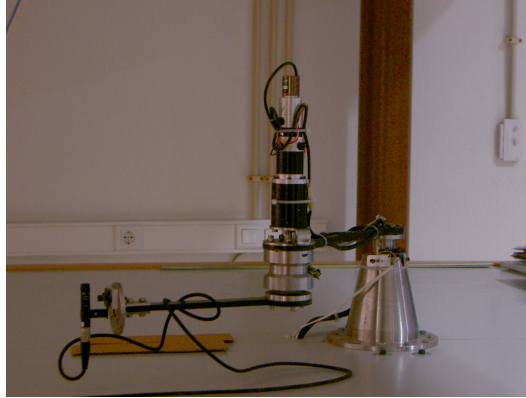




UNIVERSIDADE TÉCNICA DE LISBOA  
INSTITUTO SUPERIOR TÉCNICO



**Controlo Visual de Robôs Manipuladores**

Paulo Jorge Sequeira Gonçalves  
(Mestre)

Dissertação para obtenção do Grau de Doutor em  
Engenharia Mecânica

**Orientador:** Doutor João Rogério Caldas Pinto

**Júri:**

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutor François Chaumette

Doutor José Manuel Gutierrez Sá da Costa

Doutor João Rogério Caldas Pinto

Doutor José Alberto Rosado dos Santos Victor

Doutor João Manuel Gouveia de Figueiredo

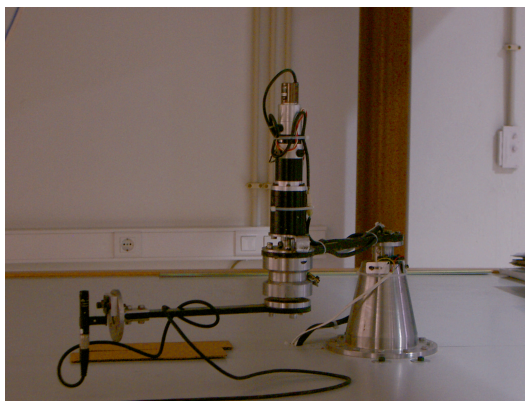
Doutor João Miguel da Costa Sousa

**Abril de 2005**





UNIVERSIDADE TÉCNICA DE LISBOA  
INSTITUTO SUPERIOR TÉCNICO



## Controlo Visual de Robôs Manipuladores

Paulo Jorge Sequeira Gonçalves  
(Mestre)

Dissertação para obtenção do Grau de Doutor em  
Engenharia Mecânica

**Orientador:** Doutor João Rogério Caldas Pinto

**Júri:**

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutor François Chaumette

Doutor José Manuel Gutierrez Sá da Costa

Doutor João Rogério Caldas Pinto

Doutor José Alberto Rosado dos Santos Victor

Doutor João Manuel Gouveia de Figueiredo

Doutor João Miguel da Costa Sousa

Abril de 2005



*À Beatriz*



# Resumo

Na presente tese é abordado o controlo visual de robôs manipuladores. Sobre o tema é apresentado o estado da arte e ainda as ferramentas de visão por computador necessárias à sua implementação. São apresentadas seis contribuições ao controlo visual de robôs manipuladores, nomeadamente o desenvolvimento de um aparato experimental, dois controladores visuais dinâmicos, a aplicação de filtros *fuzzy* ao controlo visual cinemático, a modelação *fuzzy* do sistema robô-câmara e o controlo *fuzzy* do sistema baseado no modelo inverso.

O aparato experimental desenvolvido é composto por três partes, nomeadamente um robô manipulador planar de dois graus de liberdade, um sistema de visão com 50Hz de frequência de amostragem e o *software* desenvolvido para controlar e interligar os dois componentes anteriores. O aparato experimental desenvolvido permitiu validar experimentalmente, em tempo real, os controladores propostos nesta tese.

O controlo visual dinâmico actua directamente os motores do robô, em contraste com o controlo visual cinemático que gera uma velocidade de junta a seguir pelo robô, através da utilização de um controlo interno em velocidade. A primeira contribuição ao controlo visual dinâmico é um controlador baseado na imagem, especialmente desenvolvido para o robô do aparato experimental, na configuração *eye-in-hand*. A segunda contribuição é o desenvolvimento de um controlador visual dinâmico baseado em posição para a configuração *eye-in-hand*, não estando restringido a um número fixo de graus de liberdade do robô. É ainda demonstrada a estabilidade assintótica de ambos os controladores.

A aplicação de lógica *fuzzy* ao controlo visual cinemático de robôs manipuladores baseado na imagem, revelou três contribuições. Com a aplicação de filtros *fuzzy* ao controlo visual cinemático, com planeamento de trajectórias ou em regulação, o desempenho do controlador é melhorado, i.e. as velocidades de junta do robô diminuem nos instantes iniciais e o carácter oscilatório destas é atenuado quando o tempo de amostragem de visão é elevado. Foi obtido o modelo inverso do sistema robô-câmara através de modelação *fuzzy*, tendo sido desenvolvida uma metodologia conducente à obtenção do referido modelo. O modelo inverso *fuzzy* é utilizado como controlador do sistema robô-câmara, com o objectivo de fornecer as velocidades de junta capazes de mover o robô para a posição desejada. Foi ainda utilizado um compensador *fuzzy* para compensar eventuais discrepâncias entre o modelo obtido e o sistema real.

**Palavras Chave:** Visão por Computador, Controlo Visual, Robôs Manipuladores, Filtros *Fuzzy*, Modelação *Fuzzy*, Controlo por Modelo Inverso





# Abstract

The work in thesis aims at the visual control of robotic manipulators, i.e. visual servoing. It is presented the state-of-the-art on the subject and the computer vision tools needed to its implementation. In this thesis are presented six contributions to visual servoing, namely the development of an experimental apparatus, two dynamic visual servoing controllers, the application of fuzzy filters to kinematic visual servoing, the fuzzy modeling of the robot-camera system and the fuzzy control based on the inverse model.

The experimental apparatus has three different components, namely a planar robotic manipulator with two degrees of freedom, a 50 Hz vision system and the developed software to control and inter-connect the two previous components. The developed experimental apparatus allowed the real-time experimental validation of the controllers proposed in this thesis.

The robot joint actuators are directly driven by dynamic visual servoing, in opposition to kinematic visual servoing that generates the joint velocities needed to drive the robot, by means of an inner velocity control loop. The first contribution to dynamic visual servoing is an image based control law specially developed to the robot of the experimental apparatus, with the eye-in-hand. The second contribution is a position based control law to the eye-in-hand configuration, applicable to robots with more than two degrees of freedom. For both the controllers the asymptotic stability is demonstrated.

The application of fuzzy logic to image based kinematic visual servoing, revealed three contributions. With the application of fuzzy filters to path planning and to regulator control, the overall performance of visual servoing is improved. The robot joint velocities diminish at the initial control steps and its oscillatory behavior is also diminished when the vision sample time is high. The inverse model of the robot-camera system is obtained by means of fuzzy modeling. A practical methodology for obtaining the model is also presented. The fuzzy inverse model is directly used as the controller of the robot-camera system, in order to deliver the joint velocities, needed to drive the robot to the desired position. It was also used a fuzzy compensator to compensate possible mismatches between the obtained model and the robot-camera system.

**Key-words:** Computer Vision, Visual Servoing, Robotic Manipulators, *Fuzzy* Filters, *Fuzzy* Modeling, Inverse Model Control



# Agradecimentos

A concretização da presente dissertação não teria sido possível sem o apoio de um grupo de pessoas e instituições, às quais estou sinceramente agradecido.

Em primeiro lugar agradeço ao Professor João Rogério Caldas Pinto, o meu orientador científico, pelo empenho na orientação do trabalho aqui apresentado e também pela sincera amizade demonstrada.

Ao Centro de Sistemas Inteligentes do Instituto de Engenharia Mecânica (IDMEC), na pessoa do Professor Sá da Costa, por me ter acolhido no seio do grupo de Automação e Visão, e por me ter proporcionado as condições laboratoriais necessárias ao trabalho de investigação apresentado na presente tese. Ainda no Instituto Superior Técnico, gostaria de agradecer ao Departamento de Engenharia Mecânica o apoio que me tem sido dado desde o início da década de noventa. No Grupo de Controlo, Automação e Robótica gostaria de salientar o espírito de grupo existente, o ambiente pluridisciplinar, e agradecer às pessoas que mais de perto comigo privaram: Luís Mendonça, João Sousa, Paulo Ferreira, Luís Baptista, João Reis, Jorge Martins, Sérgio Fernandes, Miguel Silva, Mário Ramalho, Carlos Cardeira.

À Escola Superior de Tecnologia do Instituto Politécnico de Castelo Branco, na pessoa do Professor Valter Lemos, todo o apoio concedido. No Departamento de Engenharia Industrial gostaria de agradecer ao Armando, ao António e ao Nuno.

Ao Fundo Social Europeu, que possibilitou a minha dispensa de serviço docente, através da acção 5.3 (formação avançada de docentes do ensino superior) do programa PRODEP III, no âmbito do III Quadro Comunitário de Apoio. À Fundação para a Ciência e Tecnologia pelo apoio através do projecto POCTI/EME/39946/2001.

Aos meus amigos de sempre: Pedro, Paula, Luís, Zé Carlos, Cristina, Jorge, Paula, por me terem aturado ao longo dos anos. Aos meus amigos mais recentes: Inês, Eduardo, Gonçalo, os momentos felizes que me têm proporcionado. Aos meus sogros, madrinha, pais e irmão todo o apoio demonstrado.

Ao Dr. Castelo-Branco Silveira, ao Dr. Eduardo Pereira, ao Dr. Miguel Areia, ao Dr. João Almeida, ao Dr. José Carlos Campos e a todo o pessoal do Hospital Amato Lusitano e Hospitais da Universidade de Coimbra, por me terem conservado no mundo dos vivos.

É agora altura para agradecer à pessoa chave durante todo o tempo que dediquei à elaboração da dissertação, a minha mulher, Beatriz. Foi ela que me apoiou na decisão de me deslocar de Castelo Branco para Lisboa, durante a dispensa de serviço docente, e tratou do nosso lar. Deu-me amor, carinho e amizade, demonstrando uma dedicação tal que eu espero ter conseguido retribuir.



# Notação

## Regras gerais

- letra minúscula: escalar
- letra minúscula a negrito: vector
- letra maiúscula a negrito: matriz
- ${}^c(\cdot)_a$ : indica que uma entidade do referencial  $a$  está expressa relativamente ao referencial  $c$ .
- ${}^a\mathbf{p}(\cdot)$ : indica que uma entidade está expresso no referencial  $a$ .

## Matemática

- $\mathbf{S}(\mathbf{p})$ : matriz anti-simétrica associada ao vector  $\mathbf{p}$ .
- $\mathbf{I}_3$ : matriz identidade ( $3 \times 3$ ).
- $\mathbf{0}_3$ : matriz de zeros ( $3 \times 3$ ).
- $\dot{\mathbf{p}}$ : derivada no tempo de uma entidade, aqui representado um vector.

## Visão por Computador

- $\mathbf{x}$ : vector no espaço euclideano
- $\tilde{\mathbf{x}}$ : vector no espaço projectivo, símbolo til por cima da letra do vector
- $\pi$ : plano da imagem
- $\mathbf{P}$ : matriz de projecção perspectiva
- $\mathbf{R}$ : matriz de rotação
- $\mathbf{u}\theta$ : Vector que expressa uma rotação de valor  $\theta$  em torno de um vector  $\mathbf{u}$ , decomposto a partir de uma matriz de rotação
- $\mathbf{t}$ : vector de translacção

## Robótica

$\mathbf{q}$  : vector das posições de junta do robô manipulador

$\mathbf{J}_R$  : matriz jacobiana do robô

$\mathbf{T}$  : Matriz de transformação entre dois referenciais

$\tau$  : vector que define os binários das juntas do robô

$\mathbf{M}$  : matriz de massas do robô

$\mathbf{C}$  : vector que define as forças centrífugas e de coriolis

$\mathbf{g}$  : vector que define as forças gravíticas

## Controlo Visual de Robôs

$(.)^*$  : indica que uma entidade se encontra na posição desejada

$\mathbf{s}$  : vector das características do objecto na imagem

$\mathbf{s}^P$  : vector das características do objecto na imagem, planeadas

$\mathbf{r}_C$  : vector da posição e orientação da câmara

$\mathbf{r}_E$  : vector da posição e orientação do elemento terminal do robô manipulador

$\mathbf{J}_i$  : matriz jacobiana da imagem

$\mathbf{x}_{3D}$  : ponto no espaço tri-dimensional

$X, Y, Z$  : coordenadas de um vector no espaço tri-dimensional

$\mathbf{v}$  : vector de velocidade no espaço cartesiano, tissor de velocidade

$\mathbf{m}$  : ponto no plano da imagem em coordenadas métricas

$\mathbf{m}_e$  : ponto no plano da imagem em coordenadas métricas, extendido

$x, y$  : coordenadas métricas de um ponto no plano da imagem

$\mathbf{p}$  : ponto no plano da imagem, em pixels

$u, v$  : coordenadas de um ponto no plano da imagem, em pixels

$\mathbf{J}_v$  : jacobiano da imagem relativo à translacção

$\mathbf{J}_{(v,w)}$  : jacobiano da imagem relativo à rotação

$\mathbf{W}$  : matriz de transformação, entre referenciais, do tissor de velocidade

$\mathbf{e}$  : vector do erro entre as características actuais e as desejadas, do objecto na imagem

$\mathbf{e}_p$  : vector do erro entre as características actuais e planeadas do objecto na imagem, entre dois instantes de tempo consecutivos

$\mathbf{e}_{pp}$  : vector do erro das características planeadas do objecto na imagem entre o instante actual e o anterior

$\lambda$  : ganho proporcional da lei de controlo visual

$\mathbf{J}_{2D}$  : matriz jacobiana para o controlo visual baseado na imagem

$\mathbf{J}_{3D}$  : matriz jacobiana para o controlo visual baseado em posição

$\mathbf{J}_{2\frac{1}{2}D}$  : matriz jacobiana para o controlo visual  $2\frac{1}{2}D$

$\Upsilon$  : pose utilizada no planeamento de trajectórias na imagem

$\mathbf{F}, \mathbf{F}_a, \mathbf{F}_r$  : forças utilizadas no planeamento de trajectórias na imagem (composta, atractiva e repulsiva)

## Abordagens *Fuzzy* ao Controlo Visual

$\mathbf{s}_f$  : vector das características do objecto na imagem, filtrado

$\mathbf{e}_{\min}$  : vector do erro mínimo das características do objecto na imagem

$\mathbf{e}_{\max}$  : vector do erro máximo das características do objecto na imagem

$\mathbf{e}_f$  : vector do erro das características do objecto na imagem, filtrado

$(\mathbf{e}_p)_f$  : vector do erro entre as características actuais e planeadas do objecto na imagem, entre dois instantes de tempo consecutivos

$(\mathbf{e}_{pp})_f$  : vector do erro das características planeadas do objecto na imagem entre o instante actual e o anterior

$\gamma$  : factor de escala *fuzzy*





# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contribuições . . . . .	3
1.2	Organização da Tese . . . . .	4
<b>2</b>	<b>Visão por Computador</b>	<b>7</b>
2.1	Geometria Projectiva . . . . .	10
2.2	Modelação da Câmara . . . . .	11
2.2.1	Projectão Perspectiva . . . . .	12
2.2.2	Projectão Perspectiva Fraca e Para-perspectiva . . . . .	13
2.3	Geometria Epipolar . . . . .	17
2.3.1	Matriz de Homografia . . . . .	17
2.3.2	Matriz Essencial e Matriz Fundamental . . . . .	21
2.4	Extracção de Características de um Objecto . . . . .	26
2.4.1	Primitivas de um Objecto na Imagem . . . . .	26
2.4.2	Cálculo da Pose de um Objecto . . . . .	27
2.5	Correspondências de Características em Imagens Sucessivas . . . . .	33
<b>3</b>	<b>Controlo Visual de Robôs Manipuladores</b>	<b>35</b>
3.1	Introdução . . . . .	35
3.2	Informação Visual para Controlo . . . . .	38

3.2.1	Visão Monocular . . . . .	39
3.2.2	Visão Estéreo . . . . .	41
3.2.3	Sistema com Câmaras Redundantes . . . . .	42
3.3	Controlo Visual Cinemático . . . . .	44
3.3.1	Controlo Visual Cinemático baseado na imagem . . . . .	45
3.3.2	Controlo Visual Cinemático baseado em posição . . . . .	51
3.3.3	Controlo Visual Cinemático Híbrido . . . . .	55
3.4	Controlo Visual Dinâmico . . . . .	57
3.5	Controlo Visual por estimação do modelo de interacção . . . . .	60
3.6	Planeamento de Trajectórias para Controlo Visual . . . . .	63
<b>4</b>	<b>Controlo Visual Dinâmico baseado em Posição</b>	<b>67</b>
4.1	Introdução . . . . .	67
4.2	Definição do Problema de Controlo . . . . .	68
4.3	Lei de Controlo Visual Dinâmica . . . . .	69
4.3.1	Extensão do Controlador Dinâmico - Kelly . . . . .	69
4.3.2	Controlo Visual Dinâmico, <i>eye-in-hand</i> , baseado em posição . . . . .	70
4.4	Análise de Estabilidade da Lei de Controlo . . . . .	72
<b>5</b>	<b>Abordagens Fuzzy ao Controlo Visual</b>	<b>75</b>
5.1	Introdução . . . . .	75
5.2	Filtros Fuzzy . . . . .	77
5.2.1	Filtros Fuzzy aplicados ao Controlo Visual - Regulador . . . . .	77
5.2.2	Filtros Fuzzy aplicados ao Controlo Visual - Trajectória Planeada . . . . .	81
5.3	Modelação <i>Fuzzy</i> . . . . .	81
5.3.1	Identificação do Modelo Inverso <i>Fuzzy</i> . . . . .	85
5.3.2	Identificação do Modelo Inverso Fuzzy do Aparato Experimental . . . . .	86
5.4	Controlo baseado no modelo inverso <i>Fuzzy</i> . . . . .	91
5.4.1	Compensador <i>Fuzzy</i> . . . . .	93

<b>6</b>	<b>Resultados Experimentais em Controlo Visual</b>	<b>97</b>
6.1	Introdução . . . . .	97
6.2	Condições Experimentais . . . . .	98
6.3	Filtros Fuzzy . . . . .	103
6.3.1	Aplicação ao caso do Regulador . . . . .	103
6.3.2	Aplicação ao caso do Planeamento de Trajectórias . . . . .	110
6.3.3	Conclusões da Aplicação de Filtros <i>Fuzzy</i> . . . . .	113
6.4	Modelação Fuzzy . . . . .	114
6.5	Controlo Fuzzy . . . . .	118
6.6	Controlo Visual Dinâmico . . . . .	122
6.6.1	Extensão do Controlador Dinâmico - Kelly . . . . .	123
6.6.2	Controlo Visual Dinâmico, <i>eye-in-hand</i> , baseado em posição . . . . .	124
6.6.3	Conclusões sobre o Controlo Visual Dinâmico . . . . .	126
6.7	Abordagens Clássicas . . . . .	127
6.7.1	Controlo Visual Cinemático Baseado na Imagem . . . . .	127
6.7.2	Controlo Visual Cinemático Híbrido . . . . .	128
<b>7</b>	<b>Conclusões</b>	<b>135</b>
7.1	Sumário e Contribuições . . . . .	135
7.2	Discussão . . . . .	137
7.3	Trabalho Futuro . . . . .	139
<b>A</b>	<b>Modelação de Robôs Manipuladores</b>	<b>157</b>
A.1	Movimento do Corpo Rígido . . . . .	157
A.1.1	Mudança de Coordenadas . . . . .	157
A.1.2	Torsor de Velocidade . . . . .	159
A.2	Cinemática de Robôs Manipuladores . . . . .	160

A.2.1	Cinemática Directa de Robôs Manipuladores . . . . .	161
A.2.2	Cinemática Inversa de Robôs Manipuladores . . . . .	162
A.2.3	Cinemática Diferencial de Robôs Manipuladores . . . . .	163
A.3	Dinâmica de Robôs Manipuladores . . . . .	164
A.4	Controlo em Velocidade de Robôs Manipuladores . . . . .	166
<b>B</b>	<b>Aparato Experimental</b>	<b>167</b>
B.1	Sistema de Visão . . . . .	167
B.1.1	Hardware . . . . .	167
B.1.2	Software . . . . .	170
B.2	Sistema Robótico . . . . .	173
B.3	Interligação dos Sistemas . . . . .	173

# Lista de Figuras

2.1	Resumo dos procedimentos de Visão por Computador utilizados na Tese. . . . .	8
2.2	Formação da imagem com o modelo da câmara pontual. . . . .	12
2.3	Representação genérica da projecção perspectiva de pontos no referencial do objecto, directamente no plano da imagem. . . . .	14
2.4	A geometria epipolar. . . . .	18
3.1	Visão monocular com a câmara colocada no elemento terminal e a olhar para o objecto, <i>eye-in-hand</i> . . . . .	40
3.2	Visão monocular com a câmara a olhar para o objecto e para o robô manipulador, <i>eye-to-hand</i> . . . . .	41
3.3	Visão estéreo com as câmaras colocadas no elemento terminal e a olhar para o objecto. . . . .	42
3.4	Visão estéreo com as câmaras a olhar para o objecto e para o robô manipulador. . . . .	42
3.5	Controlo Visual Cinemático. . . . .	44
3.6	Controlo Visual Cinemático baseado na imagem. . . . .	46
3.7	Controlo Visual Cinemático baseado em posição. . . . .	52
3.8	Características do objecto na imagem para controlo visual baseado em posição. . . . .	53
3.9	Controlo Visual Dinâmico. . . . .	59
3.10	Controlo Visual Dinâmico baseado na imagem. . . . .	59
3.11	Controlo Visual Dinâmico baseado em posição. . . . .	60

5.1	Função de pertença triangular. . . . .	79
5.2	Configuração do sistema robô-câmara para identificação do modelo inverso. . . . .	89
5.3	Entradas e saídas do modelo inverso a ser identificado. . . . .	91
5.4	O compensador <i>fuzzy</i> inserido no controlo por modelo inverso. . . . .	94
5.5	Definição da fronteira de <i>SS</i> , onde a compensação está activa. . . . .	95
6.1	Imagem correspondente à posição 1. . . . .	99
6.2	Imagem correspondente à posição 2. . . . .	100
6.3	Imagem correspondente à posição 3. . . . .	100
6.4	Fotografia de conjunto do robô manipulador, com a câmara acoplada, e o objecto (alvo) planar. . . . .	101
6.5	Comparação do tursor de velocidade da câmara durante o controlo visual, com e sem a utilização de filtros <i>fuzzy</i> . Cima - velocidades de translação. Baixo - velocidades de rotação. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	105
6.6	Comparação do erro das características do objecto na imagem, durante o controlo visual. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	106
6.7	Comparação das trajectórias das características do objecto na imagem, durante o controlo visual. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . (+) posição desejada. . . . .	107
6.8	Comparação das velocidades de junta durante o controlo visual, com e sem a utilização de filtros <i>fuzzy</i> . Tempo de amostragem de visão 20[ms]. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . Cima - $\varepsilon$ pequeno. Baixo - $\varepsilon$ grande. . . . .	108
6.9	Comparação das velocidades de junta durante o controlo visual, com e sem a utilização de filtros <i>fuzzy</i> . Tempo de amostragem de visão 100[ms]. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	109
6.10	Comparação do erro das características do objecto na imagem, com e sem a utilização de filtros <i>fuzzy</i> . Tempo de amostragem de visão 20[ms]. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . Cima - $\varepsilon$ pequeno. Baixo - $\varepsilon$ grande. . . . .	109
6.11	Comparação das trajectórias das características do objecto na imagem. Tempo de amostragem vídeo 20[ms]. Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . (+) posição desejada. . . . .	110

