x_2x_2} &= 1.47 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{y_2y_2} &= 55.5 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{z_2z_2} &= 54.4 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{x_2z_2} &= 8.32 \text{ kg} \cdot \text{cm}^2 \\
 ({}^3\bar{x}_2, {}^3\bar{y}_2, {}^3\bar{z}_2) &= (-26, 0, 4) \text{ cm}
 \end{aligned} \tag{B.15}$$

Eixo

$$\begin{aligned}
 {}^4m_2 &= 0.07 \text{ kg} \\
 {}^4J_{x_2x_2} &= 1 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{y_2y_2} &= 48.3 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{z_2z_2} &= 48 \text{ kg} \cdot \text{cm}^2 \\
 ({}^4\bar{x}_2, {}^4\bar{y}_2, {}^4\bar{z}_2) &= (-26, 0, 0) \text{ cm}
 \end{aligned} \tag{B.16}$$

Rolamentos

$$\begin{aligned} {}^5m_2 &= 0.08 \text{ kg} \\ {}^5J_{x_2x_2} &= 2.2 \text{ kg} \cdot \text{cm}^2 \\ {}^5J_{y_2y_2} &= 2.2 \text{ kg} \cdot \text{cm}^2 \\ {}^5J_{z_2z_2} &= 0.037 \text{ kg} \cdot \text{cm}^2 \\ ({}^5\bar{x}_2, {}^5\bar{y}_2, {}^5\bar{z}_2) &= (0, 0, 0) \text{ cm} \end{aligned} \tag{B.17}$$

Mancais

$$\begin{aligned} {}^6m_2 &= 0.06 \text{ kg} \\ {}^6J_{x_2x_2} &= 1.74 \text{ kg} \cdot \text{cm}^2 \\ {}^6J_{y_2y_2} &= 1.74 \text{ kg} \cdot \text{cm}^2 \\ {}^6J_{z_2z_2} &= 0.24 \text{ kg} \cdot \text{cm}^2 \\ ({}^6\bar{x}_2, {}^6\bar{y}_2, {}^6\bar{z}_2) &= (0, 0, 0) \text{ cm} \end{aligned} \tag{B.18}$$

B.3. Segmento 3

Partes que compõem o segmento:

1. laterais;
2. tampas superior e inferior;
3. motores;
4. eixos dos motores;
5. eixo que liga o segmento 2 ao segmento 3;
6. suportes dos eixos dos motores;
7. polia;
8. engrenagens ligadas aos eixos dos motores;
9. placas dos suportes dos eixos dos motores;
10. rolamentos;
11. engrenagens cônicas.

Laterais

$$\begin{aligned}
 {}^1m_3 &= 0.24 \text{ kg} \\
 {}^1J_{x_3x_3} &= 4.56 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{y_3y_3} &= 37 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{z_3z_3} &= 34.8 \text{ kg} \cdot \text{cm}^2 \\
 ({}^1\bar{x}_3, {}^1\bar{y}_3, {}^1\bar{z}_3) &= (-9, 0, 0) \text{ cm}
 \end{aligned}
 \tag{B.19}$$

Tampas superior e inferior

$$\begin{aligned}
 {}^2m_3 &= 0.18 \text{ kg} \\
 {}^2J_{x_3x_3} &= 3.8 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{y_3y_3} &= 43 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{z_3z_3} &= 42 \text{ kg} \cdot \text{cm}^2 \\
 ({}^2\bar{x}_3, {}^2\bar{y}_3, {}^2\bar{z}_3) &= (-14.3, 0, 0) \text{ cm}
 \end{aligned}
 \tag{B.20}$$