6.12	Comparação das velocidades de junta durante o controlo visual com trajectória planeada, com e sem a utilização de filtros <i>fuzzy</i> . Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	111
6.13	Comparação do erro das características do objecto na imagem, entre a posição actual e a trajectória planeada, com e sem a utilização de filtros <i>fuzzy</i> . Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	112
6.14	Comparação do erro das características do objecto na imagem, entre a posição actual e a desejada, com e sem a utilização de filtros <i>fuzzy</i> . Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . . . . .	112
6.15	Comparação entre a trajectória planeada e as trajectórias com e sem a utilização de filtros <i>fuzzy</i> , com e sem a utilização de filtros <i>fuzzy</i> . Cheio - com filtros <i>fuzzy</i> . Traço-ponto - sem filtros <i>fuzzy</i> . Traço interrompido - trajectória planeada . . . . .	113
6.16	Dados de entrada/saída para a identificação do modelo inverso. Cima: posições de junta. Baixo: variação das características do objecto na imagem.	115
6.17	Validação do modelo inverso <i>fuzzy</i> (velocidades de junta). Cheio – valor real do dado de saída. Traço-ponto – saída do modelo inverso <i>fuzzy</i> . . . . .	116
6.18	Funções de pertença $A_{ij}$ obtidas para a saída $\delta q_1$ . . . . .	117
6.19	Membership functions $A_{ij}$ obtained for the output $\delta q_2$ . . . . .	119
6.20	Erro das características do objecto na imagem, $\mathbf{e}$ , durante o controlo. Cheio – controlo <i>fuzzy</i> . Traço-ponto – controlo visual cinemático baseado na imagem. . . . .	120
6.21	Velocidade das juntas do robô, $\dot{\mathbf{q}}$ , durante o controlo. Cheio – controlo <i>fuzzy</i> . Traço-ponto – controlo visual cinemático baseado na imagem. . . . .	120
6.22	Trajectória das características do objecto na imagem $\mathbf{s}$ , no plano da imagem. Cheio – controlo <i>fuzzy</i> . Traço-ponto – controlo visual cinemático baseado na imagem. . . . .	121
6.23	Erro das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com a extensão do controlador dinâmico - Kelly. . . . .	123
6.24	Trajectória das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com a extensão do controlador dinâmico - Kelly. '+' posição final. . . . .	124
6.25	Erro das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com o controlador visual dinâmico baseado em posição. . . . .	125
6.26	Trajectória dos pontos necessários à reconstrução da pose, com o controlador visual dinâmico baseado em posição. '+' posição final. . . . .	126

6.27	Erro das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com o controlador visual baseado na imagem. . . . .	129
6.28	Trajectória das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com o controlador visual baseado na imagem. '+' posição final. . . .	129
6.29	Velocidades das juntas dos robô, com o controlador visual baseado na imagem.	130
6.30	Erro das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com o controlador visual híbrido ( $2\frac{1}{2}D$ ). . . . .	132
6.31	Trajectória das características do objecto na imagem $\mathbf{s}$ , no plano da imagem, com o controlador visual híbrido ( $2\frac{1}{2}D$ ). '+' posição final. . . . .	132
6.32	Velocidades das juntas dos robô, com o controlador visual híbrido ( $2\frac{1}{2}D$ ). . .	133
A.1	Matriz de transformação entre dois referenciais . . . . .	159
A.2	Controlo em velocidade do robô planar de dois graus de liberdade . . . . .	166
B.1	Fotografia de conjunto do aparato experimental. . . . .	168
B.2	Computador de Visão. . . . .	169
B.3	Correspondências entre imagens sucessivas. '+', centróides na imagem anterior. '•', centróides na imagem actual. Traço interrompido, janela de procura. . . . .	172
B.4	Robô planar com a câmara acoplada ao elemento terminal. . . . .	174
B.5	Vista dos amplificadores de potência, fonte de alimentação e placa de ligações entre o computador, amplificadores de potência e actuadores. . . .	175
B.6	Interligação dos Sistemas de Visão e Robótico. . . . .	176
B.7	Implementação da lei de controlo visual em <i>MatLab Simulink xPC</i> . . . . .	177
B.8	Implementação do cálculo do jacobiano global para a lei de controlo visual em <i>MatLab Simulink xPC</i> . . . . .	178
B.9	Implementação da lei de controlo em velocidade em <i>MatLab Simulink xPC</i>	179



# Capítulo 1

## Introdução

O objectivo do presente trabalho é o desenvolvimento de técnicas para o controlo visual de robôs manipuladores. Robôs manipuladores são sistemas dinâmicos de elevada complexidade que podem utilizar vários tipos de sensores para reconhecer o ambiente de trabalho em que se inserem. Nomeadamente, sensores de ultrasons e laser para medir distâncias, sensores de força para reconhecer o contacto com o ambiente de trabalho e câmaras para visualizar o ambiente de trabalho na sua globalidade. Actualmente, na indústria, os robôs manipuladores são utilizados em ambientes de trabalho especificamente projectados para a sua operação. A utilização de câmaras poderá, num futuro que esperamos próximo, tornar os robôs completamente autónomos para operar em meios desconhecidos ou dificilmente modeláveis.

A utilização da visão por computador em tempo-real para o controlo de robôs tem estado constringida, devido aos elevados requisitos de cálculo. De facto, no caso mais geral, para a aplicação de algoritmos de visão por computador é necessária uma fase para o processamento de imagem, geralmente dispendiosa em tempo de cálculo. No final da década de setenta, data do primeiro sistema de controlo visual, o tempo necessário à extracção de características do objecto da imagem era na ordem de dezenas de segundos. A visão era utilizada para obter a posição desejada do objecto, sendo o controlo efectuado em anel aberto em termos do sensor de visão. Desde o início dos anos noventa, com o

aumento da capacidade de cálculo dos computadores pessoais, é possível fechar o controlo com o sensor de visão e torná-lo cada vez mais "tempo-real", i.e. o tempo de amostragem de visão tem vindo a diminuir, sendo possível encontrar aplicação com 1[ms]. Este tipo de controlo, em que se fecha o anel com o sensor de visão, é designado por controlo visual e tem como objectivo anular o erro entre as características do objecto na posição desejada e as calculadas durante o controlo.

A utilização de controlo visual em robôs colocados em ambientes industriais, desconhecidos ou dificilmente modeláveis requer uma elevada precisão, portabilidade, fiabilidade e facilidade de colocação em funcionamento. Para que os aspectos práticos referidos possam ser atingidos, é necessário que vários objectivos teóricos do controlo visual sejam atingidos:

- a obtenção de uma elevada precisão no controlo visual está associada à qualidade da extracção das informações visuais do objecto no meio em que se encontra. Assim, pode ser necessário utilizar, em vez da visão monocular, visão stereo ou ainda aumentar o número de câmaras a observar o objecto. A utilização em conjunto com visão de outros tipos de sensores, como os já referidos ultrasons, lasers ou sensores de força, permite aumentar o conhecimento do meio em que o robô se movimenta e também aumentar a precisão nas medidas necessárias ao controlo.
- a portabilidade de um sistema de controlo visual de robôs está associada ao conhecimento que a lei de controlo tem de ter do meio que rodeia o robô e dos objectos observados, passíveis de serem manipulados. Assim para que o sistema de controlo seja o mais portátil possível, deverá requerer um conhecimento mínimo do modelo do objecto e do meio.
- a fiabilidade de um sistema de controlo visual, à parte da fiabilidade dos equipamentos mecânicos e eléctricos que constituem o sistema, está associada à robustez na variação dos parâmetros de calibração e à convergência a partir de uma qualquer posição inicial.

- a facilidade de colocação em funcionamento de um sistema de controlo visual está dependente da sua fase de calibração, i.e. esta fase deverá ser simplificada o mais possível e no caso ideal eliminá-la.

As leis de controlo visual existentes na literatura ainda não são capazes de abarcar os quatro aspectos enumerados anteriormente, apenas o controlo visual híbrido (no que se refere à portabilidade e à fiabilidade) e o controlo baseado na estimação do modelo de interacção robô-câmara (no que se refere à portabilidade, fiabilidade e facilidade de colocação em funcionamento) se aproximam do requerido. A precisão do controlo visual pode ser aumentada como visto anteriormente, independentemente da abordagem ao controlo utilizada.

Na secção seguinte são apresentadas de forma sucinta as contribuições propostas nesta tese para aumentar a fiabilidade, a portabilidade e a facilidade de colocação em funcionamento de um sistema de controlo visual.

## 1.1 Contribuições

Ao controlo visual dinâmico são propostas duas abordagens diferentes, a primeira baseada em características do objecto na imagem e a segunda baseada na extracção da pose do objecto através da imagem. Para ambos os controladores foi demonstrada a estabilidade assintótica na convergência para a posição desejada. Estas contribuições asseguram a fiabilidade do sistema de controlo.

A estimação do modelo inverso da interacção robô-câmara foi abordada na presente tese através de modelação *fuzzy*. Com este tipo de modelo obvia-se a calibração do sistema e assim aumenta-se a facilidade de colocação em funcionamento. O modelo inverso obtido é utilizado como controlador do sistema robô-câmara, com o objectivo de fornecer as velocidades de junta capazes de mover o robô para a posição desejada. Foi ainda utilizado

um compensador *fuzzy* para compensar eventuais discrepâncias entre o modelo obtido e o sistema real.

Com o objectivo de suavizar as velocidades geradas no controlo visual cinemático, foram aplicados filtros *fuzzy* ao erro das características do objecto na imagem. Assim, é diminuída a amplitude das velocidades nos instantes iniciais e ainda atenuado o seu carácter oscilatório, no caso do tempo de amostragem de visão ser elevado. Esta abordagem tem a vantagem de ter apenas um parâmetro para ajustar, em contraste com as abordagens existentes na literatura. Consegue-se assim aumentar a fiabilidade e a facilidade de colocação em funcionamento do sistema de controlo.

As leis de controlo visual propostas na presente tese, assim como os filtros *fuzzy*, foram verificados experimentalmente no aparato experimental desenvolvido especificamente para esta tese.

## 1.2 Organização da Tese

O capítulo dois introduz os conceitos e técnicas de visão por computador utilizadas em controlo visual de robôs de manipuladores, em especial as utilizadas na presente tese: geometria projectiva, modelação de câmaras, geometria epipolar, estimação da pose e correspondências entre imagens sucessivas.

O capítulo três apresenta o estado da arte do controlo visual de robôs manipuladores. São apresentados os controladores visuais cinemáticos baseados na imagem, em posição e híbridos. São ainda abordados o controlo visual dinâmico, a estimação do modelo de interacção robô-câmara e o planeamento de trajectórias na imagem.

No capítulo quatro são propostas as contribuições ao controlo visual dinâmico, a extensão do controlador Kelly e o controlo visual dinâmico baseado em posição, com estimação da pose. A estabilidade assintótica dos controladores é também demonstrada.

No capítulo cinco são propostas soluções para o controlo visual de robôs manipuladores, com base em lógica *fuzzy*. São aplicados filtros *fuzzy* para melhorar o desempenho da lei de controlo visual cinemática. O modelo inverso da interacção entre robô e câmara é estimado através de modelação *fuzzy*, sendo posteriormente utilizado como controlador do sistema, em conjunto com um compensador *fuzzy*.

No capítulo seis são apresentadas as condições experimentais e os resultados obtidos no aparato experimental, descrito no anexo B, para a implementação dos filtros e controladores apresentados nos capítulos anteriores.

Finalmente, o capítulo sete resume as contribuições propostas na tese e apresenta reflexões e possíveis linhas de trabalho futuro.



## Capítulo 2

# Visão por Computador

Nas actuais aplicações de robôs na indústria, estes movimentam-se em ambientes estruturados, i.e. em ambientes de trabalho especialmente projectados para a realização das várias tarefas para que são programados. Porém, o mundo em que vivemos e em que o robô se encontra inserido tem características dinâmicas, quer através do seu próprio movimento, quer através do movimento de objectos no ambiente de trabalho do robô, ou ainda de outros agentes aí existentes a realizar as suas tarefas [39], i.e. ambientes não-estruturados. É considerando este novo facto que é imperioso, por forma a que o robô possa interagir de uma forma harmoniosa com o meio que o rodeia, muni-lo com sensores capazes de adquirir mais informação que os clássicos encoders e tacómetros, colocados nas juntas do robô. O sensor que melhor reúne as características necessárias ao adequado desempenho em ambientes não-estruturados é o de visão. Tal escolha é justificada pelos muitos sistemas biológicos que o utilizam para reunir informação sobre o ambiente que os rodeia, pelo incremento das suas características e crescente disponibilidade no mercado a cada vez mais baixo custo [39], podendo ainda ser facilmente acoplados a computadores. Assim, através do sensor de visão, a câmara, é possível extrair do ambiente de trabalho a informação visual necessária à realização de tarefas por parte dos robôs. Para tal, são apresentadas no presente capítulo os fundamentos teóricos de visão por computador que serão utilizados no controlo visual de robôs, no presente caso, manipuladores.

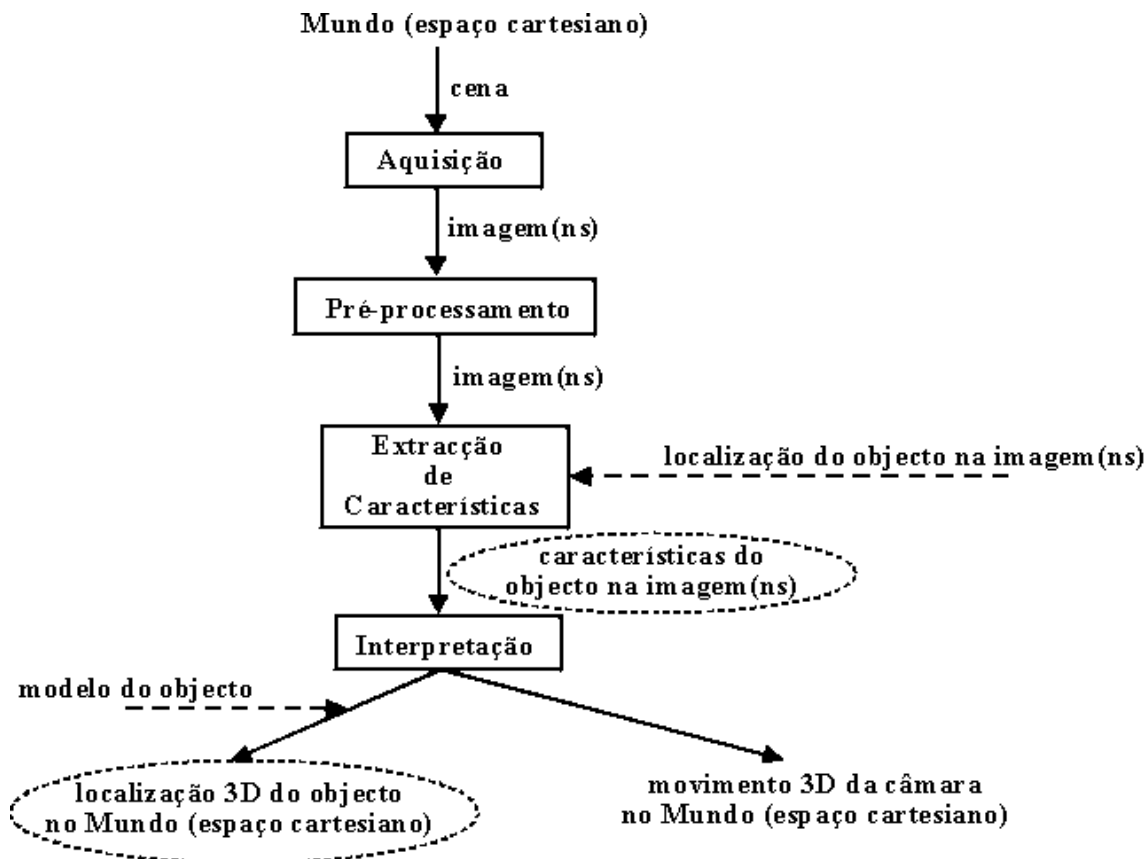


Figura 2.1: Resumo dos procedimentos de Visão por Computador utilizados na Tese.

Como introdução ao capítulo, na figura 2.1 encontram-se esquematizadas todas as operações conducentes à extracção de informação de uma imagem ou sequência de imagens, e que serve de base para o que se apresentará nas secções seguintes. Considerando todo o espaço cartesiano, i.e. o mundo, e uma ou mais câmaras a olhar para uma determinada cena no mundo, o primeiro passo para um sistema de visão por computador é a *aquisição* [131] da imagem ou imagens. Como segundo passo é necessário pré-processar [131] a(s) imagem(ns) e assim realizar a preparação para os procedimentos seguintes. Note-se que é nesta fase que se retira ou atenua o ruído da imagem adquirida, sendo esta fase importante pois todos os sistemas de visão por computador se baseiam no processamento dos *pixels* da imagem. As operações de aquisição e pré-processamento da imagem referidas não serão abordadas no presente capítulo, mas sim no anexo B que diz respeito ao aparato experimental.



A construção de um sistema de visão por computador tem como objectivo principal extrair informação de uma ou várias imagens sobre a mesma cena. Esta informação visual a extrair de uma imagem depende dos *pixels* nela existentes e ainda do ruído associado, que deverá ser atenuado. Numa imagem podem co-existir vários objectos sendo que só um deles é, geralmente, de interesse para a tarefa a realizar pelo robô, logo é necessário localizar este último na imagem. Existem na literatura vários algoritmos para localizar um determinado objecto na imagem [131], sendo que essa informação pode ser conhecida a priori como é o caso das experiências apresentadas no capítulo 6, em que só existe um objecto na cena observada. Após o objecto estar localizado na imagem é necessário extrair as suas características, i.e. as características do objecto na imagem, sobre as quais se procederá à fase de interpretação da imagem. Estas características podem ser pontos, curvas ou estruturas particulares de níveis de cinzento, entre outras explicitadas em [39, 68, 115, 119, 131]. De referir que as características do objecto na imagem, evidenciadas numa elipse a tracejado na figura 2.1 são a base do controlo visual baseado na imagem, definido na secção 3.3.1. Como referenciado na literatura, por exemplo [131], a captura de uma imagem envolve o mapeamento de coordenadas tri-dimensionais do mundo em coordenadas bi-dimensionais da imagem, com a consequente perda de informação. O controlo visual de robôs manipuladores baseado em posição assim como o controlo visual híbrido, definido nas secções 3.3.2 e 3.3.3 respectivamente, tem por base a extracção de características tri-dimensionais do objecto ou do movimento da câmara. Estas características são definidas como a localização tri-dimensional do objecto no mundo, i.e. a sua pose relativamente à câmara (cuja estimação necessita do conhecimento prévio do modelo do objecto), e a rotação e translação da câmara durante o seu movimento (movimento tri-dimensional da câmara no mundo).

No presente capítulo irão ser apresentadas as ferramentas de visão por computador, necessárias ao desenvolvimento e aplicação das técnicas de controlo visual de robôs manipuladores apresentadas na presente tese. Assim na secção 2.1 são apresentados os fundamentos

teóricos de geometria projectiva. Em seguida, na secção 2.2 é dada ênfase à modelação de câmaras, utilizando a transformação perspectiva total, a transformação perspectiva fraca e a transformação para-perspectiva. A geometria epipolar é apresentada na secção 2.3, relativamente a transformações que se referem a objectos no plano (homografias) e a transformações genéricas (matriz essencial e fundamental). Na secção 2.4 são apresentados os algoritmos utilizados para extrair as primitivas do objecto na imagem e para o cálculo da pose, ambas utilizadas na presente tese. É ainda apresentado, na secção 2.5, o algoritmo utilizado para o seguimento das características do objecto durante o controlo visual.

## 2.1 Geometria Projectiva

A presente secção apresenta os fundamentos teóricos de geometria projectiva que serão usados nos capítulos e secções seguintes. A teoria subjacente à geometria projectiva está perfeitamente consolidada na comunidade científica, como descrito em [39, 59, 131].

A presente secção tem início com a descrição das propriedades básicas da geometria projectiva, tais como as noções de espaço projectivo e transformação projectiva, ou colineação. A geometria projectiva será também utilizada para obter o modelo perspectivo da câmara.

**Definição 1 (Espaço Afim e Espaço Projectivo)** *O conjunto de pontos parametrizado por um conjunto de vectores, de dimensão  $n$   $(x_1, \dots, x_n)^T \in \mathbb{R}^n$  é denominado Espaço Afim (Euclideano). O conjunto de pontos representados por um vector de dimensão  $n + 1$   $(x_1, \dots, x_n, x_{n+1})^T$  é denominado Espaço Projectivo  $\mathbb{P}^n$  se a seguinte condição e propriedade forem consideradas:*

1. *Pelo menos uma das  $n + 1$  coordenadas do vector deve ser diferente de zero.*
2. *Os dois vectores  $(x_1, \dots, x_n, x_{n+1})^T$  e  $(\lambda x_1, \dots, \lambda x_n, \lambda x_{n+1})^T$ , representam o mesmo ponto para qualquer  $\lambda \neq 0$ .*

Os elementos  $x_i$ , ( $i = 1, \dots, n + 1$ ) de um espaço projectivo são usualmente denominados

de *coordenadas homogêneas* ou *coordenadas projectivas*. O espaço afim  $\mathbb{R}^n$  pode ser considerado como estando contido no espaço projectivo  $\mathbb{P}^n$  utilizando o seguinte mapeamento injectivo canónico  $(x_1, \dots, x_n)^T \longrightarrow (x_1, \dots, x_n, 1)^T$ . Inversamente, é possível recuperar as coordenadas afins de um ponto a partir das suas coordenadas homogêneas utilizando o seguinte mapeamento:

$$\left( x_1, \dots, x_{n+1} \right)^T \doteq \left( \frac{x_1}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}}, 1 \right)^T \longrightarrow \left( \frac{x_1}{x_{n+1}}, \dots, \frac{x_n}{x_{n+1}} \right)^T \quad (2.1)$$

em que  $\doteq$  denota uma igualdade a menos de um escalar, propriedade das coordenadas homogêneas. Se a última coordenada de um ponto  $\tilde{\mathbf{x}} \in \mathbb{P}^n$  é nula, i.e.  $x_{n+1} = 0$  então  $\tilde{\mathbf{x}}$  é denominado um ponto no infinito.

**Definição 2 (Colienação)** *Uma transformação linear ou colienação num espaço projectivo  $\mathbb{P}^n$  é definida por uma matriz  $\mathbf{A}$ , não-singular e de dimensão  $(n+1) \times (n+1)$ .*

A matriz  $A$  caracteriza-se por mapeamento invertível de  $\mathbb{P}^n$  nele próprio e é definida a menos de um escalar diferente de zero. Usualmente é representada por  $\lambda \tilde{\mathbf{y}} = \mathbf{A} \tilde{\mathbf{x}}$ .

## 2.2 Modelação da Câmara

O modelo mais usado em visão por computador para modelar uma câmara é o *modelo de câmara pontual*, *pinhole camera model*, como representado na figura 2.2. É um modelo simples e eficaz para modelar a grande maioria de câmaras existentes no mercado. Considera que os raios de luz emitidos ou reflectidos por um dado objecto passam todos por um "furo", *pinhole*, sendo projectados numa superfície plana, i.e. *plano da imagem*.

O modelo de câmara pontual consiste num plano  $\pi$ , o *plano da imagem*, e um ponto no espaço euclideano  $\mathbf{c}$ , o *centro da projecção*. A distância entre o plano da imagem e o centro de projecção é denominada como *distância focal*,  $f$ . A linha que passa pelo centro de projecção e é perpendicular ao plano da imagem é denominada como *eixo óptico*. A

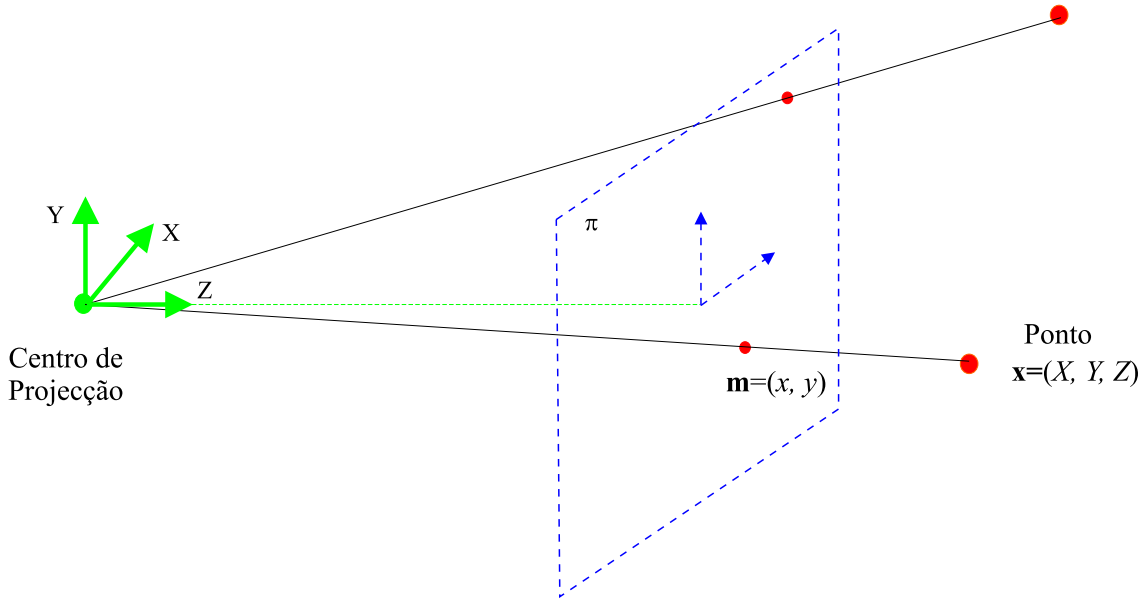


Figura 2.2: Formação da imagem com o modelo da câmara pontual.

intersecção entre o eixo óptico e o plano da imagem é um ponto denominado o *centro da imagem*. O *referencial ortonormado da câmara* é ainda definido através do centro de projecção e pelo eixo óptico que coincide com o eixo Z do referido referencial. A utilização da geometria projectiva permite que o modelo de projecção perspectiva seja descrito por uma relação linear, tornando-o mais simples de trabalhar. Assim, uma câmara pode ser descrita através do mapeamento projectivo linear entre o espaço projectivo  $\mathbb{P}^3$  e  $\mathbb{P}^2$ .

### 2.2.1 Projecção Perspectiva

O modelo da câmara pontual associado a uma projecção perspectiva, mapeia um ponto genérico no espaço euclideano, aqui expresso em coordenadas homogéneas no referencial do mundo "0",  ${}^0\tilde{\mathbf{x}} = (X \ Y \ Z \ 1)^T$ , num ponto na imagem em coordenadas métricas  $\tilde{\mathbf{m}} = (x, y, 1)^T$  e em coordenadas da imagem [pixel]  $\tilde{\mathbf{p}} = (u, v, 1)^T$ :

$$\lambda\tilde{\mathbf{m}} = [{}^c\mathbf{R}_0 \quad {}^c\mathbf{t}_0] \cdot {}^0\tilde{\mathbf{x}} \quad (2.2)$$

$$\tilde{\mathbf{p}} = \mathbf{K} \cdot \lambda\tilde{\mathbf{m}} \quad (2.3)$$

Note-se que a equação (2.2) e a definição de coordenadas métricas na imagem se refere a uma câmara normalizada, i.e. com distância focal unitária.

A matriz  ${}^c\mathbf{R}_0$  de dimensão  $(3 \times 3)$  e o vector  ${}^c\mathbf{t}_0$  de dimensão  $(3 \times 1)$  descrevem respectivamente, a rotação e a translação entre os referenciais do mundo e da câmara. Genericamente:

$$\mathbf{R} = \begin{bmatrix} \mathbf{i}^T \\ \mathbf{j}^T \\ \mathbf{k}^T \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (2.4)$$

A matriz  $\mathbf{K}$  de dimensão  $(3 \times 3)$  depende dos parâmetros internos da câmara, i.e. os *parâmetros intrínsecos*:

$$\mathbf{K} = \begin{bmatrix} f \cdot k_u & f \cdot k_\theta & u_0 \\ 0 & f \cdot k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Os parâmetros  $k_u$  e  $k_v$  definem factores de escala ao longo das direcções horizontal e vertical na imagem. Adicionalmente o parâmetro  $k_\theta$  define a distorção entre as mesmas direcções. Usualmente a origem das coordenadas da imagem não se encontra no centro da imagem, mas sim no canto superior esquerdo. Assim, é necessário efectuar a respectiva translação, através das coordenadas do centro da imagem,  $u_0$  e  $v_0$ . A distância focal  $f$  foi já definida anteriormente. Define-se ainda a matriz de projecção perspectiva total  $\mathbf{P}$ :

$$\mathbf{P} = \mathbf{K} \cdot \begin{bmatrix} {}^c\mathbf{R}_0 & {}^c\mathbf{t}_0 \end{bmatrix} \quad (2.6)$$

### 2.2.2 Projecção Perspectiva Fraca e Para-perspectiva

Nesta sub-secção é apresentado um modelo simplificado da projecção perspectiva total. Este modelo é uma boa aproximação da projecção perspectiva total quando o

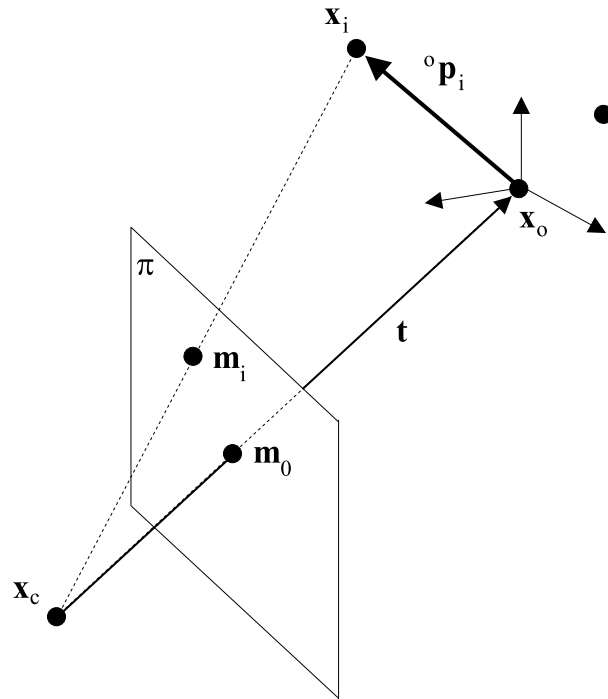


Figura 2.3: Representação genérica da projecção perspectiva de pontos no referencial do objecto, directamente no plano da imagem.

ponto/objecto se encontra na vizinhança do eixo óptico da câmara. Este modelo será utilizado na sub-secção 2.4.2.

Considerando um objecto no espaço euclideo ao qual está associado um referencial ortonormado "o" com origem em  $\mathbf{x}_o$ , e seja  $\mathbf{x}_i$  um ponto desse objecto, ver figura 2.3, o correspondente ponto na imagem em coordenadas métricas pode ser determinado pelas equações:

$$x_i = \frac{\mathbf{i} \cdot {}^o\mathbf{p}_i + t_x}{\mathbf{k} \cdot {}^o\mathbf{p}_i + t_z} \quad (2.7)$$

$$y_i = \frac{\mathbf{j} \cdot {}^o\mathbf{p}_i + t_y}{\mathbf{k} \cdot {}^o\mathbf{p}_i + t_z} \quad (2.8)$$

em que os vectores  $\mathbf{i}, \mathbf{j}, \mathbf{k}$  definem a rotação entre o referencial da câmara (c) e o do objecto (o), e os escalares  $t_x, t_y, t_z$  definem o respectivo vector de translação.

Dividindo o numerador e o denominador das duas equações anteriores por  $t_z$  e introdu-

zindo as seguintes notações:

- $\mathbf{i}_s = \mathbf{i}/t_z$  é a primeira linha da matriz de rotação escalada com a terceira componente do vector de translação;
- $\mathbf{j}_s = \mathbf{j}/t_z$  é a segunda linha da matriz de rotação escalada com a terceira componente do vector de translação;
- $x_o = t_x/t_z$  e  $y_o = t_y/t_z$  são as coordenadas métricas do ponto  $\mathbf{p}_o$  na imagem;
- $\varepsilon_i = \frac{\mathbf{k} \cdot \mathbf{p}_i}{t_z}$ .

as equações (2.7) (2.8) podem ser reescritas da seguinte forma:

$$x_i = \frac{\mathbf{i}_s \cdot \mathbf{p}_i + x_o}{1 + \varepsilon_i} \quad (2.9)$$

$$y_i = \frac{\mathbf{j}_s \cdot \mathbf{p}_i + y_o}{1 + \varepsilon_i} \quad (2.10)$$

ou ainda:

$$x_i(1 + \varepsilon_i) - x_o = \mathbf{i}_s \cdot \mathbf{p}_i \quad (2.11)$$

$$y_i(1 + \varepsilon_i) - y_o = \mathbf{j}_s \cdot \mathbf{p}_i \quad (2.12)$$

A partir das equações anteriores é possível proceder a aproximações no modelo perspectivo global, i.e. obter um modelo perspectivo fraco (utilizado no algoritmo de Dementhon [33]) e um modelo paraperspectivo (utilizado em [67]).

### Perspectiva Fraca

O modelo perspectivo fraco assume que os pontos que caracterizam o objecto se encontram num plano paralelo ao plano da imagem e que passa pelo eixo óptico da câmara. Esta aproximação é equivalente a:

$$\frac{1}{1 + \varepsilon_i} \approx 1 \quad (2.13)$$

Com esta aproximação, as equações 2.9 e 2.10 podem ser reescritas:

$$(x_i)_f - x_o = \mathbf{i}_s \cdot {}^o\mathbf{p}_i \quad (2.14)$$

$$(y_i)_f - y_o = \mathbf{j}_s \cdot {}^o\mathbf{p}_i \quad (2.15)$$

em que  $(x_i)_f$  e  $(y_i)_f$  significam os valores aproximados pela transformação perspectiva fraca das coordenadas métricas do ponto  $\mathbf{x}_i$ , na imagem. Através das equações (2.11) e (2.15), obtém-se a relação entre as transformações perspectivas fraca e total:

$$(x_i)_f = x_i(1 + \varepsilon_i) \quad (2.16)$$

$$(y_i)_f = y_i(1 + \varepsilon_i) \quad (2.17)$$

Uma característica importante desta aproximação é o ser válida, sempre que o objecto esteja na vizinhança do eixo óptico [67].

### Para-perspectiva

A transformação para-perspectiva é vista como uma aproximação de primeira ordem da transformação perspectiva total, [67]. Esta aproximação é equivalente a:

$$\frac{1}{1 + \varepsilon_i} \approx 1 - \varepsilon_i \quad (2.18)$$

Com esta aproximação, as equações (2.9) e (2.10) podem ser reescritas:

$$(x_i)_p - x_o = \frac{\mathbf{i}_s - x_o\mathbf{k}}{t_z} \cdot {}^o\mathbf{p}_i \quad (2.19)$$

$$(y_i)_p - y_o = \frac{\mathbf{j}_s - y_o\mathbf{k}}{t_z} \cdot {}^o\mathbf{p}_i \quad (2.20)$$

em que  $(x_i)_p$  e  $(y_i)_p$  significam os valores aproximados pela transformação para-perspectiva



das coordenadas métricas do ponto  $\mathbf{x}_i$ , na imagem. Através das duas equações anteriores e das equações (2.11) e (2.12), obtém-se a relação entre as transformações para-perspectiva e perspectiva total [67]:

$$(x_i)_p = x_i(1 + \varepsilon_i) - x_o\varepsilon_i \quad (2.21)$$

$$(y_i)_p = y_i(1 + \varepsilon_i) - y_o\varepsilon_i \quad (2.22)$$

O modelo para-perspectivo permite melhorar a aproximação da transformação perspectiva fraca sempre que a origem do referencial do objecto "o" seja convenientemente escolhida [67].

## 2.3 Geometria Epipolar

### 2.3.1 Matriz de Homografia

Considerando duas câmaras a "olhar" o mesmo objecto (figura 2.4) e sem perda de generalidade é assumido que a primeira câmara se encontra no referencial do mundo. Assim, as matrizes de projecção para a primeira e segunda câmara são definidas por:

$$\mathbf{P}_1 = \mathbf{K}_1 \cdot \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (2.23)$$

$$\mathbf{P}_2 = \mathbf{K}_2 \cdot \begin{bmatrix} {}^1\mathbf{R}_2 & {}^1\mathbf{t}_2 \end{bmatrix} \quad (2.24)$$

Considere um plano  $\pi$  associado ao objecto, que não contém os eixos ópticos de cada uma das câmaras. Este plano é definido pela sua normal  $\mathbf{n}$  e pela distância  $d$  do plano ao centro óptico da primeira câmara, medida segundo a normal ao plano. Considere ainda os pontos  $\tilde{\mathbf{p}}_1$  e  $\tilde{\mathbf{p}}_2$ , como sendo as projecções de um mesmo ponto 3D do objecto,  $\tilde{\mathbf{x}}$ , dentro do campo de visão de ambas as câmaras. Ambos os pontos  $\tilde{\mathbf{p}}_1$  e  $\tilde{\mathbf{p}}_2$  estão relacionados por

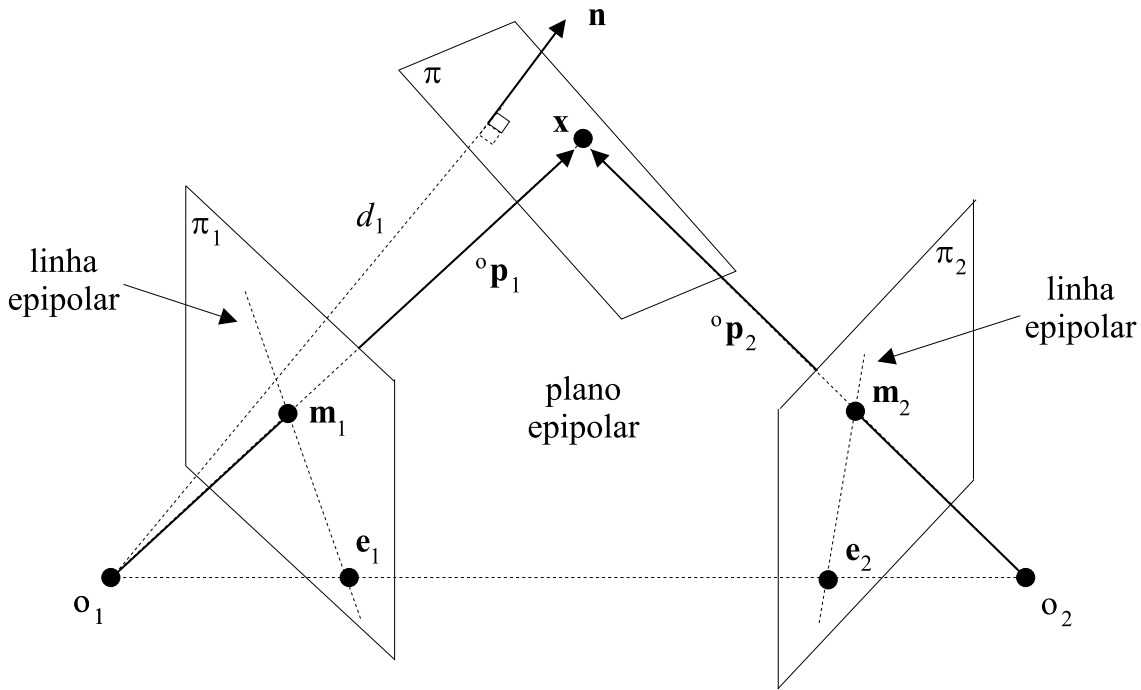


Figura 2.4: A geometria epipolar.

uma colineação em  $\mathbb{P}^2$ ,  $H_{12}$ , que em seguida se deduz. O ponto  $\tilde{\mathbf{x}}$  satisfaz simultaneamente a equação de projecção relativa a qualquer uma das câmaras e a equação do plano  $\pi$ ,

$$\lambda \tilde{\mathbf{p}}_1 = \mathbf{K}_1 \cdot \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \end{bmatrix} \cdot \tilde{\mathbf{x}} \quad (2.25)$$

$$\begin{bmatrix} \mathbf{n}^T & -d_1 \end{bmatrix} \cdot \tilde{\mathbf{x}} = 0 \quad (2.26)$$

As duas igualdades anteriores podem agrupar-se na seguinte forma matricial,

$$\begin{bmatrix} \lambda \cdot \mathbf{K}_1^{-1} \cdot \tilde{\mathbf{p}}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{n}^T & -d_1 \end{bmatrix} \cdot \tilde{\mathbf{x}} \quad (2.27)$$

Tendo sido definido que o plano  $\pi$  não passava por nenhum dos centros ópticos das câmaras e uma vez que a matriz do segundo membro da equação anterior é invertível, pois  $d_1 \neq 0$ ,

$$\tilde{\mathbf{x}} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3 \times 1} \\ \mathbf{n}^T & -\frac{1}{d_1} \end{bmatrix} \cdot \begin{bmatrix} \lambda \cdot \mathbf{K}_1^{-1} \cdot \tilde{\mathbf{p}}_1 \\ 0 \end{bmatrix} = \lambda \cdot \begin{bmatrix} \mathbf{I}_3 \\ \frac{\mathbf{n}^T}{d_1} \end{bmatrix} \cdot \mathbf{K}_1^{-1} \cdot \tilde{\mathbf{p}}_1 \quad (2.28)$$

Projectando o ponto  $\tilde{\mathbf{x}}$  na segunda câmara,

$$\lambda \tilde{\mathbf{p}}_2 = \mathbf{K}_2 \cdot \begin{bmatrix} {}^1\mathbf{R}_2 & {}^1\mathbf{t}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_3 \\ \frac{\mathbf{n}^T}{d_1} \end{bmatrix} \cdot \mathbf{K}_1^{-1} \cdot \tilde{\mathbf{p}}_1 \quad (2.29)$$

que resulta em,

$$\lambda \tilde{\mathbf{p}}_2 = \mathbf{K}_2 \cdot \left( {}^1\mathbf{R}_2 + {}^1\mathbf{t}_2 \cdot \frac{\mathbf{n}^T}{d_1} \right) \cdot \mathbf{K}_1^{-1} \cdot \tilde{\mathbf{p}}_1 \quad (2.30)$$

sobre a qual podemos retirar a seguinte relação

$$\lambda \tilde{\mathbf{p}}_2 = \mathbf{H}_{12} \cdot \tilde{\mathbf{p}}_1 \quad (2.31)$$

e a matriz de colineação planar, i.e. matriz de homografia,

$$\mathbf{H}_{12} = \mathbf{K}_2 \cdot \left( {}^1\mathbf{R}_2 + {}^1\mathbf{t}_2 \cdot \frac{\mathbf{n}^T}{d_1} \right) \cdot \mathbf{K}_1^{-1} \quad (2.32)$$

### Estimação da Matriz de Homografia

Para estimar a matriz de homografia, i.e. uma colineação planar, é necessário considerar a equação (2.31) e assim obter o seguinte sistema de equações:

$$\mathbf{A}_{\mathbf{H}} \cdot \mathbf{h} = 0 \quad (2.33)$$

em que  $\mathbf{A}_{\mathbf{H}}$  é a matriz ( $2n \times 9$ ) de coeficientes do sistema,  $\mathbf{h}$  é o vector com as nove componentes da matriz de homografia, e  $n$  é o número de pontos utilizados na estimação.

A matriz de coeficientes é assim definida:

$$\mathbf{A}_{\mathbf{H}} = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_2 u_1 & -u_2 v_1 & -u_2 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v_2 u_1 & -v_2 v_1 & -v_2 \\ & & & & & \vdots & & & \\ u_n & v_n & 1 & 0 & 0 & 0 & -u_n u_n & -u_n v_n & -u_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -v_n u_n & -v_n v_n & -v_n \end{bmatrix} \quad (2.34)$$

Para determinar a solução do sistema (2.33) são necessários no mínimo quatro pontos e resolver o problema clássico de minimização:

$$\mathbf{h} = \min_h \|\mathbf{A}_H \cdot \mathbf{h}\|, \quad \text{sujeito ao constrangimento} \quad \|\mathbf{h}\| = 1 \quad (2.35)$$

A solução para o problema exposto em (2.35) é o vector próprio que corresponde ao valor próprio mais pequeno de  $\mathbf{A}_H$ . O vector próprio é então calculado utilizando a decomposição SVD [103].

### Recuperação do movimento da câmara

A recuperação do movimento da câmara entre duas imagens sucessivas pode ser realizada através da matriz de homografia  $\mathbf{H}$  expressa em coordenadas métricas. Para tal é necessário utilizar os parâmetros intrínsecos da seguinte forma:

$$\mathbf{H} = \mathbf{K}_2^{-1} \cdot \mathbf{H}_{12} \cdot \mathbf{K}_1 \quad (2.36)$$

A matriz de homografia pode ser escrita através da soma de uma matriz de rotação e de uma matriz de característica 1 [40]:

$$\mathbf{H} = \mathbf{R} + \frac{\mathbf{t}}{d_1} \mathbf{n}^T \quad (2.37)$$

Tendo previamente estimado a matriz de homografia, é possível obter a matriz de rotação  $\mathbf{R}$ , o vector de translação  $\mathbf{t}$ , o escalar  $d_1$  e a normal ao plano do objecto  $\mathbf{n}$  utilizando o método proposto em [40] ou [137], sendo que existem duas soluções diferentes. É possível obter uma só solução se o objecto for planar e se for possível fixar a normal  $\mathbf{n}$ . No caso do objecto não ser planar é proposto em [40] um método para obter uma só solução, que

consiste na utilização de um novo plano de referência para escolher a solução que é comum aos dois planos.

### 2.3.2 Matriz Essencial e Matriz Fundamental

Os referenciais das câmaras 1 e 2, ver figura 2.4, estão relacionados pela respectiva rotação  $\mathbf{R}$  e translacção  $\mathbf{t}$ . Estas transformações permitem determinar a relação entre  ${}^o\mathbf{p}_1$  e  ${}^o\mathbf{p}_2$ ,

$${}^o\mathbf{p}_2 = \mathbf{R} \cdot ({}^o\mathbf{p}_1 - \mathbf{t}) \quad (2.38)$$

A relação entre um ponto no espaço cartesiano e a sua projecção perspectiva é descrita pelas equações na forma vectorial:

$$\tilde{\mathbf{m}}_i = \frac{f_i}{Z_i} \cdot {}^o\mathbf{p}_i \quad (2.39)$$

em que  $i = 1, 2$ .

A geometria epipolar descrita na figura 2.4 tem a característica de que o plano epipolar, identificado pelos pontos  $\mathbf{x}, \mathbf{o}_1, \mathbf{o}_2$ , intersecta cada imagem na linha epipolar. Esta linha contém os epipolos  $\mathbf{e}_1, \mathbf{e}_2$  definidos pela intersecção da linha definida pelos centros de projecção  $\mathbf{o}_1, \mathbf{o}_2$  e o plano da imagem. Assim, o epipolo  $\mathbf{e}_1$  é a imagem do centro de projecção  $\mathbf{o}_2$  e vice-versa para o epipolo  $\mathbf{e}_2$ . Outra característica importante é a do constrangimento epipolar [131], que permite obter um mapeamento entre pontos na imagem 1 e linhas na imagem 2, e vice-versa.

A equação do plano epipolar pode ser escrita através da condição de co-planaridade dos vectores  ${}^o\mathbf{p}_1, \mathbf{t}$  e  ${}^o\mathbf{p}_1 - \mathbf{t}$ , ver figura 2.4.

$$({}^o\mathbf{p}_1 - \mathbf{t})^T \cdot \mathbf{t} \times {}^o\mathbf{p}_1 = 0 \quad (2.40)$$

Utilizando a equação (2.38) em (2.40), obtém-se:

$$(\mathbf{R}^T \cdot {}^o\mathbf{p}_2)^T \cdot \mathbf{t} \times {}^o\mathbf{p}_1 = 0 \quad (2.41)$$

Sabendo que o produto externo entre dois vectores pode ser expresso através da utilização da matriz anti-simétrica  $\mathbf{S}$ , ver anexo A.1.1, i.e.

$$\mathbf{t} \times {}^o\mathbf{p}_1 = \mathbf{S}(\mathbf{t}) \cdot {}^o\mathbf{p}_1 \quad (2.42)$$

Assim a equação (2.41) pode ser escrita da seguinte forma:

$${}^o\mathbf{p}_2^T \cdot \mathbf{R} \cdot \mathbf{S}(\mathbf{t}) \cdot {}^o\mathbf{p}_1 = 0 \quad (2.43)$$

Denomina-se matriz essencial  $\mathbf{E}$ , a matriz dada por:

$$\mathbf{E} = \mathbf{R} \cdot \mathbf{S}(\mathbf{t}) \quad (2.44)$$

Utilizando as equações (2.39) e dividindo por  $Z_1 Z_2$ , a equação (2.43) é reescrita da seguinte forma:

$$\tilde{\mathbf{m}}_2^T \cdot \mathbf{E} \cdot \tilde{\mathbf{m}}_1 = 0 \quad (2.45)$$

que expressa a relação entre os pontos da imagem em coordenadas métricas.