Motores

$$\begin{aligned}
 {}^3m_3 &= 0.72 \text{ kg} \\
 {}^3J_{x_3x_3} &= 7.3 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{y_3y_3} &= 866 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{z_3z_3} &= 870 \text{ kg} \cdot \text{cm}^2 \\
 {}^3J_{y_3z_3} &= -2.808 \text{ kg} \cdot \text{cm}^2 \\
 ({}^3\bar{x}_3, {}^3\bar{y}_3, {}^3\bar{z}_3) &= (-30.3, 0, 0) \text{ cm}
 \end{aligned}
 \tag{B.21}$$

Eixos dos motores

$$\begin{aligned}
 {}^4m_3 &= 0.14 \text{ kg} \\
 {}^4J_{x_3x_3} &= 0.04 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{y_3y_3} &= 23.8 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{z_3z_3} &= 23.6 \text{ kg} \cdot \text{cm}^2 \\
 {}^4J_{y_3z_3} &= -0.264 \text{ kg} \cdot \text{cm}^2 \\
 ({}^4\bar{x}_3, {}^4\bar{y}_3, {}^4\bar{z}_3) &= (-11.8, 0, 0) \text{ cm}
 \end{aligned}
 \tag{B.22}$$

Eixo que liga o segmento 2 ao segmento 3

$$\begin{aligned}
 {}^5m_3 &= 0.03 \text{ kg} \\
 {}^5J_{x_3x_3} &= 2809 \text{ kg} \cdot \text{cm}^2 \\
 {}^5J_{y_3y_3} &= 14.8 \text{ kg} \cdot \text{cm}^2 \\
 {}^5J_{z_3z_3} &= 14.52 \text{ kg} \cdot \text{cm}^2 \\
 ({}^5\bar{x}_3, {}^5\bar{y}_3, {}^5\bar{z}_3) &= (-22, 0, 0) \text{ cm}
 \end{aligned} \tag{B.23}$$

Suportes dos eixos dos motores

$$\begin{aligned}
 {}^6m_3 &= 0.06 \text{ kg} \\
 {}^6J_{x_3x_3} &= 0.1374 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{y_3y_3} &= 2.42 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{z_3z_3} &= 2.11 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{x_3y_3} &= 0.027 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{y_3z_3} &= -0.036 \text{ kg} \cdot \text{cm}^2 \\
 {}^6J_{x_3z_3} &= -0.0648 \text{ kg} \cdot \text{cm}^2 \\
 ({}^6\bar{x}_3, {}^6\bar{y}_3, {}^6\bar{z}_3) &= (-5.8, 0, 0) \text{ cm}
 \end{aligned} \tag{B.24}$$

Polia

$$\begin{aligned}
 {}^7m_3 &= 0.10 \text{ kg} \\
 {}^7J_{x_3x_3} &= 2.16 \text{ kg} \cdot \text{cm}^2 \\
 {}^7J_{y_3y_3} &= 50.6 \text{ kg} \cdot \text{cm}^2 \\
 {}^7J_{z_3z_3} &= 48.6 \text{ kg} \cdot \text{cm}^2 \\
 {}^7J_{x_3z_3} &= 9.9 \text{ kg} \cdot \text{cm}^2 \\
 ({}^7\bar{x}_3, {}^7\bar{y}_3, {}^7\bar{z}_3) &= (-22, 0, -4.5) \text{ cm}
 \end{aligned} \tag{B.25}$$

Engrenagens ligadas aos eixos dos motores

$$\begin{aligned}
 {}^8m_3 &= 0.06 \text{ kg} \\
 {}^8I_{x_3x_3} &= 0.109 \text{ kg} \cdot \text{cm}^2 \\
 {}^8I_{y_3y_3} &= 0.640 \text{ kg} \cdot \text{cm}^2 \\
 {}^8I_{z_3z_3} &= 0.553 \text{ kg} \cdot \text{cm}^2 \\
 ({}^8\bar{x}_3, {}^8\bar{y}_3, {}^8\bar{z}_3) &= (-3, 0, 0) \text{ cm}
 \end{aligned} \tag{B.26}$$