Considerando a transformação entre os pontos da imagem em coordenadas métricas e em pixels, definida na equação (2.3) através dos parâmetros intrínsecos, a equação (2.45) toma a seguinte forma:

$$\tilde{\mathbf{p}}_2^T \cdot \mathbf{F} \cdot \tilde{\mathbf{p}}_1 = 0 \quad (2.46)$$

que expressa a relação entre os pontos da imagem em pixels.

A matriz  $\mathbf{F}$  (matriz fundamental) é expressa em função da matriz essencial através da expressão:

$$\mathbf{F} = \mathbf{K}_2^{-T} \cdot \mathbf{E} \cdot \mathbf{K}_1^T \quad (2.47)$$

Conclui-se assim que formalmente as matrizes fundamental e essencial são similares, uma vez que ambas permitem a reconstrução da geometria epipolar. A principal diferença é que a matriz fundamental é definida com os pontos da imagem em pixels e a matriz essencial é definida com os pontos da imagem em coordenadas métricas. Note-se que esta matriz é singular e com característica dois, i.e.  $\det(\mathbf{F}) = 0$ , o que implica a introdução de um constrangimento não-linear para obter os elementos da matriz fundamental [86].

### Estimação da Matriz Fundamental

Entre os vários métodos para estimar a matriz fundamental, o mais simples e um marco na literatura é o algoritmo dos oito pontos (*eight point algorithm*). Este algoritmo foi proposto por Longuet-Higgins [83], sendo que Hartley [58] introduziu no algoritmo um procedimento de normalização para evitar instabilidades numéricas. Outros algoritmos para estimação da matriz fundamental são revistos em [3, 86, 131].

O algoritmo requer no mínimo as coordenadas  $u, v$  de oito pontos observados pelas câmaras 1 e 2. Cada ponto gera uma equação linear homogénea (2.46) para as nove entradas da matriz  $\mathbf{F}$ :

$$u_1 u_2 f_{11} + u_1 v_2 f_{21} + u_1 f_{31} + u_2 v_1 f_{12} + v_2 v_1 f_{22} + v_1 f_{32} + u_2 f_{13} + v_2 f_{23} + f_{33} = 0 \quad (2.48)$$

obtendo-se assim um sistema de equações lineares:

$$\mathbf{A}_\mathbf{F} \cdot \mathbf{f} = 0 \quad (2.49)$$

em que  $\mathbf{A}_\mathbf{F}$  é a matriz ( $n \times 9$ ) de coeficientes do sistema,  $\mathbf{f}$  é o vector com as nove componentes da matriz fundamental, e  $n$  é o número de pontos utilizados na estimação.

Para determinar a solução do sistema (2.49) são necessários no mínimo oito pontos e resolver o problema clássico de minimização:

$$\mathbf{f} = \min_f \|\mathbf{A}_F \cdot \mathbf{f}\|, \quad \text{sujeito ao constrangimento} \quad \|f\| = 1 \quad (2.50)$$

Utilizando a decomposição SVD [103] é obtida a seguinte decomposição,

$$\mathbf{A}_F = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^T \quad (2.51)$$

A solução para o problema é a coluna da matriz  $\mathbf{V}$  que corresponde ao único valor próprio nulo (ou de menor valor na presença de ruído) de  $\mathbf{A}_F$ . Para assegurar o constrangimento associado à singularidade da matriz fundamental é necessário realizar a decomposição SVD da já estimada matriz  $\mathbf{F} = \mathbf{U} \cdot \mathbf{D} \cdot \mathbf{V}^T$ . Posteriormente, é necessário substituir artificialmente na matriz  $\mathbf{D}$ , entretanto estimada, o valor da diagonal mais pequeno por zero e assim obter a nova matriz  $\mathbf{D}'$ . A matriz fundamental com o constrangimento de singularidade corrigido é então obtida através da matriz de coeficientes corrigida:  $\mathbf{F}' = \mathbf{U} \cdot \mathbf{D}' \cdot \mathbf{V}^T$ .

### Recuperação do movimento da câmara

A partir da estimação da matriz fundamental é possível obter a matriz essencial a menos de um factor de escala [131], utilizando a equação (2.47), em que se assume o conhecimento dos parâmetros intrínsecos:

$$\mathbf{E} = \mathbf{K}_2^T \cdot \mathbf{F} \cdot \mathbf{K}_1 \quad (2.52)$$

Através da definição da matriz essencial (2.44), tem-se que:

$$\mathbf{E}^T \cdot \mathbf{E} = \mathbf{S}(\mathbf{t})^T \cdot \mathbf{S}(\mathbf{t}) \quad (2.53)$$

e que:

$$\mathbf{E}^T \cdot \mathbf{E} = \begin{bmatrix} t_y^2 + t_z^2 & -t_x t_y & -t_x t_z \\ -t_y t_x & t_x^2 + t_z^2 & -t_y t_z \\ -t_z t_x & -t_z t_y & t_x^2 + t_y^2 \end{bmatrix} \quad (2.54)$$



Uma vez que a matriz essencial é definida a menos de um factor de escala arbitrário, é necessário normalizá-la. Este processo é realizado recorrendo ao facto de o traço da matriz essencial estar relacionado com a norma do vector de translação  $\mathbf{t}$ , da seguinte forma:

$$\text{Traço}(\mathbf{E}^T \cdot \mathbf{E}) = 2 \cdot \|\mathbf{t}\|^2 \quad (2.55)$$

Dividindo todos os elementos da matriz definida em (2.54) por:

$$\sqrt{\text{Traço}(\mathbf{E}^T \cdot \mathbf{E})/2} \quad (2.56)$$

obtém-se a normalização da matriz  $\mathbf{E}^T \mathbf{E}$  e ao mesmo tempo do vector  $\mathbf{t}$ :

$$\overline{\mathbf{E}}^T \cdot \overline{\mathbf{E}} = \begin{bmatrix} \bar{t}_y^2 + \bar{t}_z^2 & -\bar{t}_x \bar{t}_y & -\bar{t}_x \bar{t}_z \\ -\bar{t}_y \bar{t}_x & \bar{t}_x^2 + \bar{t}_z^2 & -\bar{t}_y \bar{t}_z \\ -\bar{t}_z \bar{t}_x & -\bar{t}_z \bar{t}_y & \bar{t}_x^2 + \bar{t}_y^2 \end{bmatrix} \quad (2.57)$$

em que  $\overline{\mathbf{E}}$  representa a matriz essencial normalizada e  $\bar{\mathbf{t}}$  o vector de translação normalizado. Este vector retira-se facilmente dos elementos da matriz definida em (2.57). Note-se que os elementos da matriz encontram-se definidos relativamente aos valores quadráticos dos componentes do vector de translação, induzindo duas soluções para este vector que diferem de um sinal [131].

Supondo que o vector de translação foi devidamente recuperado, a matriz de rotação é recuperada através das equações que em seguida se apresentam. Definindo o vector:

$$\mathbf{w}_i = \bar{\mathbf{e}}_i \times \bar{\mathbf{t}} \quad (2.58)$$

em que  $i = 1, 2, 3$  e  $\bar{\mathbf{e}}_i$  são as três linhas da matriz normalizada  $\overline{\mathbf{E}}$ . Definindo  $\mathbf{r}_i$  como as três linhas da matriz de rotação, a seguinte equação pode ser obtida:

$$\mathbf{r}_i = \mathbf{w}_i + \mathbf{w}_j \times \mathbf{w}_k \quad (2.59)$$

em que  $i, j, k$  são as permutações cíclicas de (1,2,3).

A recuperação da matriz de rotação e do vector de translação, a partir da matriz essencial leva a que sejam encontradas quatro soluções [131], i.e. quatro pares  $(\mathbf{R}, \bar{\mathbf{t}})$ . Para encontrar a solução verdadeira são reconstruídas as coordenadas  $Z_1$  e  $Z_2$ , do ponto  $\mathbf{x}$  no espaço euclideano, e escolhido o par  $(\mathbf{R}, \bar{\mathbf{t}})$  em que as coordenadas anteriores são ambas positivas.

## 2.4 Extracção de Características de um Objecto

O controlo visual de robôs manipuladores requer o reconhecimento de um determinado objecto e sua caracterização, através de imagem. Os pontos na imagem são as primitivas geométricas mais utilizadas em controlo visual de robôs. Este tipo de primitivas pode ainda ser utilizado para obter da imagem, através da fase de interpretação, ver figura 2.1, outra informação relevante tal como seja a pose do objecto relativamente à câmara, sub-secção 2.4.2, ou a rotação e translação da câmara durante o seu movimento entre duas posições consecutivas, sub-secção 2.3. Outro tipo de características podem ser utilizadas tais como linhas [20], segmentos de recta e elipses, [20], e momentos [22].

### 2.4.1 Primitivas de um Objecto na Imagem

Na presente tese são utilizados pontos, como a primitiva geométrica, necessária para caracterizar o objecto. Devido à construção do objecto/alvo, anexo B, este é caracterizado por círculos na imagem, cujos centróides serão os pontos usados na presente tese como características do objecto na imagem. Condições especiais de iluminação levam a que na imagem capturada apenas exista o objecto pretendido, o que torna simples o seu reconhecimento. No anexo B é descrito com detalhe o *hardware* utilizado e *software* desenvolvido, no sistema de visão do aparato experimental.

### 2.4.2 Cálculo da Pose de um Objecto

O cálculo da pose consiste na determinação da posição e orientação de um objecto relativamente a uma câmara, utilizando para tal uma imagem desse objecto e o conhecimento prévio do modelo desse objecto. Este cálculo serve de base ao controlo visual baseado em posição, como se descreverá na secção 3.3.2 e ainda no capítulo 4. É ainda de destacar o facto de que o cálculo da pose tem suscitado grande interesse em visão por computador. Este interesse é justificado pelas aplicações possíveis, tais como o próprio controlo visual de robôs, realidade aumentada [27] e o seguimento de objectos. É desejável que um algoritmo para determinar a pose seja robusto, computacionalmente rápido, preciso e que possa ser rapidamente implementado. Se os requisitos expostos forem satisfeitos poder-se-á aplicar o controlo visual baseado em posição, permitindo assim que o robô possa ser controlado directamente através das coordenadas cartesianas do elemento terminal. Contudo o cálculo da pose é um dos problemas no controlo visual de robôs quando se pretende aplicar algoritmos baseados em posição ou híbridos, ver secção 3.3.3.

Existem publicados numerosos métodos para determinar a pose. Estes são baseados no conhecimento dos parâmetros intrínsecos de calibração da câmara e do modelo tridimensional do objecto. A maioria dos métodos publicados tem como base de cálculo a primitiva geométrica o ponto, existindo também métodos com outras primitivas tais como: segmentos, linhas, contornos, cónicas e objectos cilíndricos, como referido em [27]. Os métodos de cálculo da pose podem ser divididos em várias classes segundo o método de cálculo utilizado: métodos analíticos; métodos numéricos, tanto lineares como não-lineares.

Os métodos puramente analíticos [35, 66] baseiam-se na resolução de equações lineares e revelam pouca precisão e robustez nos resultados, sendo difícil a sua aplicação prática. Os métodos numéricos podem ainda ser subdivididos em lineares iterativos e não-lineares. Os métodos numéricos lineares iterativos têm a particularidade de numa primeira fase

começar como uma transformação perspectiva fraca, Dementhon & Davis [33], ou com uma transformação para-perspectiva, [67], e iterativamente recuperar a transformação perspectiva total. O método original de Dementhon & Davis [33] aplica-se a um conjunto de pontos não coplanares, e a sua extensão para o caso de um conjunto de pontos coplanares foi feita em [100]. Posteriormente o método original foi extendido em [67], para o caso de pontos coplanares e não-coplanares, em que é utilizada uma transformação para-perspectiva em vez da transformação perspectiva fraca. Este método tem ainda a característica de garantir a ortonormalidade da matriz de rotação estimada. Os métodos lineares iterativos têm a particularidade de serem rápidos e relativamente precisos, muito por causa das simplificações iniciais efectuadas à transformação perspectiva total.

Os métodos não-lineares [1, 84, 85], têm a particularidade de minimizar o erro quadrático entre as observações dos pontos do objecto na imagem e a projecção do modelo do objecto para uma determinada pose entretanto recuperada. Esta minimização é geralmente realizada utilizando técnicas clássicas como a de Newton-Raphson ou de Levenberg-Marquardt [103]. Esta classe de métodos para recuperar a pose tem como principal vantagem a precisão dos resultados, acompanhada com a desvantagem da complexidade do método e o facto de este ser computacionalmente exigente. Existem ainda as desvantagens inerentes à utilização das técnicas numéricas de minimização, i.e. necessitam de uma inicialização próxima do valor exacto, podem encontrar mínimos locais e ainda podem divergir da solução óptima.

O trabalho proposto por Lowe [84], teve a característica de ser pioneiro na utilização de técnicas de optimização e na demonstração de que estes podem aumentar a robustez do cálculo da pose, relativamente aos métodos analíticos. Em [1] é proposta uma evolução do método original de Lowe considerando a transformação perspectiva total, i.e. não são realizadas quaisquer aproximações na derivação das equações do algoritmo de recuperação da pose. Em [85] é proposto uma nova técnica para resolver o problema de optimização (*orthogonal iteration*) em contraposição às técnicas clássicas, que segundo o autor é glo-

balmente convergente e rápido o suficiente para aplicações em tempo real. Tudo isto conjugado com o facto de ter a mesma precisão que as técnicas clássicas.

A simplicidade, a rapidez de convergência e os resultados obtidos através do algoritmo de Dementhon, levaram que este tenha sido utilizado para a estimação da pose no controlo visual baseado em posição. Quando se pretendem resultados mais precisos para a estimação da pose é utilizado um algoritmo numérico não-linear, em que a sua inicialização é usualmente a pose determinada pelo algoritmo de Dementhon. Assim e nesta tese é utilizado o algoritmo de Dementhon para a estimação da pose, tendo também em atenção as simplificações que podem ser realizadas dada a natureza do aparato experimental, descrito no apêndice B.

Dementhon & Davis verificaram que as equações da transformação perspectiva total (2.11 e 2.12) e fraca (2.14 e 2.15) eram bastante similares. Assim, atribuindo um valor nulo a  $\varepsilon_i$  verificaram que:

- sempre que se fixa o valor de  $\varepsilon_i$ , as equações (2.11) e (2.12) tornavam-se lineares em  $\mathbf{i}_s$  e  $\mathbf{j}_s$  e que era possível obter uma solução com pelo menos três pontos do objecto;
- era possível resolver iterativamente as equações (2.11) e (2.12) por sucessivas aproximações lineares.

O método iterativo proposto em [33] começa por utilizar o modelo perspectivo fraco para iterativamente se aproximar do modelo perspectivo total e assim determinar a pose. Em seguida é apresentado o algoritmo recorrendo à notação apresentada na secção 2.2.1:

1. Para todos os pontos ( $i$ ) do objecto,  $i \in \{1, 2, \dots, n\}$ ,  $n \geq 3$ ,  $\varepsilon_i = 0$
2. Resolver o sistema dado pelas equações (2.11) e (2.12) nos  $n$  pontos do objecto, para assim obter uma estimação dos vectores  $\mathbf{i}_s$  e  $\mathbf{j}_s$

3. Calcular a posição e orientação do referencial do objecto relativamente ao referencial da câmara, através das equações:

$$\begin{aligned} t_z &= \frac{1}{2} \left( \frac{1}{\|\mathbf{i}_s\|} + \frac{1}{\|\mathbf{j}_s\|} \right) \\ t_x &= x_o \cdot t_z \\ t_y &= y_o \cdot t_z \\ \mathbf{i} &= \frac{\mathbf{i}_s}{\|\mathbf{i}_s\|} \\ \mathbf{j} &= \frac{\mathbf{j}_s}{\|\mathbf{j}_s\|} \\ \mathbf{k} &= \mathbf{i} \times \mathbf{j} \end{aligned}$$

4. Para todos os pontos ( $i$ ) do objecto, calcular:

$$\varepsilon_i = \frac{\mathbf{k} \cdot {}^o\mathbf{p}_i}{t_z} \quad (2.60)$$

e se o erro entre o valor calculado na actual iteração e na iteração anterior for igual ou inferior a um determinado valor limite, parar o algoritmo. Caso contrário voltar ao passo 2.

No passo dois do algoritmo é necessário resolver um sistema de equações lineares (2.11) e (2.12) em ordem a  $\mathbf{i}_s$  e  $\mathbf{j}_s$ , para  $n$  pontos do objecto. Como tal é necessário escrever estas equações na forma matricial:

$$\mathbf{P} \cdot \mathbf{i}_s = \mathbf{m}_x \quad (2.61)$$

$$\mathbf{P} \cdot \mathbf{j}_s = \mathbf{m}_y \quad (2.62)$$

em que  $\mathbf{P}$  é uma matriz  $n \times 3$  formada pelas coordenadas euclidianas dos pontos  $i$  do objecto,  ${}^o\mathbf{p}_i$ ,  $i = 1, \dots, n$ :

$$\mathbf{P} = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix} \quad (2.63)$$

e em que  $\mathbf{m}_x = [x_1, \dots, x_n]^T$  e  $\mathbf{m}_y = [y_1, \dots, y_n]^T$ .

Para o caso em que o objecto é caracterizado por um conjunto de pontos não-coplanares, a característica da matriz é sempre três e os vectores  $\mathbf{i}_s$  e  $\mathbf{j}_s$  podem ser determinados por:

$$\mathbf{i}_s = (\mathbf{P}^T \cdot \mathbf{P})^{-1} \cdot \mathbf{P}^T \cdot \mathbf{m}_x \quad (2.64)$$

$$\mathbf{j}_s = (\mathbf{P}^T \cdot \mathbf{P})^{-1} \cdot \mathbf{P}^T \cdot \mathbf{m}_y \quad (2.65)$$

pode-se verificar que nas equações anteriores a pseudo-inversa da matriz  $\mathbf{P}$  pode ser calculada *off-line*, o que diminui o tempo de computação.

No caso dos pontos que caracterizam o objecto serem coplanares, i.e. o objecto ser planar, a característica da matriz  $\mathbf{P}$  não é três mas sim dois e a solução anterior não pode ser considerada. Neste caso é seguido o método proposto em [100] para o cálculo dos vectores  $\mathbf{i}_s$  e  $\mathbf{j}_s$ . Assim, cada um destes vectores é descrito pela soma de um outro vector pertencente ao mesmo plano e outro perpendicular a esse plano,  $\mathbf{u}$ :

$$\mathbf{i}_s = \mathbf{i}_{s0} + \lambda \mathbf{u} \quad (2.66)$$

$$\mathbf{j}_s = \mathbf{j}_{s0} + \mu \mathbf{u} \quad (2.67)$$

substituindo as expressões anteriores nas equações (2.61) e (2.62), obtém-se:

$$\mathbf{P} \cdot \mathbf{i}_{s0} = \mathbf{m}_x \quad (2.68)$$

$$\mathbf{P} \cdot \mathbf{j}_{s0} = \mathbf{m}_y \quad (2.69)$$

em que estas equações podem ser resolvidas desde que os constrangimentos lineares seguintes sejam satisfeitos:

$$\mathbf{u} \cdot \mathbf{i}_{s0} = 0 \quad (2.70)$$

$$\mathbf{u} \cdot \mathbf{j}_{s0} = 0 \quad (2.71)$$

Assim, obtêm-se as soluções:

$$\mathbf{i}_{s0} = (\mathbf{P}'^T \cdot \mathbf{P}')^{-1} \cdot \mathbf{P}'^T \cdot \begin{bmatrix} \mathbf{m}_x \\ 0 \end{bmatrix} \quad (2.72)$$

$$\mathbf{j}_{s0} = (\mathbf{P}'^T \cdot \mathbf{P}')^{-1} \cdot \mathbf{P}'^T \cdot \begin{bmatrix} \mathbf{m}_y \\ 0 \end{bmatrix} \quad (2.73)$$

em que  $\mathbf{P}'$  tem característica três e é dada por:

$$\mathbf{P}' = \begin{bmatrix} \mathbf{P} \\ \mathbf{u} \end{bmatrix} \quad (2.74)$$

Chegando a este ponto é necessário agora determinar os escalares  $\lambda$  e  $\mu$ , para assim estimar os vectores  $\mathbf{i}_s$  e  $\mathbf{j}_s$ . Em [100] são introduzidos os constrangimentos  $\|\mathbf{i}_s\| = \|\mathbf{j}_s\|$  e  $\mathbf{i}_s \cdot \mathbf{j}_s = 0$ , necessários para obter os vectores desejados. No entanto e devido à coplanaridade do conjunto de pontos que define o objecto, em cada iteração do algoritmo surgem duas soluções, o que no caso extremo levaria a existirem  $2^n$  soluções para  $n$  iterações. Como primeira abordagem é logo descartada a solução que envolve  $t_z \leq 0$ , em seguida denominada solução impossível. Sempre que continuarem a existir duas soluções possíveis é adoptada a seguinte metodologia:

- se na primeira iteração for encontrada uma solução impossível, as soluções possíveis são imediatamente diminuídas em metade. Nas iterações seguintes foi verificado [100] que continua a surgir uma solução impossível, o que leva a que exista no final uma única solução possível.
- se na primeira iteração não for encontrada uma solução impossível é necessário percorrer todas as soluções possíveis e ir descartando as de menor qualidade. Para tal é necessário definir um escalar como medida de qualidade, que será a distância média entre os pontos actuais da imagem e a projecção dos pontos do objecto através da pose calculada, como descrito em [100].



## 2.5 Correspondências de Características em Imagens Sucessivas

A correspondência entre características de um determinado objecto em imagens sucessivas, é alvo de estudo detalhado na comunidade científica internacional. Esta área da visão por computador tem evoluído bastante até ao ponto onde se procura realizar correspondências entre imagens de cenas reais em ambientes de trabalho em constante mutação. Em [131] são apresentados vários métodos para obter correspondências e que são baseados em técnicas de correlação e em técnicas baseadas nas características de objectos. As técnicas de correlação incluem o método SSD (*sum of squared differences*), baseado na procura de correspondências numa janela de tamanho fixo ou variável, através da maximização de um critério de similaridade. No caso das técnicas baseadas em características do objecto, estas são extraídas em cada imagem capturada, por exemplo a orientação de uma linha na imagem, e posteriormente aplicado um critério de similaridade.

Nesta tese e para demonstração do funcionamento das leis de controlo visual de robôs manipuladores, apresentadas nos capítulos seguintes, o ambiente de trabalho do robô é controlado por forma a facilitar o processamento de imagem e a obtenção das correspondências. Assim, o algoritmo para determinar as correspondências entre imagens sucessivas, utilizado na presente tese e descrito no apêndice B, consiste na procura numa vizinhança pré-definida de cada ponto, identificado na imagem como do objecto.



# Capítulo 3

## Controlo Visual de Robôs Manipuladores

### 3.1 Introdução

A utilização de visão por computador para controlar um robô é denominada na literatura [72] como *visual servoing*, controlo visual de robôs, tendo sido o termo anglo-saxónico proposto por Hill e Park [65]. A inclusão da visão em sistemas robóticos tem como principal objectivo o aumento da flexibilidade e precisão destes sistemas, sendo que um dos primeiros trabalhos a utilizar a visão para controlo de robôs foi apresentado por Shirai e Inoue [116], há mais de três décadas. O papel da visão por computador é o de fornecer ao robô o estado do seu ambiente de trabalho, para que esta informação possa ser utilizada no seu controlo. No caso de robôs manipuladores, a classe de robôs abordada na presente dissertação, no seu ambiente de trabalho encontra-se um, ou vários, objecto(s) a manipular.

A informação sobre o ambiente de trabalho do robô obtida através da imagem, informação visual, pode ser utilizada para controlo de duas formas distintas [72]. A primeira abordagem é denominada na literatura como *Open-Loop-Robot-Control*, controlo em anel aberto e tem por base a separação entre a parte relacionada com a obtenção e processamento da informação visual e o controlo do robô. A designação de controlo em anel aberto surge de-

vido à não utilização de informação visual durante o controlo, pois o anel de retroacção é fechado com coordenadas de junta ou cartesianas. Pode-se afirmar que segundo esta abordagem a posição inicial e desejada do robô é determinada *off-line* através da informação visual, sendo que o movimento do robô é feito "às cegas" pois o controlo é unicamente realizado com base nos sensores de junta. As posições iniciais e finais são determinadas através de algoritmos de estimação da pose, sendo para tal necessário calibrar a câmara e ter o modelo do objecto.

A segunda abordagem, à qual se convencionou chamar *visual servoing*, utiliza directa ou indirectamente na lei de controlo a informação visual. Neste caso, é usualmente utilizado um controlo interno do robô, com retroacção das variáveis de junta para estabilizar o robô em torno da acção de comando vinda do anel exterior de visão. A esmagadora maioria dos sistemas existentes na bibliografia seguem esta segunda abordagem [72]. A utilização directa de informação visual implica ter como variáveis controladas as coordenadas 2D do plano da imagem, enquanto a utilização indirecta de informação visual implica a extracção de características relevantes do objecto a partir das coordenadas 2D do plano da imagem. Como exemplo desta última abordagem temos a pose 3D do objecto relativamente à câmara.

As aplicações típicas de controlo visual de robôs podem ser classificadas como i) *posicionamento* do robô ou do seu elemento terminal relativamente a um determinado objecto; ii) *seguimento* de um objecto em movimento, mantendo uma relação constante entre este e o robô. O objecto em causa é parte fundamental do processo, pois é relativamente a este e à forma como é caracterizado, através da imagem, que a formulação do problema é obtida. É portanto fundamental obter informação visual que caracterize o objecto, sendo esta utilizada para medir o erro entre a posição actual do robô e a sua posição desejada. Esta informação visual, pode ser obtida por uma ou mais câmaras em que esta(s) pode(m) estar colocada(s) no espaço (visualizando o objecto e o elemento terminal do robô) ou no elemento terminal (visualizando o objecto).

A caracterização do objecto através da imagem pode ser realizada i) utilizando informação

bi-dimensional expressa directamente nas coordenadas do plano da imagem ou ii) utilizando informação tri-dimensional em que os modelos da câmara e objecto são utilizados para determinar a pose do objecto relativamente aos referenciais da câmara, robot ou do mundo [72]. Actualmente esta caracterização pode também englobar as duas abordagens anteriores, por forma a melhorar o desempenho global do sistema. Esta última abordagem tem sido denominada na literatura como controlo visual híbrido, *hybrid visual servoing* [60], tendo o primeiro trabalho nesta área sido apresentado por Malis em [91].

Para utilizar as características do objecto na imagem por forma a controlar robôs, é necessário estabelecer a relação entre estas e as coordenadas da câmara, relativamente ao referencial do mundo ou do elemento terminal do robô. Esta relação, modelo de interacção, tem sido objecto da esmagadora maioria dos trabalhos na área de controlo visual de robôs, existindo três abordagens consideradas como as mais importantes, por motivos históricos, ao problema. As abordagens denominam-se controlo visual 2D, controlo visual 3D e controlo visual híbrido.

Os modelos da interacção entre as coordenadas da câmara e as características do objecto na imagem, podem ser agrupados em dois grupos. O primeiro engloba os modelos conhecidos a priori através de relações analíticas e o segundo grupo engloba os modelos estimados numericamente. Estes modelos são utilizados para gerar os comandos de movimento para o robô e são denominados na literatura, entre outros termos [60], por jacobiano da imagem. Existe ainda a possibilidade de relacionar directamente as características do objecto na imagem com as variáveis de junta do robô, sendo para tal necessário o conhecimento do jacobiano do robô, e da relação que existe entre a posição do elemento terminal e da câmara.

A literatura existente sobre controlo visual de robôs cresceu substancialmente, ao longo dos anos, em torno dos problemas relacionados com a cinemática da lei de controlo. Neste tipo de controlo, controlo visual cinemático, o robô é considerado como um posicionador perfeito. Em [72] este tipo de controlo é definido como *dynamic look-and-move*. Na realidade o modelo de um robô tem não-linearidades associadas, sendo pois vantajoso

tê-las em consideração na obtenção da lei de controlo. Este último tipo de controlo é denominado controlo visual dinâmico [12, 28, 76], podendo também ter em consideração problemas relacionados com o atraso gerado pelo anel de visão. Quando o controlo visual dinâmico calcula directamente os binários para os actuadores do robô, este é denominado, *direct visual servo* [72].

Nos seguintes sub-capítulos serão abordados e aprofundados os temas referidos nesta introdução, que globamente expressam o estado da arte na área de controlo visual de robôs manipuladores.

## 3.2 Informação Visual para Controlo

Em robôs manipuladores controlados por visão são genericamente utilizadas medidas visuais obtidas através de características na imagem ou imagens, de objectos colocados no espaço de trabalho do robô. A obtenção das referidas medidas depende do número de câmaras utilizadas, da sua configuração relativamente ao robô, da sua calibração e ainda do conhecimento prévio que se poderá obter do ambiente de trabalho onde se encontra o objecto. Tendo também em conta o descrito anteriormente, esta informação visual pode ainda ser obtida directamente (baseada nas medidas das características do objecto no plano da imagem) ou indirectamente da imagem (baseado na estimação da pose, posição tri-dimensional do objecto relativamente à(s) câmara(s), estimação do movimento da câmara). A utilização em conjunto dos dois tipos de informação visual descritos, permite ainda obter vários grupos distintos de medidas do objecto.

Como foi atrás referido, a informação visual necessária ao controlo visual de robôs manipuladores depende de dois factores fundamentais, o primeiro relacionado com a parte física da interacção entre a(s) câmara(s) e o objecto e o segundo com a parte computacional necessária ao processamento e à interpretação da imagem (ou imagens) capturada(s). Relativamente ao segundo factor, este é documentado no capítulo 2 do presente docu-

mento e na secção 3.3 do presente capítulo.

O modelo de interacção entre a(s) câmara(s), o robô e o objecto que existe dentro do espaço de trabalho do robô, como visto anteriormente, depende dos parâmetros da câmara e do conhecimento prévio que se poderá obter do ambiente de trabalho onde se encontra o objecto, temas abordados no Capítulo 2. Existem ainda dependências relativamente:

- ao número de câmaras do sistema, permitindo desde logo definir **visão monocular** (uma câmara) e **visão estéreo** (duas câmaras ligadas rigidamente) e ainda **sistemas de câmaras redundantes**;
- à configuração da(s) câmara(s) relativamente ao robô, em que existe um caso em que a(s) câmara(s) se encontra(m) ligada rigidamente ao elemento terminal do robô e a olhar para o objecto, *eye-in-hand*, e um outro caso em que a(s) câmara(s) se encontra(m) no espaço e a olhar para o elemento terminal do robô e para o objecto simultaneamente, *eye-to-hand*;

sendo que estes temas irão ser abordados nas sub-secções seguintes.

### 3.2.1 Visão Monocular

Um sistema de visão monocular utiliza somente uma câmara, que pode ser colocada no elemento terminal do robô a olhar o objecto ou ainda no espaço a olhar simultaneamente para o elemento terminal do robô e para o objecto. A principal vantagem deste tipo de sistema é o de minimizar o tempo de processamento necessário para extrair as necessárias informações visuais, sendo outra vantagem o seu custo. No entanto tem a desvantagem de não ser possível determinar a profundidade, distância entre a câmara e o objecto, de uma forma exacta sendo necessária a sua estimação utilizando um modelo do objecto, ver capítulo 2, ou ainda utilizar outro tipo de abordagens durante o controlo, ver secção 3.3.1.

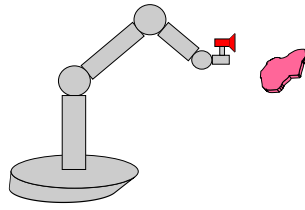


Figura 3.1: Visão monocular com a câmara colocada no elemento terminal e a olhar para o objecto, *eye-in-hand*.

### Câmara colocada no elemento terminal do robô e a olhar o objecto

O sistema de visão monocular em que a câmara se encontra colocada no elemento terminal do robô, *eye-in-hand*, e a olhar para o objecto, ver figura 3.1, tem a particularidade de ser o mais comum nas aplicações de controlo visual de robôs manipuladores. Embora a matriz de transformação entre o referencial do elemento terminal e o referencial da câmara necessite ser determinada, este procedimento pode ser realizado facilmente pois a câmara e o elemento terminal do robô encontram-se rigidamente ligados.

Com este tipo de configuração as aplicações típicas são as de seguir um determinado objecto dentro da imagem ou de mover o robô entre duas posições pré-definidas na imagem, usualmente na mesma imagem. Como exemplo de posições definidas em imagens diferentes, temos o recente trabalho de Remazeilles [106], baseado numa base de dados de imagens, memória visual.

### Câmara a olhar o robô manipulador e o objecto

O tipo de sistemas em que uma câmara se encontra a olhar o robô manipulador e o objecto, ver figura 3.2, *eye-to-hand*, foram os primeiros a surgir nos trabalhos sobre controlo visual de robôs manipuladores, [116]. Estes tipos de sistemas requerem calibração pois utilizam como variáveis a controlar a pose do objecto relativamente à câmara ou ao elemento terminal do robô. Neste tipo de sistemas é ainda necessário determinar a posição do elemento terminal do robô relativamente à câmara, o mesmo é dizer as coordenadas da



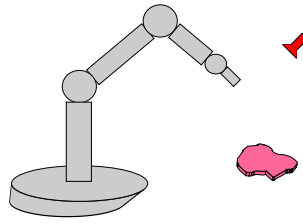


Figura 3.2: Visão monocular com a câmara a olhar para o objecto e para o robô manipulador, *eye-to-hand*.

câmara no referencial do mundo.

### 3.2.2 Visão Estéreo

Um sistema de visão estereo é constituído por duas câmaras rigidamente ligadas com a finalidade de obter duas imagens do mesmo objecto. Este sistema de visão é bastante útil quando se pretende obter informação tri-dimensional, por exemplo de um determinado objecto dentro do campo de visão do par estereo. Relativamente ao sistema de visão monocular, a utilização de visão estereo facilita a obtenção da profundidade embora penalizando o tempo de processamento das imagens e o custo do sistema. Em seguida são apresentadas as configurações mais usuais deste sistema de visão, em tudo idênticas à visão monocular, i.e. com as câmaras a olhar para o objecto ou a olhar para o elemento terminal do robô e para o objecto simultaneamente.

#### Câmaras a olhar o objecto

A utilização de um par estereo rigidamente ligado ao elemento terminal, ver figura 3.3, *eye-in-hand*, não é muito usual, devido ao facto de a este tipo de sistema estar associado um maior volume (relativamente à visão monocular) o que dificultará outro tipo de aplicações, tais como a manipulação de objectos. Contudo, a miniaturização cada vez maior dos sistemas de visão tem vindo a esbater este obstáculo. A utilização deste tipo de sistema introduz um outro problema que tem a ver com a perda de precisão dos algoritmos de

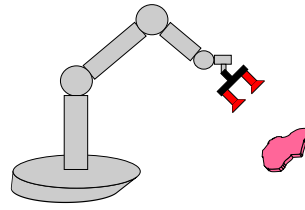


Figura 3.3: Visão estéreo com as câmaras colocadas no elemento terminal e a olhar para o objecto.

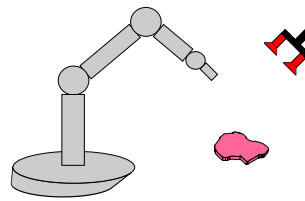


Figura 3.4: Visão estéreo com as câmaras a olhar para o objecto e para o robô manipulador.

reconstrução tri-dimensional, pois a distância entre os eixos ópticos das duas câmaras, *baseline* [131], deste tipo de sistemas deverá ser pequena.

### Câmaras a olhar o robô manipulador e o objecto

Quando as câmaras do par estéreo são colocadas num local pré-definido no espaço, ver figura 3.4, *eye-to-hand*, os sistemas de visão estéreo são mais utilizados no controlo visual de robôs manipuladores. Tal deve-se ao facto de neste caso não existir restrição à *baseline* do par estéreo que quando colocado no elemento terminal, a condicionava. Consegue-se assim o comprimento necessário na *baseline* do par estéreo para que os resultados da reconstrução 3D sejam precisos.

### 3.2.3 Sistema com Câmaras Redundantes

Quando são utilizadas mais que duas câmaras, os sistemas definem-se como redundantes, pois a partir de um par estéreo (duas câmaras) é possível reconstruir a informação tri-dimensional de um objecto que se encontra no seu campo de visão. A utilização deste

tipo de sistemas permite obter informação adicional sobre o que se está a visualizar [39]. No entanto, o *matching* [39] entre as várias imagens da cena que se está a visualizar é um procedimento complexo, pesado computacionalmente e com custos mais elevados. Como consequência dos factos apresentados a utilização destes sistemas no controlo visual de robôs é bastante rara. Note-se ainda que neste tipo de sistemas também se incluem a combinação de visão estéreo com visão monocular, como é o caso por exemplo de um robô com uma câmara no elemento terminal e um par estéreo a visualizar o robô e o objecto.

Existem vários trabalhos publicados entre os quais se destacam:

- o trabalho proposto em [92], no qual foi desenvolvido um sistema *eye-in-hand* multi-câmara para controlo visual, por forma a concluir que a estabilidade do sistema global é assegurada pela estabilidade de cada sub-sistema;
- o trabalho proposto em [47], no qual são utilizadas três câmaras para determinar a pose do objecto a manipular. Posteriormente é realizado o controlo visual utilizando uma câmara no elemento terminal do robô a olhar o objecto;
- a formulação teórica baseada em sistemas multi-câmaras apresentada em [111];
- o sistema baseado em três câmaras proposto em [80] para manipulação de objectos, tendo sido demonstrado que a utilização de uma câmara *eye-to-hand* melhora a precisão da estimação do objecto, relativamente à utilização de um sistema estéreo *eye-in-hand* com *baseline* pequena.
- o método proposto em [132, 133] para lidar com sistemas com câmaras redundantes, no que respeita à robustez do controlo;
- a utilização de várias câmaras em paralelo para aumentar a frequência de amostragem, [112].

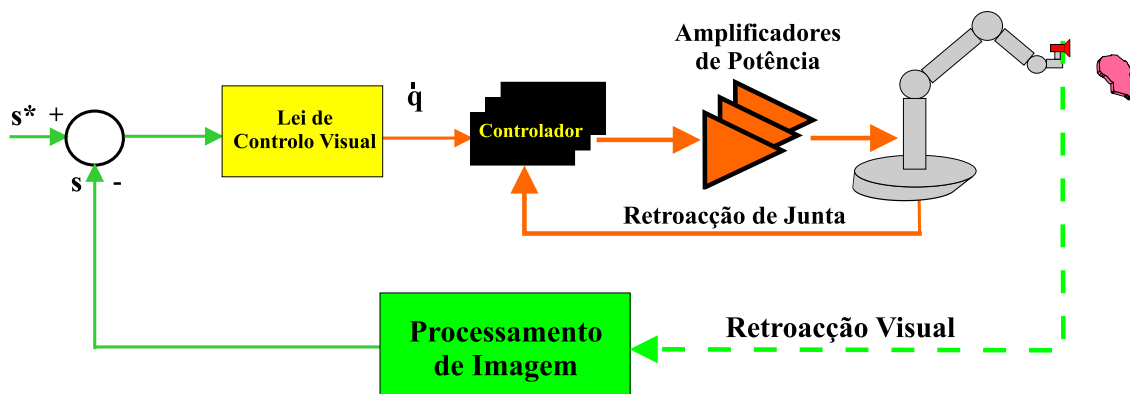


Figura 3.5: Controlo Visual Cinemático.

### 3.3 Controlo Visual Cinemático

O termo "controlo visual de robôs", *visual servoing* [72, 65], surge na literatura para definir o controlo de robôs em que o anel é fechado com a utilização de câmara(s) para extrair informação visual do espaço de trabalho do robô. Assim, o sinal de referência para o anel de controlo é obtido directa ou indirectamente a partir da informação presente no plano da imagem. Através do sinal de referência e da informação vinda da câmara, entretanto já processada, o controlador visual cinemático irá fornecer ao robô uma referência em velocidade para que este vá de encontro ao sinal de referência fornecido. A referência em velocidade pode ser definida em coordenadas de junta, velocidades de junta - *joint velocities*, ou em coordenadas cartesianas, tissor de velocidade - *velocity screw*. O diagrama da figura 3.5 apresenta a estrutura do controlo visual cinemático, onde se pode verificar a existência de um anel de controlo interior realimentado com as velocidades de junta do robô. Nas sub-secções seguintes são apresentadas as estruturas de controlo visual cinemático existentes na literatura, nomeadamente: controlo visual cinemático baseado na imagem, controlo visual cinemático baseado em posição e controlo visual híbrido. Existe ainda na literatura [31, 110] o denominado controlo visual baseado em movimento, onde se controlam os movimentos da câmara de forma a ir de encontro a um campo de velocidades desejado na imagem. Este tipo de controlo visual é baseado na medição do fluxo óptico entre duas imagens consecutivas durante o controlo, o que permite realizar o controlo sem

qualquer conhecimento a priori do objecto.

### 3.3.1 Controlo Visual Cinemático baseado na imagem

No controlo visual cinemático baseado na imagem, as características de um determinado objecto são extraídas directamente da imagem, sem que para tal seja necessário interpretar a imagem, ver figura 2.1. Apenas são utilizados algoritmos para extracção de características do(s) objecto(s) na imagem definidas, por exemplo, através das seguintes primitivas [20, 38]: pontos, linhas e elipses. Um exemplo mais recente de características do objecto na imagem são os seus momentos [129], sendo que os mais utilizados em toda a literatura são os pontos.

Durante o controlo é calculada a diferença entre as características do objecto na imagem desejadas e as actuais, sendo que esta diferença é a responsável pela acção de controlo que irá mover o robô para a posição desejada, ver figura 3.6. Numa primeira análise e face ao já exposto esta arquitectura de controlo tem as seguintes características:

- não necessita da fase de interpretação da imagem, necessária à reconstrução tridimensional do objecto, possibilitando a diminuição do tempo de amostragem do controlo de visão e evitando os problemas de visão por computador inerentes (calibração, sensibilidade ao ruído na imagem e incertezas no modelo do objecto);
- não é necessário o cálculo da cinemática directa do robô [113], apenas o jacobiano do robô é necessário e este pode ainda estar dentro do anel interno do robô, separando-se assim os problemas de singularidades.

A acção de controlo atrás referida, tursor de velocidade ou velocidade de juntas, é determinada através do erro entre as características do objecto na imagem, com base numa matriz que estabelece o modelo de interacção. Esta matriz é definida na presente dissertação como jacobiano da imagem, *image jacobian* [62, 72]. Também é designada na

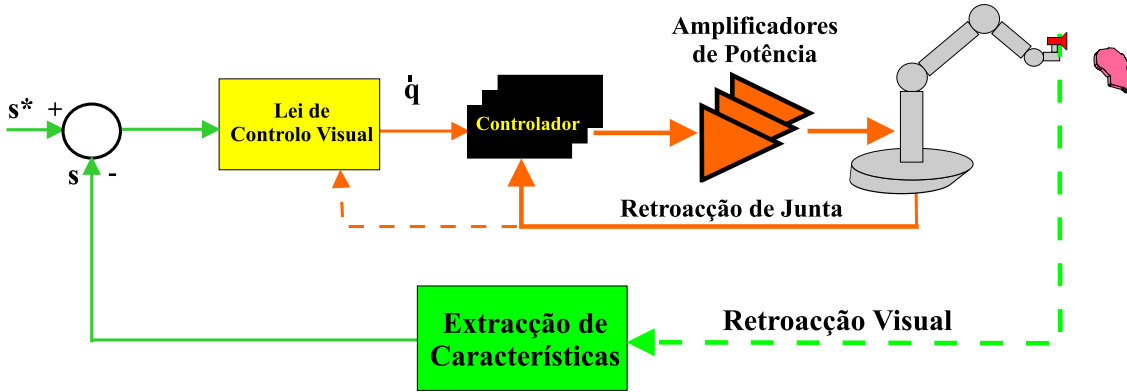


Figura 3.6: Controlo Visual Cinemático baseado na imagem.

literatura como *feature jacobian* [41], *feature sensitivity matrix* [75], *interaction matrix* [23].

Considerando o vector que representa as características do objecto na imagem  $\mathbf{s}$ , e um segundo vector que representa a posição e orientação da câmara,  $\mathbf{r}_c$ , o jacobiano da imagem  $\mathbf{J}_i$  é definido por:

$$\mathbf{J}_i = \frac{\partial \mathbf{s}}{\partial \mathbf{r}_c} \quad (3.1)$$

O vector  $\mathbf{r}_c$ , tem dimensão  $(6 \times 1)$  uma vez que as três primeiras coordenadas referem-se à posição da câmara e as restantes três à sua orientação. Usualmente a orientação é definida pelos ângulos de rotação em torno dos eixos coordenados do referencial da câmara. O vector das características do objecto na imagem  $\mathbf{s}$ , não tem dimensão fixa e depende do problema e de como ele é abordado, i.e. a escolha das características do objecto na imagem e do seu número. Assim a matriz jacobiana da imagem pode ou não ser quadrada.

Apresenta-se a seguir o cálculo da matriz jacobiana da imagem para o caso descrito anteriormente de visão monocular com a câmara a olhar o objecto. Considerando um ponto genérico no espaço cartesiano,  $\mathbf{x}_{3D}$ :

$$\mathbf{x}_{3D} = [ X \quad Y \quad Z ]^T \quad (3.2)$$

A velocidade do ponto  $\mathbf{x}_{3D}$ , relativamente a um referencial em movimento, com velocidade  $\mathbf{v}$  é dada por:

$$\dot{\mathbf{x}}_{3D} = [ -\mathbf{I}_3 \quad S(\mathbf{x}_{3D}) ] \cdot \mathbf{v} \quad (3.3)$$

Considere um ponto  $\mathbf{m}$ , na imagem em coordenadas métricas,

$$\mathbf{m} = [x \ y]^T = \left[ \frac{X}{Z} \ \frac{Y}{Z} \right]^T \quad (3.4)$$

Calculando a derivada do ponto  $\mathbf{m}$  em relação ao tempo, obtém-se a sua velocidade,

$$\dot{\mathbf{m}} = [\dot{x} \ \dot{y}]^T = \begin{bmatrix} \frac{\dot{X} \cdot Z - X \cdot \dot{Z}}{Z^2} \\ \frac{\dot{Y} \cdot Z - Y \cdot \dot{Z}}{Z^2} \end{bmatrix} = \frac{1}{Z} \cdot \begin{bmatrix} 1 & 0 & -\frac{X}{Z} \\ 0 & 1 & -\frac{Y}{Z} \end{bmatrix} \cdot \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (3.5)$$

$$\dot{\mathbf{m}} = -\mathbf{J}_v(\mathbf{m}) \cdot \dot{\mathbf{x}}_{3D} \quad (3.6)$$

$$\mathbf{J}_v(\mathbf{m}) = \frac{1}{Z} \begin{bmatrix} -1 & 0 & \frac{X}{Z} \\ 0 & -1 & \frac{Y}{Z} \end{bmatrix} \quad (3.7)$$

Sabendo que  $\mathbf{x}_{3D} = Z \cdot \tilde{\mathbf{m}}$  e utilizando as equações (3.3) e (3.6), obtém-se a seguinte relação entre as velocidades do referencial em movimento  $\mathbf{v}$ , e do ponto na imagem :

$$\dot{\mathbf{m}} = -\mathbf{J}_v(\mathbf{m}) \cdot [ -\mathbf{I}_3 \ S(\mathbf{x}) ] \cdot \mathbf{v} \quad (3.8)$$

$$\dot{\mathbf{m}} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{X}{Z^2} & x \cdot y & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{Y}{Z^2} & 1+y^2 & -x \cdot y & -x \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.9)$$

Logo, pode-se concluir que:

$$\dot{\mathbf{m}} = [ \mathbf{J}_v(\mathbf{m}) \ \mathbf{J}_{(v,w)}(\mathbf{m}) ] \cdot \mathbf{v} \quad (3.10)$$

Reescrevendo (3.10) para o caso em que a característica do objecto na imagem é um ponto descrito pelas suas coordenadas métricas  $x$  e  $y$ , tem-se a seguinte equação que corresponde à implementação, para este caso de (3.1):

$$\dot{\mathbf{s}} = \mathbf{J}_i \cdot {}^c \dot{\mathbf{r}}_C = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & x \cdot y & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -x \cdot y & -x \end{bmatrix} \cdot {}^c \dot{\mathbf{r}}_C \quad (3.11)$$

Como se verifica, (3.11) depende do conhecimento da coordenada  $Z$  do ponto, i.e. a profundidade, distância entre a câmara e o objecto medida sobre o eixo óptico desta. Esta coordenada não pode ser medida directamente de uma imagem (visão monocular). No caso de visão estéreo a profundidade pode ser calculada através de técnicas de reconstrução 3D, descritas no capítulo 2. No caso de visão monocular existem três opções possíveis para determinar  $Z$ , [21]:

- considerar durante toda a duração do controlo visual o seu valor na posição desejada das características do objecto na imagem, i.e. o seu valor é constante e igual a  $Z^*$ ;
- estimar numericamente quer o seu valor ou mesmo o jacobiano da imagem (3.11), durante o controlo visual, sem utilizar o respectivo modelo, ver secção 3.5;
- determinar a profundidade utilizando o modelo 3D do objecto, capítulo 2, ou através do movimento da câmara [79, 118].

todas elas com as suas vantagens e desvantagens, [21].