Placas dos suportes dos eixos dos motores

$$\begin{aligned}
 {}^9m_3 &= 0.06 \text{ kg} \\
 {}^9J_{x_3x_3} &= 0.206 \text{ kg} \cdot \text{cm}^2 \\
 {}^9J_{y_3y_3} &= 1.974 \text{ kg} \cdot \text{cm}^2 \\
 {}^9J_{z_3z_3} &= 1.87 \text{ kg} \cdot \text{cm}^2 \\
 ({}^9\bar{x}_3, {}^9\bar{y}_3, {}^9\bar{z}_3) &= (-5.25, 0, 0) \text{ cm}
 \end{aligned} \tag{B.27}$$

Rolamentos

$$\begin{aligned}
 {}^{10}m_3 &= 0.24 \text{ kg} \\
 {}^{10}J_{x_3x_3} &= 2.04 \text{ kg} \cdot \text{cm}^2 \\
 {}^{10}J_{y_3y_3} &= 2.04 \text{ kg} \cdot \text{cm}^2 \\
 {}^{10}J_{z_3z_3} &= 1.11 \text{ kg} \cdot \text{cm}^2 \\
 ({}^{10}\bar{x}_3, {}^{10}\bar{y}_3, {}^{10}\bar{z}_3) &= (0, 0, 0) \text{ cm}
 \end{aligned} \tag{B.28}$$

Engrenagens cônicas

$$\begin{aligned}
 {}^{11}m_3 &= 0.24 \text{ kg} \\
 {}^{11}J_{x_3x_3} &= 2.56 \text{ kg} \cdot \text{cm}^2 \\
 {}^{11}J_{y_3y_3} &= 2.56 \text{ kg} \cdot \text{cm}^2 \\
 {}^{11}J_{z_3z_3} &= 0.95 \text{ kg} \cdot \text{cm}^2 \\
 ({}^{11}\bar{x}_3, {}^{11}\bar{y}_3, {}^{11}\bar{z}_3) &= (0, 0, 0) \text{ cm}
 \end{aligned} \tag{B.29}$$

B.4. Segmento 4

Partes que compõem o segmento:

1. suporte do punho;
2. eixo do pulso;

Suporte do punho

$$\begin{aligned}
 {}^1m_4 &= 0.03 \text{ kg} \\
 {}^1J_{x_4x_4} &= 0.045 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{y_4y_4} &= 0.03 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{z_4z_4} &= 0.045 \text{ kg} \cdot \text{cm}^2 \\
 ({}^1\bar{x}_4, {}^1\bar{y}_4, {}^1\bar{z}_4) &= (0, 0, 0) \text{ cm}
 \end{aligned} \tag{B.30}$$

Eixo do pulso

$$\begin{aligned}
 {}^2m_4 &= 0.04 \text{ kg} \\
 {}^2J_{x_4x_4} &= 0.25 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{y_4y_4} &= 0.25 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{z_4z_4} &= 0.013 \text{ kg} \cdot \text{cm}^2 \\
 ({}^2\bar{x}_4, {}^2\bar{y}_4, {}^2\bar{z}_4) &= (0, 0, 2.3) \text{ cm}
 \end{aligned}
 \tag{B.31}$$

B.5. Segmento 5

Partes que compõem o segmento:

1. engrenagem do pulso;
2. suporte da pinça;
3. carga.

Engrenagem do pulso

$$\begin{aligned}
 {}^1m_5 &= 0.08 \text{ kg} \\
 {}^1J_{x_5x_5} &= 0.52 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{y_5y_5} &= 0.52 \text{ kg} \cdot \text{cm}^2 \\
 {}^1J_{z_5z_5} &= 0.1 \text{ kg} \cdot \text{cm}^2 \\
 ({}^1\bar{x}_5, {}^1\bar{y}_5, {}^1\bar{z}_5) &= (0, 0, 2.4) \text{ cm}
 \end{aligned}
 \tag{B.32}$$