Como se pode verificar em (3.11) a matriz jacobiana da imagem,  $\mathbf{J}_i$ , não é quadrada ( $2 \times 6$ ). No caso apresentado existem seis graus de liberdade a controlar, os seis componentes da velocidade  ${}^c\mathbf{r}_c$ , assim são necessárias pelo menos três pontos na imagem para realizar o controlo. Ao serem utilizados três pontos como características a matriz jacobiana é quadrada e a sua inversa pode ser calculada, desde que esta não seja singular, sendo que em [98] é apresentado um estudo sobre as singularidades no espaço euclideano para um robô de seis graus de liberdade. Em [53] o autor apresentou as singularidades cinemáticas para um robô planar (anexo B) e para várias configurações possíveis das câmaras. Quando o número de características do objecto na imagem for maior que o número de graus de liberdade a controlar, estamos perante o caso da existência de características redundantes. Em [63] é apresentado um estudo, validado experimentalmente, que afirma que ao serem



utilizadas características redundantes a rapidez de convergência e a precisão do posicionamento da câmara melhoram consideravelmente. O jacobiano da imagem para apenas um ponto na imagem é descrito em (3.11), e caso se pretenda utilizar mais pontos basta somente inserir as respectivas linhas na matriz.

O anel interno do robô, descrito na figura 3.5, usualmente recebe da lei de controlo de visão a velocidade do elemento terminal do robô  ${}^e\dot{\mathbf{r}}_e$ , ou as suas velocidades de junta  $\dot{\mathbf{q}}$ . Para colocar o tursor de velocidade expresso no referencial da câmara,  ${}^c\dot{\mathbf{r}}_c$ , no referencial do elemento terminal é necessário utilizar a matriz de transformação  $\mathbf{W}$ , como descrita em (3.12).

$${}^c\dot{\mathbf{r}}_c = {}^c\mathbf{W}_e \cdot {}^e\dot{\mathbf{r}}_e \quad (3.12)$$

A matriz de transformação,  ${}^c\mathbf{W}_e$ , contém a translacção,  ${}^c\mathbf{t}_e$ , e a rotação,  ${}^c\mathbf{R}_e$ , entre o referencial da câmara e o referencial do elemento terminal:

$${}^c\mathbf{W}_e = \begin{bmatrix} \mathbf{I}_3 & S({}^c\mathbf{t}_e) \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} {}^c\mathbf{R}_e & \mathbf{0}_3 \\ \mathbf{0}_3 & {}^c\mathbf{R}_e \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_e & S({}^c\mathbf{t}_e) \cdot {}^c\mathbf{R}_e \\ \mathbf{0}_3 & {}^c\mathbf{R}_e \end{bmatrix} \quad (3.13)$$

A introdução da matriz de transformação,  ${}^c\mathbf{W}_e$ , obriga, à estimação dos parâmetros da matriz de rotação,  ${}^c\mathbf{R}_e$ , e do vector de translação  ${}^c\mathbf{t}_e$ , obtidos manualmente.

No caso do anel interno receber directamente as velocidades de junta,  $\dot{\mathbf{q}}$ , é necessário conhecer a relação entre estas e as velocidades cartesianas do elemento terminal  $\mathbf{r}_e$ . Tal relação é descrita em (3.14) em que o jacobiano do robô,  $\mathbf{J}_R$ , é calculado no referencial do elemento terminal:

$${}^e\dot{\mathbf{r}}_e = {}^e\mathbf{J}_R(\mathbf{q}) \cdot \dot{\mathbf{q}} \quad (3.14)$$

A relação existente entre as características do objecto na imagem e as coordenadas do elemento terminal assim como as coordenadas de junta é descrita, respectivamente, em (3.15) e em (3.16). A matriz jacobiana para o controlo visual baseado na imagem é definida em (3.17). Estas equações serão utilizadas posteriormente para construir a lei de

controlo visual.

$$\dot{\mathbf{s}} = \mathbf{J}_i \cdot {}^c\mathbf{W}_e \cdot {}^e\dot{\mathbf{r}}_e \quad (3.15)$$

$$\dot{\mathbf{s}} = \mathbf{J}_i \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R \cdot \dot{\mathbf{q}} \quad (3.16)$$

$$\dot{\mathbf{s}} = \mathbf{J}_{2D} \cdot \dot{\mathbf{q}} \quad (3.17)$$

O esquema de controlo visual clássico na imagem, foi apresentado em [38], e baseia-se na regulação a zero do erro entre as características da imagem na posição desejada e na posição actual. A estrutura global de controlo é apresentada na figura 3.6, onde no bloco *Controlador* se encontra implementado um controlador PD [4], que funciona com um tempo de amostragem de 1 [ms].

As velocidades de junta,  $\dot{\mathbf{q}}$ , necessárias ao movimento do robô para a posição pré-definida das características da imagem,  $\mathbf{s}^*$ , são geradas no bloco *Lei de Controlo Visual*. Por forma a atingir o objectivo definido atrás, o erro entre as características da imagem desejadas e actuais,  $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$ , deverá ser zero. Na posição desejada a velocidade das características da imagem deverá ser zero,  $\dot{\mathbf{s}}^* = 0$ , garantindo-se assim que o robô manipulador pára o seu movimento. Assim, a velocidade do erro entre as características da imagem e a velocidade das características da imagem, tem a seguinte relação:

$$\dot{\mathbf{e}} = \dot{\mathbf{s}} - \dot{\mathbf{s}}^* = \dot{\mathbf{s}} \quad (3.18)$$

Como sugerido em [38], é imposto um decaimento exponencial do erro das características da imagem durante o controlo:

$$\dot{\mathbf{e}} = -\lambda \cdot \mathbf{e} \quad (3.19)$$

onde  $\lambda$  é um ganho positivo, utilizado para aumentar ou diminuir a velocidade do decaimento referido atrás. Através de (3.16), (3.18) e (3.19), a lei de controlo visual é definida por [53]:

$$\dot{\mathbf{q}} = -\lambda \cdot \hat{\mathbf{J}}_{2D}^{-1} \cdot (\mathbf{s} - \mathbf{s}^*). \quad (3.20)$$

Quando o número de características de imagem for igual ao número de variáveis a controlar, i.e.  $\dim(\mathbf{s}) = \dim(\mathbf{q})$ , a matriz jacobiana  $\mathbf{J}_{2D}$  é quadrada, e a sua inversa poderá ser estimada a cada passo do tempo de amostragem de visão. Quando são utilizadas características de imagem redundantes, i.e.  $\dim(\mathbf{s}) > \dim(\mathbf{q})$ , a pseudo inversa da matriz  $\mathbf{J}_{2D}$  terá que ser calculada.

Durante o controlo, o erro das características da imagem diminui se o decaimento exponencial definido em (3.19) for especificado e se a matriz  $\mathbf{J}_{2D}$  for não-singular [38].

Quando  $\dim(\mathbf{s}) = \dim(\mathbf{q})$ , a seguinte condição deverá ser verificada por forma a assegurar a convergência da lei de controlo de visão [38]:

$$\mathbf{J}_{2D} \cdot \hat{\mathbf{J}}_{2D}^{-1} > 0 \quad (3.21)$$

Como  $\hat{\mathbf{J}}_{2D}^{-1}$  deverá ser calculada a cada tempo de amostragem de visão e depende de  $Z$ , a convergência global do controlo não é verificada. Apenas numa vizinhança da posição desejada poderá ser assegurada convergência local, se o valor de  $Z$  na referida posição for utilizado.

### 3.3.2 Controlo Visual Cinemático baseado em posição

A extracção de características do objecto indirectamente do plano da imagem, usualmente a pose (posição cartesiana) do objecto relativamente ao referencial da câmara ou a qualquer outro referencial conhecido no espaço, é a razão da definição de controlo visual baseado em posição ou ainda controlo visual tri-dimensional (3D). Na figura 3.7 é apresentado o controlo visual cinemático baseado em posição, em que as velocidades de junta do robô são determinadas pela lei de controlo de visão através do erro existente, em cada iteração, entre a pose desejada e a actual.

Este tipo de controlo não é maioritariamente utilizado pois tem duas desvantagens fundamentais face ao controlo visual baseado na imagem. A primeira desvantagem surge devido

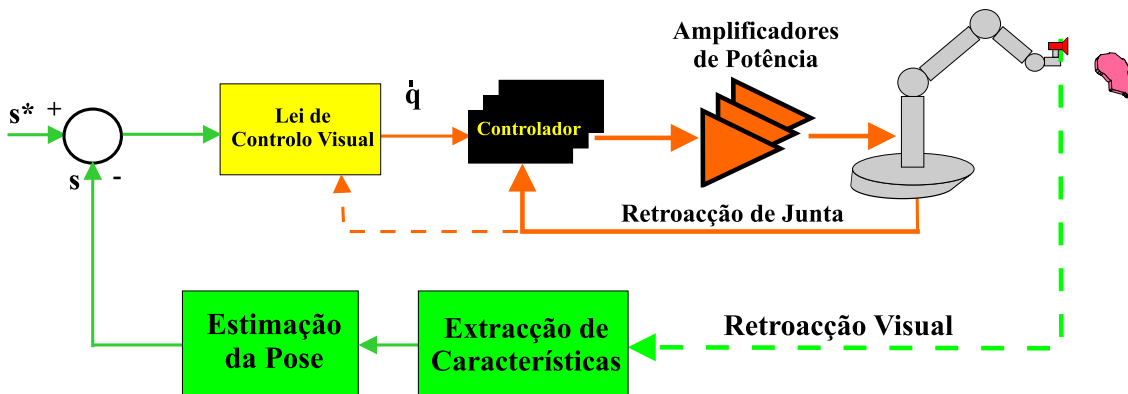


Figura 3.7: Controlo Visual Cinemático baseado em posição.

à necessidade da estimação da pose do objecto relativamente à câmara, requerendo para isso informação adicional sobre o modelo do objecto. A segunda desvantagem advém da necessidade da calibração da câmara e da ligação câmara/robô, para garantir um posicionamento perfeito do sistema. Note-se que uma calibração precisa apenas é necessária se a posição desejada for expressa em coordenadas cartesianas. Se a posição desejada for determinada ensinando o sistema, i.e. movendo o robô para a posição desejada e posteriormente estimando a pose, como o caso de controlo visual baseado na imagem, não é necessária uma calibração precisa, [95]. A principal vantagem do controlo visual baseado em posição é a separação entre as fases de interpretação da imagem adquirida e o controlo propriamente dito, i.e. o controlo é realizado unicamente com informação tri-dimensional obtida através da imagem.

De seguida é apresentado o modelo para a interacção entre o objecto (caracterizado pela sua pose relativamente à câmara) e a câmara, i.e. o jacobiano da imagem. Saliente-se que na presente sub-secção é pressuposta a obtenção de uma estimação perfeita da referida pose, através dos métodos referidos no capítulo 2. Como definido atrás, a pose do objecto é caracterizada pela rotação e pela translação existentes entre o referencial do objecto e o referencial da câmara,  ${}^c\mathbf{T}_o$ . Esta característica pode ser utilizada para o caso em que a câmara se encontra colocada no espaço a olhar o robô e o objecto, ou ainda quando se encontra no elemento terminal do robô a olhar o objecto. Na figura 3.8 é apresentada

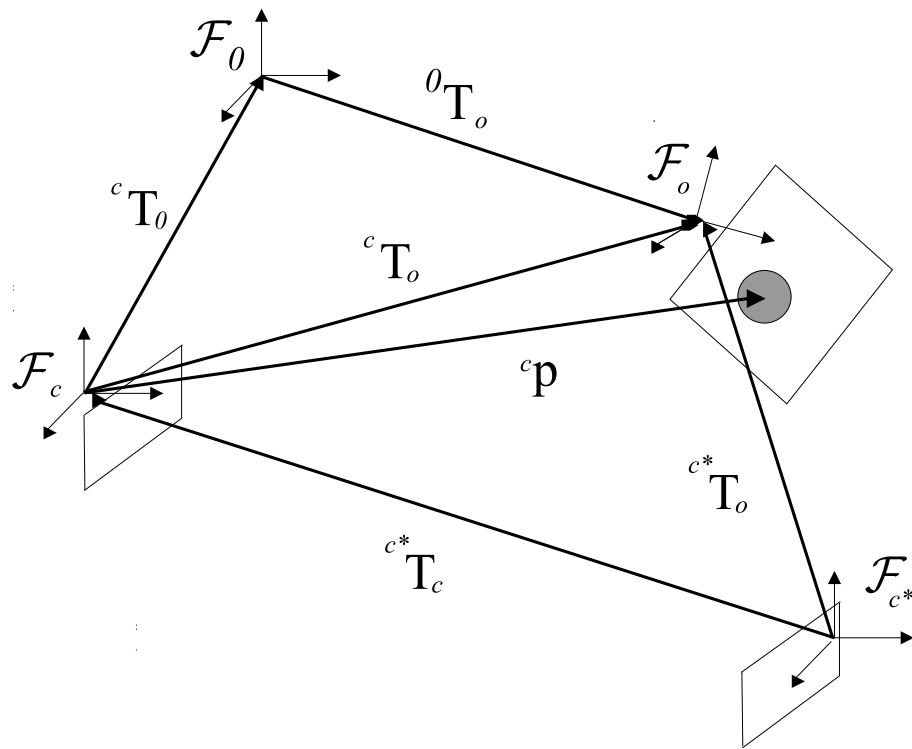


Figura 3.8: Características do objecto na imagem para controlo visual baseado em posição.

uma outra possibilidade para escolha de características de imagem baseadas na pose, i.e. a pose actual da câmara relativamente à sua posição desejada,  ${}^{c^*}\mathbf{T}_c$ , e que pode ser utilizada quando a câmara se encontra no elemento terminal do robô a olhar o objecto. Devido ao maior leque de aplicação e ao facto de garantir uma trajectória linear no espaço cartesiano, [19, 132], foi utilizada nesta tese a característica,  ${}^c\mathbf{T}_o$ , muito embora ambas as características definidas anteriormente tenham comportamentos diferentes durante o controlo e também em função da posição da câmara. Estas diferenças foram abordadas por Cervera e Martinet em [19].

$${}^c\mathbf{T}_o = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \Rightarrow \mathbf{s} = ( {}^c\mathbf{t}_o^T \quad {}^c\mathbf{u}_o^T \theta )^T \quad (3.22)$$

em que a rotação  ${}^c\mathbf{R}_o$  é expressa pelo vector  ${}^c\mathbf{u}_o\theta$ , conforme o anexo A.1.1. Note-se que para representar a rotação podia-se ainda escolher os ângulos de Euler, mas optou-se pela

primeira devido a também esta representação ser a utilizada para o caso do controlo visual  $2\frac{1}{2}D$ , ver sub-secção 3.3.3. Considerando o caso em que uma câmara se encontra a olhar para o objecto, o jacobiano entre a velocidade das características da imagem, pose do objecto relativamente à câmara, e as velocidades de junta é determinado por duas fases. A primeira relativamente à translacção e a segunda à rotação.

O jacobiano da imagem, relativamente à translacção  ${}^c\mathbf{t}_o$ , é definido por:

$${}^c\dot{\mathbf{t}}_o = \begin{bmatrix} -\mathbf{I}_3 & S({}^c\mathbf{t}_o) \end{bmatrix} \cdot \mathbf{v} \quad (3.23)$$

Considerando a parte da rotação, definida pelo vector  ${}^c\mathbf{u}_o\theta$  :

$${}^c\dot{\mathbf{u}}_o\theta = \begin{bmatrix} \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \cdot \mathbf{v} \quad (3.24)$$

em que [87] ,

$$\mathbf{L}_\omega({}^c\mathbf{u}_o, \theta) = \mathbf{I}_3 - \frac{\theta}{2} \cdot \mathbf{S}({}^c\mathbf{u}_o) + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2(\frac{\theta}{2})}\right) \cdot (\mathbf{S}({}^c\mathbf{u}_o))^2 \quad (3.25)$$

onde,

$$\text{sinc}(\theta) = \begin{cases} 1 & \text{se } \theta = 0 \\ \frac{\sin(\theta)}{\theta} & \text{se } \theta \neq 0 \end{cases} \quad (3.26)$$

O jacobiano global utilizado para o controlo visual cinemático baseado em posição,  $\mathbf{J}_{3D}$ , para a característica da imagem  ${}^c\mathbf{T}_o$ , é assim definido por:

$$\dot{\mathbf{s}} = \begin{pmatrix} {}^c\dot{\mathbf{t}}_o \\ {}^c\dot{\mathbf{u}}_o\theta \end{pmatrix} = \begin{bmatrix} -\mathbf{I}_3 & S({}^c\mathbf{t}_o) \\ \mathbf{0}_3 & \mathbf{L}_\omega \end{bmatrix} \cdot \mathbf{v} = \mathbf{J}_{3D} \cdot \mathbf{v} \quad (3.27)$$

Utilizando metodologia análoga à utilizada para a derivação da lei de controlo visual baseada na imagem, temos que a lei de controlo visual baseada em posição é dada por:

$$\dot{\mathbf{q}} = -\lambda \cdot (\hat{\mathbf{J}}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^{-1} \cdot (\mathbf{s} - \mathbf{s}^*) . \quad (3.28)$$

### 3.3.3 Controlo Visual Cinemático Híbrido

O controlo visual baseado na imagem, como visto anteriormente, é robusto a erros na modelação da cinemática do robô, da câmara e ainda da forma do objecto. Contudo o seu domínio de estabilidade é local [64], existindo ainda na literatura um exemplo de instabilidade para um movimento de rotação de  $180^\circ$  em torno do eixo óptico da câmara [21]. Com o intuito de solucionar o problema dos mínimos locais, a comunidade científica tem procurado novas metodologias para o controlo visual baseadas na construção de modelos analíticos. Estas novas metodologias englobam características do controlo visual baseado na imagem e do controlo baseado em posição, como tal são denominadas na literatura como controlo visual híbrido, [30, 60].

A primeira metodologia de controlo visual híbrido foi proposta por Malis [90, 91], controlo visual  $2\frac{1}{2}$  D, e é baseada na utilização de seis características do objecto na imagem. As duas primeiras dizem respeito a um ponto do objecto na imagem, a terceira refere-se à profundidade  $Z$ , e as três coordenadas seguintes referem-se à rotação entre a posição actual e desejada da câmara,  ${}^c\mathbf{R}_c$ . Em seguida é deduzido o jacobiano da imagem para esta metodologia.

Incluindo artificialmente, na terceira coordenada do ponto  $\mathbf{m}$ , definido na equação (3.4), a variável  $z = \ln(Z)$ , obtém-se o vector de coordenadas métricas estendido,  $\mathbf{m}_e$ :

$$\mathbf{m}_e = [x \ y \ z]^T = \left[ \frac{X}{Z} \ \frac{Y}{Z} \ \ln(Z) \right]^T \quad (3.29)$$

Calculando a derivada do ponto  $\mathbf{m}_e$  em relação ao tempo, obtém-se a sua velocidade:

$$\dot{\mathbf{m}}_e = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T = \begin{bmatrix} \frac{\dot{X}\cdot Z - X\cdot\dot{Z}}{Z^2} \\ \frac{\dot{Y}\cdot Z - Y\cdot\dot{Z}}{Z^2} \\ \frac{\dot{Z}}{Z} \end{bmatrix} = \frac{1}{Z} \cdot \begin{bmatrix} 1 & 0 & -\frac{X}{Z} \\ 0 & 1 & -\frac{Y}{Z} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} \quad (3.30)$$

$$\dot{\mathbf{m}}_e = -\mathbf{L}_v(\mathbf{m}_e) \cdot \dot{\mathbf{x}} \quad (3.31)$$

Utilizando as equações (3.3) e (3.31), obtém-se a seguinte relação entre as velocidades do referencial em movimento  $\mathbf{v}$ , onde está colocada a câmara, e do ponto na imagem :

$$\dot{\mathbf{m}}_e = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & x \cdot y & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -x \cdot y & -x \\ 0 & 0 & -\frac{1}{Z} & y & x & 0 \end{bmatrix} \cdot \mathbf{v} \quad (3.32)$$

em que  $Z$  é calculado utilizando as técnicas apresentadas na sub-secção 2.3.1, através da expressão  $Z = \rho_1 \cdot d_1^*$ , em que  $\rho_1 = \frac{\det(\mathbf{H})}{\mathbf{n}^T \cdot \mathbf{m}}$  e  $\mathbf{n} = \mathbf{R} \cdot \mathbf{n}^*$ . Note-se que a matriz de homografia  $\mathbf{H}$  refere-se à transformação entre a posição da câmara na posição actual e a posição desejada. A matriz de rotação  $\mathbf{R}$  e a normal  $\mathbf{n}^*$  são obtidas através da referida matriz de homografia. A distância  $d_1^*$  é obtida manualmente através do posicionamento do robô na posição desejada, não sendo necessária um grande precisão no valor estimado uma vez que este tem uma pequena influência na estabilidade do sistema [91].

Logo, pode-se concluir que:

$$\dot{\mathbf{m}}_e = \begin{bmatrix} \mathbf{L}_v(\mathbf{m}_e) & \mathbf{L}_{(v,\omega)}(\mathbf{m}_e) \end{bmatrix} \cdot \mathbf{v} \quad (3.33)$$

É ainda necessário incluir o vector  ${}^{c^*}\mathbf{u}_c\theta$  que representa a rotação, de forma análoga ao expresso na equação (3.24) para assim obter a matriz jacobiana de imagem para controlo visual  $2\frac{1}{2}D$ ,  $\mathbf{J}_{2\frac{1}{2}D}$ :

$$\dot{\mathbf{s}} = \begin{pmatrix} \dot{\mathbf{m}}_e \\ {}^{c^*}\dot{\mathbf{u}}_c\theta \end{pmatrix} = \begin{bmatrix} \mathbf{L}_v(\mathbf{m}_e) & \mathbf{L}_{(v,\omega)}(\mathbf{m}_e) \\ 0 & \mathbf{L}_\omega({}^{c^*}\mathbf{u}_c, \theta) \end{bmatrix} \cdot \mathbf{v} = \mathbf{J}_{2\frac{1}{2}D} \cdot \mathbf{v} \quad (3.34)$$

O vector de erro é definido da seguinte forma [91]:

$$\mathbf{e} = \begin{pmatrix} x - x^* \\ y - y^* \\ \ln\left(\frac{Z}{Z^*}\right) \\ {}^{c^*}\mathbf{u}_c\theta \end{pmatrix} \quad (3.35)$$

em que  $\frac{Z}{Z^*} = \rho_1 \cdot \mathbf{n}^{*T} \cdot \mathbf{m}^*$ , [91], é calculado através da matriz de homografia  $\mathbf{H}$ .



Utilizando metodologia análoga à utilizada para a derivação das leis de controlo visual baseada na imagem e em posição, temos que a lei de controlo visual  $2\frac{1}{2}D$  é dada por:

$$\dot{\mathbf{q}} = -\lambda \cdot (\hat{\mathbf{J}}_{2\frac{1}{2}D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^{-1} \cdot \mathbf{e} \quad (3.36)$$

Após o aparecimento deste primeiro método que desacopla o movimento de translacção e de rotação da câmara, ver (3.34), e que tem como característica uma convergência global para as características do objecto na imagem desejadas, foram introduzidos outros métodos. Em [32] foi proposto por Deguchi um outro método que desacopla também a translacção e rotação para o caso do controlo visual baseado na imagem, tendo sido discutida a trajectória da câmara, obtida através da lei de controlo. Um outro método para separar os movimentos de rotação e de translacção do eixo  $Z$  dos restantes eixos do movimento do referencial da câmara, foi proposto por Corke e Hutchinson [29, 30]. Os três métodos anteriores, em conjunto com o controlo visual baseado na imagem foram objecto de um estudo comparativo apresentado em [43].

Recentemente estão a ser estudados controladores baseados em sistemas híbridos (*Hybrid Switched Systems*) para o controlo visual de robôs. Estes sistemas permitem comutar entre controladores durante a execução de uma determinada tarefa, para assim utilizar as melhores características de cada controlador em determinadas situações de utilização. Exemplos da aplicação de sistemas híbridos podem ser encontrados em [26, 44, 45, 64, 94].

### 3.4 Controlo Visual Dinâmico

O controlo visual de robôs manipuladores tem vindo a ser abordado pela comunidade científica no que respeita somente à cinemática do movimento do robô (controlo visual cinemático). Como descrito na secção 3.3 é assumido que a velocidade do elemento terminal ou as velocidades de junta do robô são calculadas através de uma lei de controlo visual. Estas velocidades são posteriormente utilizadas como referência de velocidade

para um controlo de velocidade interno, que é responsável pelo movimento do robô. A lei de controlo visual é determinada considerando que o robô é um posicionador perfeito. Conclui-se portanto que este tipo de controlo visual, cinemático, não tem em linha de conta a dinâmica do robô, não sendo assim a melhor solução para movimentos rápidos deste. Considerando a dinâmica do robô, é possível projectar outro tipo de controladores que calculam directamente o binário a fornecer aos motores das juntas do robô, por forma a que este realize uma determinada tarefa utilizando informação visual. Este tipo de controlo será definido como Controlo Visual Dinâmico em contraponto com o Controlo Visual Cinemático, descrito na secção 3.3. Três razões para a utilização de controlo visual cinemático na esmagadora maioria das soluções propostas e implementadas na prática foram apresentadas em [72]:

1. O elevado tempo de amostragem da retroacção do anel de visão, torna o controlo visual dinâmico num problema bastante complexo;
2. A maioria dos robôs industriais possui uma *interface* capaz de receber velocidades ou incrementos de posição cartesianas, que simplifica a implementação prática de sistemas de controlo visual e os torna mais portáteis;
3. Existe separação entre as singularidades cinemáticas do robô e as singularidades da lei de controlo de visão, sendo as primeiras tratadas no anel interno de controlo do robô, fornecido pelo fabricante.

Na figura 3.9 é apresentado o diagrama de blocos referente ao Controlo Visual Dinâmico. Em [72] este tipo de controlo é também definido como Controlo Visual Directo, *Direct Visual Servo*, pois os binários das juntas do robô são directamente calculados pela lei de controlo visual.

A utilização de controlo visual do tipo dinâmico tem vindo a ser proposta pela comunidade científica. O seu desenvolvimento tem vindo a aumentar, devido ao facto de o tempo de amostragem do anel de visão ter vindo a diminuir e também à procura de melhorar

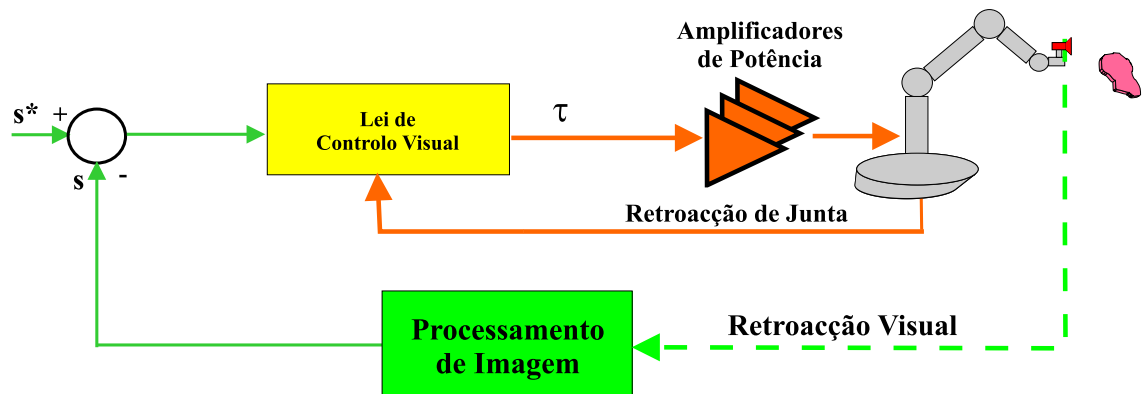


Figura 3.9: Controlo Visual Dinâmico.

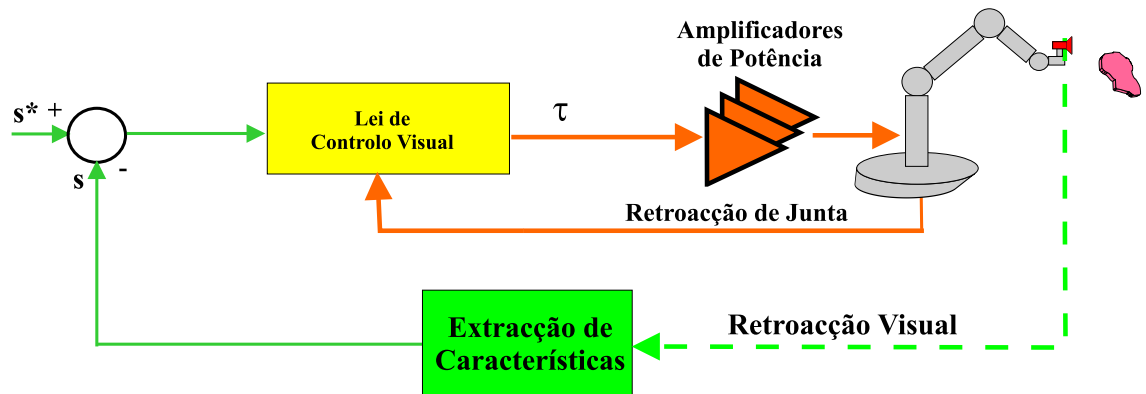


Figura 3.10: Controlo Visual Dinâmico baseado na imagem.

as performances dinâmicas do controlo visual de robôs manipuladores. À semelhança do controlo visual cinemático, o controlo visual dinâmico também pode ser definido na imagem ou em posição, como se pode verificar nas figuras 3.10 e 3.11, respectivamente.

Várias soluções têm sido propostas ao longo dos últimos anos baseadas no controlo visual dinâmico. Hashimoto em [61] derivou uma lei de controlo óptima, que aplicou a um robô manipulador planar e a um robô PUMA, ambos na configuração *eye-in-hand*. Kelly em [76] projectou um controlador utilizando o jacobiano transposto do robô, (*the Jacobian transpose approach*) [113], aplicado a um robô planar na configuração *eye-to-hand*, em que é necessário determinar a orientação da câmara relativamente ao referencial do mundo. Lefeberg [82] estendeu o trabalho anterior de Kelly, aplicando técnicas adaptativas para solucionar o cálculo da orientação da câmara. Zergeroglu [136] projectou um controlador

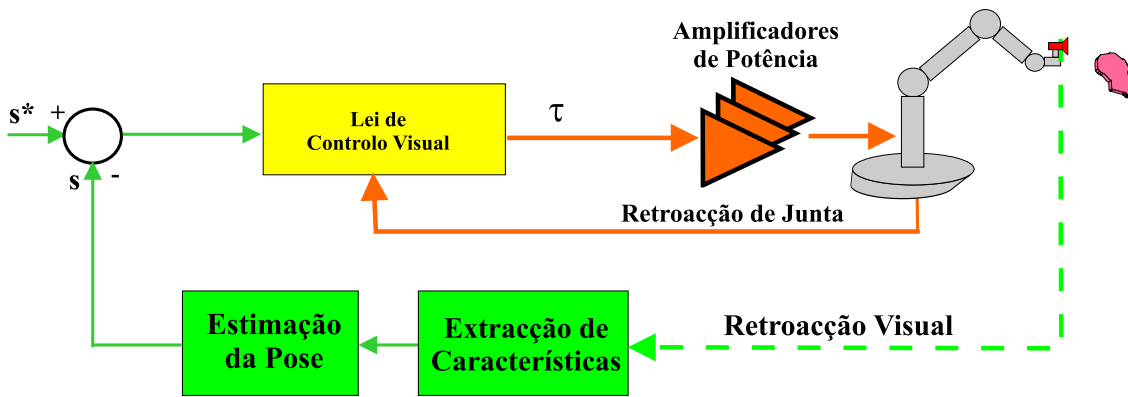


Figura 3.11: Controlo Visual Dinâmico baseado em posição.

baseado em técnicas *back-stepping* para uma robô manipulador planar em ambas configurações *eye-in-hand* e *eye-to-hand*. Cheah [24] propôs um controlador, utilizando o jacobiano transposto do robô, para o caso em que o vector das forças gravíticas do robô é desconhecido. Em [42] é aplicado um controlador óptimo, *inverse optimal H control*, a um robô manipulador planar na configuração *eye-in-hand*. Kelly, em [77] propôs um novo controlador a ser aplicado a um robô manipulador na configuração *eye-in-hand*. Outros controladores dinâmicos foram também propostos para aplicação em robótica móvel [108, 109].

No capítulo 4 é apresentado um novo controlador dinâmico baseado em características de posição, i.e. a pose tri-dimensional do objecto relativamente ao referencial da câmara. Este controlador foi introduzido pelo autor em [56]. No mesmo capítulo é também apresentada a extensão do controlador proposto por Kelly [76], para o caso *eye-in-hand* [54].

### 3.5 Controlo Visual por estimação do modelo de interacção

O modelo de interacção entre as características do objecto na imagem e o robô, foi obtido nas sub-secções anteriores de uma forma analítica, i.e. foi possível obter relações

matemáticas entre o movimento das referidas características e das juntas do robô, assim como os respectivos parâmetros do modelo. Como visto nas secções anteriores, a escolha das características tem um papel preponderante na modelação da interacção atrás referida, i.e. na obtenção a priori do jacobiano da imagem para ser utilizado no controlo visual do robô manipulador.

Para que a abordagem baseada no conhecimento a priori do modelo analítico possa ser fiável é necessário conhecer o sistema na sua totalidade ou com algumas incertezas entretanto quantificadas na literatura [87, 89], nomeadamente o modelo da câmara, o modelo da cinemática do robô e mais importante o jacobiano de imagem. Concluiu-se portanto que ao se realizar o controlo visual com este tipo de modelos, a sua performance não é degradada com a presença de erros até um valor limite, conhecido. Relativamente aos parâmetros intrínsecos, estes podem ser estimados durante o controlo, como proposto em [25]. No entanto, em [88] Malis evoluiu este conceito de incertezas no modelo, pois introduziu o controlo visual invariante aos parâmetros intrínsecos da câmara.

Como é conhecido, a calibração de robôs e de câmaras é um trabalho complexo e moroso, que requer equipamentos e algoritmos especializados [2, 107]. Para ultrapassar os factos descritos e caso não se pretenda ou não seja possível obter a priori o modelo de interacção, é necessário utilizar métodos de estimação quer através de modelos ou por aprendizagem do próprio modelo de interacção. Estes métodos de estimação podem ser utilizados antes ou durante a execução do controlo visual. A estimação do modelo de interacção também é utilizada quando o objecto a tratar é bastante complexo e difícil de caracterizar através da informação visual referida na secção 2.4. O controlo visual de robôs baseado nestes métodos tem sido denominado na literatura como não-calibrado, *uncalibrated visual servoing* [60].

No caso da obtenção do modelo de estimação através de modelos, as características do objecto na imagem utilizadas em todos os trabalhos apresentados pela comunidade científica referidos neste texto para estimar o modelo de interacção, são pontos do objecto na imagem. Quando se utilizam outros tipos de características os métodos falham devido

a que os modelos de estimação estão baseados em pontos. O estimador mais utilizado para o jacobiano é o de *Broyden*, tendo sido aplicado por Jägersand [73], Peipmeier [101, 102] e Hosoda [69, 70]. Todos os métodos referenciados estimam o jacobiano apenas na direcção da posição desejada das características da imagem, i.e. não é estimado o jacobiano em todo o espaço das características da imagem. Em [69] é estimado o modelo de interacção entre o robô e a imagem, tendo como objectivo a convergência assintótica das características da imagem para um valor desejado. Note-se portanto que os parâmetros do modelo estimado podem não convergir para os valores reais. Já em [70] o modelo do robô é conhecido por forma a incorporar um controlo de força na estrutura de controlo visual, em que o jacobiano da imagem é estimado em cada iteração do controlo. Em [73] o problema do controlo visual de robôs é formulado através de um sistema não-linear de optimização por mínimos quadrados, resolvido por um método de Newton utilizando o estimador de Broyden para o jacobiano, para mover o elemento terminal do robô relativamente a um objecto estático. No caso de objecto estar em movimento, em [101, 102] foi utilizada uma abordagem similar. Em [104, 128] são utilizados filtros de Kalman para estimar o jacobiano de imagem

O modelo de interacção pode ainda ser obtido utilizando técnicas de aprendizagem, i.e. através de redes neuronais [17, 78, 99] e redes neuro-fuzzy [126, 127]. A abordagem descrita em [99] não requer conhecimento da cinemática do robô e da velocidade ou posição do objecto relativamente ao robô, no entanto é necessário realizar o treino em todo o espaço de trabalho do robô. Em [17] para determinar a pose do objecto relativamente ao referencial do elemento terminal é utilizada uma rede neuronal, sendo posteriormente realizado o controlo no referido referencial. Esta abordagem necessita também de um treino exaustivo numa fase inicial, i.e. antes de realizado o controlo. A rede neuro-fuzzy utilizada em [126, 127], foi desenvolvida para realizar movimentos rápidos da câmara quando esta está longe da posição desejada, e movimentos mais lentos quando esta já se encontra perto da posição desejada. Esta abordagem necessita também de um treino exaustivo numa fase inicial, *off-line*, i.e. antes de realizado o controlo.

Todos os métodos apresentados nesta secção têm a particularidade de não disponibilizarem um modelo analítico do jacobiano da imagem, como visto nas secções anteriores. Tal facto torna impossível o estudo analítico da estabilidade destes métodos.

O controlo visual de robôs manipuladores através da aprendizagem do modelo de interacção irá ser objecto de estudo detalhado, no capítulo 5. No referido capítulo é proposto um método baseado em lógica *fuzzy* por forma a estimar o modelo inverso da interacção [48, 49, 50] e assim tornar o sistema independente de qualquer processo de calibração da câmara, conhecimento da cinemática do robô e ainda da obtenção do jacobiano da imagem para diferentes características. O sistema é independente da característica do objecto na imagem que se possa escolher, no entanto dever-se-à escolher a(s) característica(s) cuja obtenção seja fácil e rápida.

## 3.6 Planeamento de Trajectórias para Controlo Visual

O planeamento da trajectória das características do objecto na imagem, foi uma solução para o controlo visual de robôs manipuladores apresentada em [97] e que permite melhorar o desempenho do controlo visual cinemático baseado na imagem. A referência (trajectória no plano da imagem) a ser seguida durante o controlo é gerada *off-line*, através das imagens do objecto, inicial e desejada.

O método para gerar a trajectória das características do objecto na imagem é composto por três partes distintas, mas ao mesmo tempo interligadas entre si:

1. a primeira parte é dedicada à inicialização do método, em que a partir da imagem inicial e da imagem desejada é obtida a pose para inicializar o método, i.e. a pose da câmara na sua posição inicial relativamente à posição desejada;

2. a segunda parte consiste num processo iterativo em que é gerado um conjunto de características do objecto na imagem que compõem o caminho a percorrer no plano da imagem;
3. a terceira e última parte consiste em construir uma trajectória ao longo do tempo, composta pelo conjunto de características gerado anteriormente.

Durante a fase de inicialização do método são obtidas as características do objecto nas imagens correspondentes às duas posições da câmara, inicial e desejada. A matriz de rotação e o vector de translação entre o objecto e as posições inicial/final da câmara são calculadas utilizando um método de estimação da pose, como descrito na secção 2.4.2. A inicialização termina com o cálculo da pose inicial relativamente à desejada, através das matrizes de rotação e vectores de translação estimados anteriormente.

O processo iterativo através do qual é gerado o conjunto de características do objecto na imagem que compõem o caminho a percorrer na imagem, inicia-se com a estimação da pose no instante de tempo seguinte. Esta pose  $\Upsilon_{k+1}$ , é calculada através da pose actual,  $\Upsilon_k$ , e uma força composta,  $\mathbf{F}$ .

$$\Upsilon_{k+1} = \Upsilon_k + \varepsilon_k \cdot \frac{\mathbf{F}(\Upsilon_k)}{\|\mathbf{F}(\Upsilon_k)\|}, \quad \varepsilon_k > 0 \quad (3.37)$$

em que  $\varepsilon_k$  é um factor de escala positivo que toma o valor do incremento  $k$ .

$$\mathbf{F}(\Upsilon_k) = \mathbf{F}_a(\Upsilon_k) + \mathbf{F}_r(\Upsilon_k) \quad (3.38)$$

Esta força é composta pela soma de duas forças diferentes, uma atractiva  $\mathbf{F}_a$ , e outra repulsiva  $\mathbf{F}_r$ . A força atractiva minimiza a trajectória e a força repulsiva utilizada na presente tese evita que a trajectória atinja os limites da imagem. Outro tipo de força repulsiva, para evitar limites de junta, é descrita em [97].

Todas as poses seguintes, até se atingir a pose desejada, são estimadas através de um processo recursivo que tem início na pose inicial. A força atractiva toma o seu maior



valor no instante inicial, diminuindo depois até que se anule na pose final. O processo recursivo termina quando a força atractiva toma um valor nulo, significando que a pose actual se encontra na posição desejada. O resultado do processo iterativo é um vector de características do objecto na imagem, que contém os valores das características durante o caminho desde a posição inicial à posição final.

A trajectória final das características do objecto na imagem ao longo do tempo,  $\mathbf{s}^p$ , deverá ser contínua e diferenciável, sendo portanto escolhida uma função de classe  $C^2$ . Esta função é construída interpolando o vector obtido no processo iterativo, utilizando um tempo de amostragem video pré-definido,  $\Delta T = t_k - t_{k-1}$ , e o tempo final desejado para o controlo visual, através de uma spline cúbica.

$$\mathbf{s}^p(t) = \mathbf{A}_k \cdot t^3 + \mathbf{B}_k \cdot t^2 + \mathbf{C}_k \cdot t + \mathbf{D}_k \quad (3.39)$$

em que:  $(k-1) \cdot \Delta T \leq t \leq (k) \cdot \Delta T$ .

A lei de controlo utilizada segue o método original [97]:

$$\dot{\mathbf{q}} = -\lambda \cdot \mathbf{J}_{2D}^{-1} \cdot (\mathbf{e}_p) + \frac{1}{\Delta T} \cdot \mathbf{J}_{2D}^{-1} \cdot (\mathbf{e}_{pp}) \quad (3.40)$$

em que,  $\mathbf{e}_p = \mathbf{s}_k - \mathbf{s}_k^p$ ;  $\mathbf{e}_{pp} = \mathbf{s}_k^p - \mathbf{s}_{k-1}^p$ .



# Capítulo 4

## Controlo Visual Dinâmico baseado em Posição

### 4.1 Introdução

O controlo visual dinâmico tem sido uma alternativa ao controlo visual cinemático, como descrito na secção 3.4. No presente capítulo é apresentada a abordagem desenvolvida nesta tese e que se baseia na pose do objecto relativamente à câmara para obter a lei de controlo visual responsável pelos binários a aplicar ao robô.

A lei de controlo foi inspirada pelo trabalho apresentado por Kelly [76], desenvolvido para um robô planar de dois graus de liberdade e para a configuração *eye-to-hand*. Numa primeira fase deste capítulo foi proposto em [54] (ver secção 4.3.1) uma extensão ao controlador de Kelly para a configuração *eye-in-hand*, sendo também unicamente aplicável a um robô planar de dois graus de liberdade. Para este controlador é necessária a estimação da rotação entre o objecto e a câmara, obtida através do método descrito na sub-secção 2.4.2. Ao procurar evoluir o controlador anterior para o caso mais geral, i.e. um robô de seis graus de liberdade, a solução encontrada foi aplicar o jacobiano de imagem da lei de controlo baseada em posição (ver secção 3.3.2) à classe de controladores baseados no jacobiano transposto [113].

Os controladores apresentados no presente capítulo são todos derivados através do método

directo de Lyapunov [117], que garante a estabilidade dos referidos controladores, como apresentado na secção 4.4.

## 4.2 Definição do Problema de Controlo

Para a definição do problema de controlo visual dinâmico, baseado em posição, é considerado o modelo do robô manipulador (A.25), em que a posição da câmara relativamente ao elemento terminal,  ${}^e\mathbf{T}_c$ , é conhecida. É ainda condição necessária que durante o controlo, o objecto possa ser sempre visualizado na imagem e que a informação sensorial da posição e velocidade das juntas do robô possa ser sempre obtida.

Definindo  $\mathbf{s}^*$  como as características desejadas do objecto na imagem, e  $\mathbf{e}$  o erro entre as características desejadas e actuais,  $\mathbf{e} = \mathbf{s}^* - \mathbf{s}$ . O problema de controlo é definido através do projecto de um controlador com saídas de binário de junta, de forma a que o sistema *eye-in-hand* visualize o objecto numa posição pré-definida na imagem e ao mesmo tempo assegure estabilidade assintótica.

$$\lim_{t \rightarrow \infty} \mathbf{s}(t) = \mathbf{s}^* \Rightarrow \mathbf{e} = 0 \quad (4.1)$$

Por forma a resolver o problema de controlo definido, é necessário que as seguintes três condições se verifiquem:

1. existência de uma configuração das juntas do robô,  $\mathbf{q}^*$ , que permita ao sistema *eye-in-hand* visualizar o objecto numa posição pré-definida na imagem,  $\mathbf{s}^*$ ;
2. para uma configuração desconhecida,  $\mathbf{q}^*$ , o jacobiano  $\mathbf{J}_i \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R$  é não-singular;
3. durante o controlo, o objecto pode ser sempre visualizado na imagem e as suas características extraídas.

## 4.3 Lei de Controlo Visual Dinâmica

### 4.3.1 Extensão do Controlador Dinâmico - Kelly

Em [76] Kelly propôs um controlador assintoticamente estável (4.2) para resolver o problema de controlo descrito na sub-secção 4.2. Este controlador foi projectado especificamente para um robô planar de dois graus de liberdade com uma câmara fixa, colocada a olhar o robô e o objecto, tendo o seu eixo óptico perpendicular ao plano do robô:

$$\tau = \mathbf{J}_R(\mathbf{q})^T \cdot \mathbf{K}_p \cdot \mathbf{R}(\theta)^T \cdot \mathbf{e} - \mathbf{K}_d \cdot \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \quad (4.2)$$

em que  $\mathbf{J}_R$  é o Jacobiano do robô ( $2 \times 2$ ) no referencial do mundo, e  $\mathbf{R}(\theta)$  a matriz de rotação constante ( $2 \times 2$ ) que relaciona os referenciais da câmara e do mundo,  ${}^0\mathbf{R}_c$ . Apesar das restrições descritas anteriormente, esta lei de controlo tem as seguintes vantagens:

- não é necessário conhecer os parâmetros intrínsecos da câmara;
- como parâmetro extrínseco da câmara, apenas é necessário conhecer a rotação;
- não é necessário conhecer a cinemática inversa e o jacobiano inverso do robô.