Suporte da pinça

$$\begin{aligned}
 {}^2m_5 &= 0.04 \text{ kg} \\
 {}^2J_{x_5x_5} &= 0.53 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{y_5y_5} &= 0.53 \text{ kg} \cdot \text{cm}^2 \\
 {}^2J_{z_5z_5} &= 0.1 \text{ kg} \cdot \text{cm}^2 \\
 ({}^2\bar{x}_5, {}^2\bar{y}_5, {}^2\bar{z}_5) &= (0, 0, 3.5) \text{ cm}
 \end{aligned}
 \tag{B.33}$$

Carga

Supondo que a carga tem massa uniforme e que os seus elementos cruzados de inércia sejam nulos.

$$\begin{aligned}
 {}^3m_s &= m_c \\
 {}^3J_{x_s x_s} &= J_{xx_c} \\
 {}^3J_{y_s y_s} &= J_{yy_c} \\
 {}^3J_{z_s z_s} &= J_{zz_c} \\
 ({}^3\bar{x}_s, {}^3\bar{y}_s, {}^3\bar{z}_s) &= (0, 0, \bar{z}_c)
 \end{aligned}
 \tag{B.34}$$

B.6. Resultado final

Tabela B.2. Massa de cada junta.

	m (kg)	\bar{x} (cm)	\bar{y} (cm)	\bar{z} (cm)
junta 1	3.07	-3.6	-5.4	-0.1
junta 2	0.81	-14.6	0	0.2
junta 3	2.07	-15.4	0	-0.2
junta 4	0.07	0	0	1.3
junta 5	$0.12+m_c$	0	0	\bar{z}_s

sendo que $\bar{z}_s = \frac{0.332 + m_c \cdot \bar{z}_c}{0.12 + m_c}$.

Tabela 3. Valores dos elementos da matriz de inércia (valores em kg · cm²).

	J_{xx}	J_{yy}	J_{zz}	$J_{xy} = J_{yx}$	$J_{xz} = J_{zx}$	$J_{yz} = J_{zy}$
junta 1	323.73	291.73	207.511	58.48	0	0
junta 2	23.71	270.74	261.677	0	6.18	0
junta 3	23.1933	1044.834	1040.11 3	0.027	9.8352	-3.108
junta 4	0.295	0.28	0.058	0	0	0
junta 5	$1.05+J_{xx_c}$	$1.05+J_{yy_c}$	$0.2+J_{zz_c}$	0	0	0

Apêndice C

Modelo dinâmico

O modelo obtido leva em conta apenas as três primeiras juntas do manipulador.

São adotadas as seguintes constantes:

$$k_1 = a_3^2 m_3 + 2a_3 m_3 x_{c3} + J_{xx3} - J_{yy3}$$

$$k_2 = a_2 a_3 m_3 + a_2 m_3 x_{c3}$$

$$k_3 = J_{xy3}$$

$$k_4 = J_{yy3}$$

$$k_5 = d_2^2 m_2 + d_2^2 m_3 + 2d_2 m_2 z_{c2} + 2d_2 m_3 z_{c3} + J_{xx1} + J_{zz1} + J_{yy2} + J_{zz2} + J_{zz3}$$

$$k_6 = J_{yz3}$$

$$k_7 = a_2 d_2 m_2 + a_2 d_2 m_3 + a_2 m_2 z_{c2} + a_2 m_3 z_{c3} + d_2 m_2 x_{c2} + J_{xz2}$$

$$k_8 = a_3 d_2 m_3 + a_3 m_3 z_{c3} + d_2 m_3 x_{c3} + J_{xz3}$$

$$k_9 = a_2^2 m_2 + a_2^2 m_3 + 2a_2 m_2 x_{c2} + J_{xx2} + J_{yy2}$$

$$k_{10} = a_3^2 m_3 + 2a_3 m_3 x_{c3} + J_{xx3} + J_{yy3}$$

$$k_{11} = a_2^2 m_2 + a_2^2 m_3 + 2a_2 m_2 x_{c2} + J_{xx2} - J_{yy2}$$

$$k_{12} = a_3 d_2 m_3 + a_3 m_2 z_{c2} + d_2 m_3 x_{c3} + J_{xy3}$$

$$k_{13} = a_3 g m_3 + g m_3 x_{c3}$$

$$k_{14} = a_2 g m_2 + a_2 g m_3 + g m_2 x_{c2}$$

Também são adotadas as seguintes funções:

$$f_1 = c_{23}^2$$

$$f_2 = 2c_2c_{23}$$

$$f_3 = -s_{2233}$$

$$f_4 = s_2^2$$

$$f_5 = -c_{23}$$

$$f_6 = -s_{23}$$

$$f_7 = -s_2$$

$$f_8 = c_3$$

$$f_9 = -c_{2233}$$

$$f_{10} = -s_{223}$$

$$f_{11} = -s_2c_2$$

$$f_{12} = -c_2s_{23}$$

$$f_{13} = -c_2$$

$$f_{14} = -s_3$$

$$f_{15} = c_2^2$$

O modelo dinâmico do manipulador na forma matricial é escrito como:

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}$$

onde os elementos das matrizes $\mathbf{H}(\mathbf{q})$ e $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ e do vetor $\mathbf{g}(\mathbf{q})$ são dados por:

$$H_{11} = k_1f_1 + k_2f_2 + k_3f_3 + k_4f_4 + k_5 + (k_{11} + k_{14})f_{15}$$

$$H_{12} = k_6f_5 + k_7f_7 + k_8f_6$$

$$H_{13} = k_6f_5 + k_8f_6$$

$$H_{21} = H_{12}$$

$$H_{22} = 2k_2 f_8 + k_{10} + k_9$$

$$H_{23} = k_2 f_8 + k_{10}$$

$$H_{31} = H_{13}$$

$$H_{32} = H_{23}$$

$$H_{33} = k_{10}$$

$$C_{11} = k_3 f_9 (\dot{q}_2 + \dot{q}_3) + \frac{1}{2} k_1 f_3 (\dot{q}_2 + \dot{q}_3) + k_2 (f_{10} \dot{q}_2 + f_{12} \dot{q}_3) + k_{11} f_{11} \dot{q}_2$$

$$C_{12} = k_3 f_9 \dot{q}_1 + \frac{1}{2} k_1 f_3 \dot{q}_1 + k_2 f_{10} \dot{q}_1 + k_{11} f_{11} \dot{q}_1 + k_8 f_5 \dot{q}_2 - k_6 f_6 (\dot{q}_2 + \dot{q}_3) \\ + k_7 f_{13} \dot{q}_2 + k_8 f_5 \dot{q}_3$$

$$C_{13} = k_3 f_9 \dot{q}_1 + \frac{1}{2} k_1 f_3 \dot{q}_1 + k_2 f_{12} \dot{q}_1 + k_8 f_5 (\dot{q}_2 + \dot{q}_3) - k_6 f_6 (\dot{q}_2 + \dot{q}_3)$$

$$C_{21} = -k_3 f_9 \dot{q}_1 - \frac{1}{2} k_1 f_3 \dot{q}_1 - k_2 f_{10} \dot{q}_1 - k_{11} f_{11} \dot{q}_1$$

$$C_{22} = k_2 f_{14} \dot{q}_3$$

$$C_{23} = k_2 f_{14} (\dot{q}_2 + \dot{q}_3)$$

$$C_{31} = -k_3 f_9 \dot{q}_1 - \frac{1}{2} k_1 f_3 \dot{q}_1 - k_2 f_{12} \dot{q}_1$$

$$C_{32} = -k_2 f_{14} \dot{q}_2$$

$$C_{33} = 0$$

$$g_1 = 0$$

$$g_2 = -k_{13} f_5 - k_{14} f_{13}$$

$$g_3 = -k_{13} f_5$$