A anteriormente descrita abordagem de Kelly [76], pode ser estendida para o caso da configuração do sistema *eye-in-hand* [54], sendo necessário ter em consideração que a câmara agora está em movimento, solidária com o robô, i.e. colocada no elemento terminal deste. Assim, a anterior matriz de rotação  $\mathbf{R}(\theta)$  (4.2) terá que ser redefinida para relacionar os referenciais da câmara e do objecto,  ${}^c\mathbf{R}_o$ . Com esta nova abordagem a matriz de rotação deixa de ser constante e deverá ser calculada *on-line*, através de um dos dois métodos descritos em seguida:

- conhecendo a matriz rotação entre os referenciais do mundo e do objecto,  ${}^c\mathbf{R}_o$  pode ser calculado através de

$${}^c\mathbf{R}_o = ({}^0\mathbf{R}_e \cdot {}^e\mathbf{R}_c)^{-1} \cdot {}^0\mathbf{R}_o \quad (4.3)$$

em que  ${}^0\mathbf{R}_e$  é calculada utilizando a cinemática directa do robô, e  ${}^e\mathbf{R}_c$  é conhecida a priori.

- conhecendo o modelo tridimensional do objecto,  ${}^e\mathbf{R}_o$  pode ser estimado através de algoritmos de estimação da pose, descritos no capítulo 2.

A extensão ao algoritmo de Kelly [76], proposta nesta sub-secção, assegura todas as vantagens do método original, mas necessita do conhecimento *on-line* da matriz de rotação devido ao movimento da câmara. Na sub-secção seguinte é apresentado um controlador visual dinâmico baseado em posição, em que o robô e câmara se encontram na configuração *eye-in-hand*. Este último controlador é aplicável a robôs com mais de dois graus de liberdade.

### 4.3.2 Controlo Visual Dinâmico, *eye-in-hand*, baseado em posição

A principal vantagem da utilização do controlo visual baseado em posição, é o desacoplamento entre a interpretação da imagem e o controlador, como referido na sub-secção 3.3.2. Assim, estes dois problemas podem ser abordados separadamente em contraponto com o controlo visual baseado na imagem. Na presente sub-secção é assumido que o problema associado à interpretação da imagem é resolvido utilizando as técnicas abordadas no capítulo 2, de forma a "trabalhar" somente na lei de controlo. Tendo por base o controlador visual clássico baseado em posição, ver a sub-secção 3.3.2, é proposto o controlador visual dinâmico, *eye-in-hand*, baseado em posição. O projecto do controlador referido é baseado no método directo de Lyapunov [117] e na *Jacobian transpose approach* [113].

Seguindo o método directo de Lyapunov, em (4.4) é apresentada uma forma quadrática positiva definida como função candidata de Lyapunov:

$$\mathbf{V}(\dot{\mathbf{q}}, \mathbf{e}) = \frac{1}{2} \cdot \dot{\mathbf{q}}^T \cdot \mathbf{M}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \frac{1}{2} \cdot \mathbf{e}^T \cdot \mathbf{K}_p \cdot \mathbf{e} > 0, \quad \forall \dot{\mathbf{q}}, \mathbf{e} \neq 0 \quad (4.4)$$

em que  $\mathbf{K}_p$  é uma matriz ( $n \times n$ ) positiva definida. O primeiro termo de (4.4) expressa a

energia cinética e o segundo termo a energia potencial relativa ao erro das características do objecto,  $\mathbf{e}$ .

Diferenciando (4.4) em ordem ao tempo e tendo em atenção que o vector das características do objecto na posição desejada  $\mathbf{s}^*$  é constante:

$$\dot{\mathbf{V}}(\dot{\mathbf{q}}, \mathbf{e}) = \dot{\mathbf{q}}^T \cdot \mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \frac{1}{2} \cdot \dot{\mathbf{q}}^T \cdot \dot{\mathbf{M}}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \dot{\mathbf{e}}^T \cdot \mathbf{K}_p \cdot \mathbf{e} \quad (4.5)$$

Utilizando as equações (3.12), (3.14) e (3.27), e sabendo que  $\mathbf{s}^*$  é constante, obtém-se:

$$\dot{\mathbf{e}} = -\mathbf{J}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R \cdot \dot{\mathbf{q}} \quad (4.6)$$

Substituindo (4.6) em (4.5), obtém-se:

$$\dot{\mathbf{V}}(\dot{\mathbf{q}}, \mathbf{e}) = \frac{1}{2} \cdot \dot{\mathbf{q}}^T \cdot \mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \frac{1}{2} \cdot \dot{\mathbf{q}}^T \cdot \dot{\mathbf{M}}(\mathbf{q}) \cdot \dot{\mathbf{q}} - \dot{\mathbf{q}}^T \cdot (\mathbf{J}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^T \cdot \mathbf{K}_p \cdot \mathbf{e} \quad (4.7)$$

Utilizando (A.25) e (A.26), a equação (4.7) pode ser escrita da forma seguinte:

$$\dot{\mathbf{V}}(\dot{\mathbf{q}}, \mathbf{e}) = \dot{\mathbf{q}}^T (\tau - \mathbf{g}(\mathbf{q}) - (\mathbf{J}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^T \cdot \mathbf{K}_p \cdot \mathbf{e}) \quad (4.8)$$

A equação anterior sugere o controlador que pode ser definido através de:

$$\tau = (\mathbf{J}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^T \cdot \mathbf{K}_p \cdot \mathbf{e} + \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \cdot \dot{\mathbf{q}} \quad (4.9)$$

onde o termo  $\mathbf{K}_d \cdot \dot{\mathbf{q}}$  realça o amortecimento do sistema, e  $\mathbf{K}_p$  é uma matriz de ganhos ( $n \times n$ ) positiva definida.

O controlador proposto em (4.9), tem as seguintes características:

- não necessita conhecimento dos parâmetros extrínsecos da câmara;

- é necessário o conhecimento dos parâmetros intrínsecos da câmara;
- pode ser aplicado a um robô com  $n$  graus de liberdade;
- pode ser facilmente extensível à configuração *eye-to-hand* utilizando as técnicas clássicas de controlo visual baseado em posição;
- é assintoticamente estável, como será demonstrado na sub-secção seguinte.

## 4.4 Análise de Estabilidade da Lei de Controlo

Substituindo (4.9) em (4.8), obtém-se a equação:

$$\dot{V}(\dot{\mathbf{q}}, \mathbf{e}) = -\dot{\mathbf{q}}^T \cdot \mathbf{K}_d \cdot \dot{\mathbf{q}} < 0, \quad \forall \dot{\mathbf{q}} \neq 0 \quad (4.10)$$

em que se pode verificar que  $\dot{V}(\dot{\mathbf{q}}, \mathbf{e})$  diminui, uma vez que é semi-definida negativa, sendo portanto a função candidata descrita em (4.4) uma função de Lyapunov. Assim, a lei de controlo (4.9) assegura estabilidade em anel fechado.

No ponto de equilíbrio  $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = 0$  a dinâmica do manipulador (A.25) sob a lei de controlo (4.9), é descrita por:

$$(\mathbf{J}_{3D} \cdot {}^c\mathbf{W}_e \cdot {}^e\mathbf{J}_R)^T \cdot \mathbf{K}_p \cdot \mathbf{e} = 0$$

Tendo em consideração a segunda condição, definida na secção 4.2, o resultado desejado é então obtido:

$$\mathbf{e} = \mathbf{s}^* - \mathbf{s} = 0$$

Invocando o princípio da invariância de LaSalle [117], e uma vez que o sistema é autónomo  $\dot{\mathbf{s}}^* = 0$ , deve-se concluir que o sistema é assintoticamente estável em torno do ponto de



equilíbrio. Pode-se ainda concluir que o erro das características do objecto na imagem  $e$ , tende para zero no ponto de equilíbrio. A análise de estabilidade do controlador proposto na sub-secção 4.3.2 é idêntica à análise efectuada na presente sub-secção, obtendo-se os mesmos resultados.



# Capítulo 5

## Abordagens Fuzzy ao Controlo Visual

### 5.1 Introdução

Abordagens *fuzzy* é o termo utilizado nesta tese para definir as metodologias utilizadas neste capítulo para o controlo visual de robôs manipuladores. As metodologias utilizadas são baseadas em lógica *fuzzy* e aplicam esta teoria aos filtros *fuzzy*, secção 5.2, à modelação *fuzzy*, secção 5.3, e à compensação *fuzzy*, secção 5.4.1.

A lógica *fuzzy*, desde a sua introdução em [134], tem vindo a ser amplamente utilizada na área de controlo de sistemas [36, 123], sendo que esta teoria é também utilizada por exemplo em processamento de imagem e reconhecimento de padrões [14]. Esta expansão está muito ligada ao facto da lógica *fuzzy* se basear em regras do tipo *if...then*, muito próximas da linguagem natural e do funcionamento de sistemas reais, pois para existir um "efeito" é necessário existir uma "causa" que o active.

A aplicação de filtros *fuzzy* constitui a primeira contribuição, apresentada neste capítulo, ao controlo visual cinemático de robôs manipuladores. Com esta abordagem *fuzzy* procurou-se resolver os problemas relacionados com os valores indesejáveis das velocidades de junta que são enviadas ao controlador interno do robô, ver secção 6.3. Assim, esta abordagem *fuzzy* permite diminuir as elevadas amplitudes iniciais, bem como a variação de amplitude

destas ao longo do controlo. Esta melhoria será obtida, redistribuindo a amplitude da acção de controlo (velocidade de junta) durante o controlo.

A lógica *fuzzy* aplicada ao controlo de sistemas, com a utilização por exemplo do controlador PID *fuzzy* [36], levou também ao desenvolvimento da modelação *fuzzy* de sistemas através de modelos linguísticos [93, 135] ou modelos Takagi–Sugeno [130]. Os modelos linguísticos exigem um grande conhecimento do sistema, daí a sua dificuldade na aplicação a sistemas não-lineares, como é o caso de robôs manipuladores. A modelação de sistemas utilizando lógica *fuzzy* tem a grande vantagem de permitir obter um modelo transparente do sistema, em contraste com a utilização de redes neuronais [123]. Na secção 5.3 é apresentada a modelação *fuzzy* e posteriormente apresentada a metodologia para obter o modelo inverso do sistema robô-câmara.

A segunda contribuição deste capítulo é a obtenção do modelo inverso do sistema robô-câmara utilizando modelação *fuzzy*. Esta contribuição foi procurada para obter um modelo do sistema em que não fosse necessária calibração prévia e evitar as desvantagens associadas à utilização do jacobiano no controlo visual cinemático, nomeadamente na sua inversão *on-line*. Assim, com esta metodologia será obtido o modelo inverso do sistema, evitando assim os problemas referidos anteriormente.

Com a obtenção do modelo inverso *fuzzy* o passo seguinte na utilização de abordagens *fuzzy* é o de aplicar este modelo ao controlo do sistema robô-câmara. Esta solução para controlo baseia-se na utilização do modelo inverso como controlador do sistema, em que também é utilizado um compensador *fuzzy* (secção 5.4.1) para compensar eventuais perturbações externas e diferenças existentes entre o modelo e o sistema real modelado. Este sistema de controlo constitui assim a terceira contribuição deste capítulo ao controlo visual de robôs manipuladores.

## 5.2 Filtros Fuzzy

Os filtros *fuzzy* foram apresentados à comunidade científica internacional por Sousa em [120], para sistemas SISO (*Single-input Single-output*), com o intuito de resolver problemas associados à discretização das acções de controlo, quando aplicado ao controlo predictivo [16]. Uma extensão dos filtros *fuzzy* para sistemas MIMO (*Multiple-input Multiple-output*) foi introduzida em [96]. As características dos filtros *fuzzy*, serão aplicadas ao controlo visual cinemático de robôs manipuladores para modificar as acções de controlo, velocidades de junta, de forma a melhorar o desempenho do robô. O controlo visual a ser realizado pelo robô pode ser aplicado ao caso do regulador e ao seguimento de uma trajectória pré-definida. Ambos os casos serão estudados na presente secção.

### 5.2.1 Filtros Fuzzy aplicados ao Controlo Visual - Regulador

O controlo visual cinemático, quando aplicado ao caso do regulador tem a seguinte característica: quando o erro das características do objecto na imagem tender para zero as correspondentes velocidades de junta diminuam. Este facto permite aumentar as velocidades de junta, quando o robô está próximo da posição desejada, para também diminuir o tempo de estabelecimento da resposta à tarefa desejada. Outra característica do controlo visual cinemático, quando nos instantes iniciais do controlo ou quando o tempo de amostragem de visão é elevado, é a de que o erro (das características do objecto na imagem) e consequentemente as velocidades de junta assumem valores elevados. A estes valores elevados estão também associadas variações elevadas, indesejáveis ao controlo.

O comportamento indesejável das velocidades de junta deve-se à utilização de um ganho  $\lambda$  constante na lei de controlo visual cinemático, como descrito na secção 3.3 e em [38]. Em [11] é proposto um ganho adaptativo para a lei de controlo visual cinemático, baseado num decaimento exponencial, mas que apenas diminui o tempo necessário para a estabilização do sistema em torno da posição desejada. Este método tem ainda a desvantagem de

ser necessária a escolha de três parâmetros. Para manter o objecto no campo de visão da imagem, foi proposto em [89] um novo método para variação do ganho  $\lambda$  ao longo do controlo. Verificando as características deste método para o caso que se pretende resolver, conclui-se que este só assegura velocidades iniciais nulas no caso do movimento de rotação e apenas uma diminuição na velocidade de translação. Assim, será proposto em seguida uma abordagem para resolver os problemas identificados para as velocidades de junta, quando sujeitas ao controlo visual com ganho  $\lambda$  constante.

Face ao exposto anteriormente pretende-se com a aplicação de filtros *fuzzy* ao controlo visual cinemático que as velocidades de junta nunca ultrapassem valores capazes de prejudicar o desempenho global do robô, e assim evitar as situações indesejáveis ao controlo identificadas anteriormente. Este objectivo é conseguido tornando o robô mais lento nos instantes iniciais através da diminuição das velocidades de junta e mais rápido próximo da posição desejada em que as velocidades de junta são normalmente baixas, podendo por isso ser aumentadas e assim tornar mais rápida a convergência. Em seguida apresenta-se o filtro *fuzzy* e a forma de aplicação ao controlo visual de robôs manipuladores.

O filtro *fuzzy* será aplicado ao erro das características do objecto na imagem,

$$\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{s}^*$$

num determinado instante  $k$  do controlo. A sua aplicação será o resultado da multiplicação do erro  $\mathbf{e}(k)$  por um factor de ponderação *fuzzy*  $\gamma(k)$ , de forma a obter o erro filtrado  $\mathbf{e}_f(k)$ :

$$\mathbf{e}_f(k) = \gamma(k) \cdot \mathbf{e}(k) \quad (5.1)$$

O factor de ponderação *fuzzy*,  $\gamma(k) \in [0, 1]$ , é representado por uma função de pertinência triangular, apresentada na figura 5.1. Note-se que o factor de ponderação aumenta quando o robô está próximo da posição desejada e diminui quando o erro  $\mathbf{e}(k)$  tem um valor elevado. O suporte da função de pertinência é definido entre os valores  $\mathbf{e}_{\min}$  e  $\mathbf{e}_{\max}$ . Estes

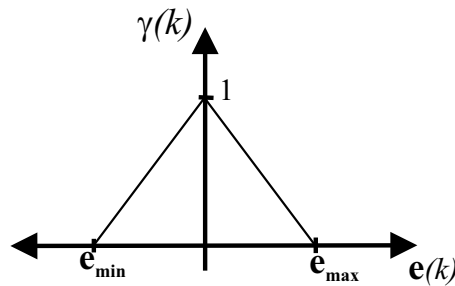


Figura 5.1: Função de pertinência triangular.

valores são iguais ao erro máximo e mínimo, definidos através do erro no instante de tempo inicial,  $\mathbf{e}(0)$ , e um pequeno  $\varepsilon > 0$  para garantir que no instante inicial,  $\gamma(0) \neq 0$ . Assim,  $\mathbf{e}_{\min}$  e  $\mathbf{e}_{\max}$  tomam os seguintes valores:

$$\begin{cases} \mathbf{e}_{\min} = -|\mathbf{e}(0)| - \varepsilon \\ \mathbf{e}_{\max} = +|\mathbf{e}(0)| + \varepsilon \end{cases} \quad (5.2)$$

Os valores  $\mathbf{e}_{\min}$  e  $\mathbf{e}_{\max}$  podem ser obtidos facilmente, pois no controlo visual cinemático é especificado um decaimento exponencial do erro das características do objecto na imagem (3.19). Ao aplicar o filtro *fuzzy* (5.1) aos vectores de erro expressos na lei de controlo visual aplicada ao caso do regulador (3.20), esta toma a forma que se descreve em seguida:

$$\dot{\mathbf{q}} = -\lambda \cdot \gamma \cdot \mathbf{J}_{2D}^{-1} \cdot (\mathbf{s} - \mathbf{s}^*) \quad (5.3)$$

### Trajectória das características da imagem

No controlo visual cinemático baseado na imagem, a trajectória das características do objecto na imagem é uma linha recta [38], desde que o sistema robô-câmara esteja perfeitamente calibrado. Mostra-se em seguida que com a aplicação de filtros *fuzzy* a este tipo de controlo, a equação da trajectória das características do objecto na imagem é quadrática, para o caso da função de pertinência escolhida ser triangular. Para diferentes

funções de pertença, é também possível obter a respectiva trajectória das características do objecto na imagem desde que se conheça a equação do factor de ponderação *fuzzy*,  $\gamma(k)$ .

Sabendo que  $\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{s}^*$  e impondo um decaimento exponencial a esse mesmo erro (3.19), a seguinte trajectória linear pode assim ser obtida:

$$\dot{\mathbf{s}}(k) = -\lambda \cdot (\mathbf{s}(k) - \mathbf{s}^*) \quad (5.4)$$

Incluindo na equação (5.4) o filtro *fuzzy* associado a  $\mathbf{e}(k)$ , o decaimento exponencial descrito anteriormente é diferente devido aos factores de ponderação associados,  $\gamma(k)$ :

$$\dot{\mathbf{s}}_f(k) = -\lambda \cdot \gamma(k) \cdot (\mathbf{s}(k) - \mathbf{s}^*) \quad (5.5)$$

Uma vez que a função de pertença escolhida é triangular, a equação de cada elemento da diagonal da matriz do factor de ponderação é a seguinte:

$$\gamma_{ii}(k) = -\frac{\lambda_{ii}}{|e_{imax}|} \cdot |e_i(k)| + 1, \quad i = 1, \dots, n_c \quad (5.6)$$

Considerando (5.5) e que  $\mathbf{e}(k) = \mathbf{s}(k) - \mathbf{s}^*$ , obtém-se finalmente a equação da trajectória quadrática para cada um das características do objecto na imagem:

$$\dot{s}_{if}(k) = \frac{\lambda_{ii}}{|e_{imax}|} \cdot s_i^2(k) - \left( \frac{2 \cdot \lambda_{ii} \cdot s_i^*}{|e_{imax}|} + \lambda_{ii} \right) \cdot s_i(k) + \left( \frac{\lambda_{ii} \cdot s_i^*}{|e_{imax}|} + \lambda_{ii} \right) \cdot s_i^* \quad (5.7)$$

Utilizando (5.5) e (5.6) obtém-se a nova equação para o decaimento do erro, que como se pode verificar já não é puramente exponencial:

$$\dot{e}_i(k) = \frac{\lambda_{ii}}{|e_{imax}|} \cdot e_i^2(k) - \lambda_{ii} \cdot e_i(k) \quad (5.8)$$



### 5.2.2 Filtros Fuzzy aplicados ao Controlo Visual - Trajectória Planeada

Quando o controlo visual cinemático é aplicado ao controlo de trajectórias planeadas na imagem, a abordagem utilizada na secção anterior pode ser aplicada. Neste caso o objectivo é o de diminuir o erro das características do objecto na imagem,  $\mathbf{e}_p$ , entre cada iteração da lei de controlo.

Ao aplicar o filtro *fuzzy* (5.1) aos vectores de erro expressos na lei de controlo visual aplicada ao planeamento de trajectórias (3.40), esta toma a forma que se descreve em seguida:

$$\dot{\mathbf{q}} = -\lambda \cdot \mathbf{J}_{2D}^{-1} \cdot (\mathbf{e}_p)_f + \frac{1}{\Delta T} \cdot \mathbf{J}_{2D}^{-1} \cdot (\mathbf{e}_{pp})_f \quad (5.9)$$

Os valores dos parâmetros da função de pertença triangular do filtro *fuzzy*, i.e.  $\mathbf{e}_{min}$  e  $\mathbf{e}_{max}$  não podem ser determinados da mesma forma que no caso do regulador, exposto na sub-secção anterior, uma vez que a definição do vector de erro é agora diferente. Assim estes devem tomar os valores que reflectam os máximos e mínimos admissíveis para os vectores de erro  $\mathbf{e}_p$  e  $\mathbf{e}_{pp}$ , e que deverão ser determinados em função da tarefa a desempenhar pelo robô.

## 5.3 Modelação *Fuzzy*

A modelação *fuzzy* de sistemas pode basear-se no conhecimento do sistema, transmitido através de peritos, expresso em formas verbais através de regras *if-then*, criando-se assim uma estrutura para o modelo do sistema. Os parâmetros da estrutura identificada terão que ser posteriormente adaptados ao modelo real, utilizando conjuntos de dados de entrada/saída do sistema [123]. Quando não existe qualquer conhecimento prévio do sistema, i.e. não existe o conhecimento transmitido através de peritos, é possível construir o modelo *fuzzy* do sistema tendo só por base as medições das variáveis de entrada

e de saída do sistema [123]. Na presente tese será abordada esta última metodologia, onde a modelação é realizada utilizando dados de entrada/saída do sistema. A análise dos dados será baseada em *fuzzy clustering* [5, 114]. Esta abordagem tem demonstrado melhores resultados que outros métodos [123], tais como redes neuronais [71] ou ANFIS (*Adaptive-Network-Based Fuzzy Inference Systems*) [74].

Assumindo que são observados os dados de entrada e saída de um determinado sistema desconhecido,  $\mathbf{y} = F(\mathbf{x})$ , o objectivo será construir uma função determinística,  $\mathbf{y} = f(\mathbf{x})$ , através dos dados observados, que aproxime  $F(\mathbf{x})$ . A função  $f$  é representada através de um conjunto de regras *fuzzy*. Dependendo da forma das proposições e da estrutura base das regras, diferentes tipos de modelos baseados em regras *fuzzy* podem ser obtidos, tal como descrito na introdução do presente capítulo.

Na presente tese, são considerados os modelos Takagi-Sugeno para construção de regras em detrimento dos modelos linguísticos, como descrito na secção 5.1. Na sub-secção seguinte estes modelos serão apresentados, assim como o procedimento de identificação baseado em *fuzzy clustering*.

### O modelo *fuzzy* Takagi–Sugeno

O tipo de modelo considerado é baseado em regras *fuzzy*, Takagi-Sugeno [130]. O modelo consiste em regras *fuzzy* que descrevem localmente a relação entre os dados de entrada e saída do modelo, de uma forma linear. O modelo apresentado em seguida descreve sistemas MISO (*Multiple Input and Single Output*), com várias entradas e uma só saída:

$$\begin{aligned}
 R_i : & \text{If } x_1 \text{ is } A_{i1} \text{ and } \dots \text{ and } x_n \text{ is } A_{in} \\
 & \text{then } y_i = \mathbf{a}_i \mathbf{x} + b_i, \quad i = 1, 2, \dots, K.
 \end{aligned}
 \tag{5.10}$$

Na equação (5.10),  $R_i$ , representa a regra  $i$ , onde  $\mathbf{x} = [x_1, \dots, x_n]^T$  identifica a variável

de entrada (antecedente),  $A_{i1}, \dots, A_{in}$  são os conjuntos *fuzzy* definidos no espaço dos antecedentes,  $y_i$  é a variável de saída (consequente) da regra  $i$ , e  $K$  indica o número de regras. A saída do modelo que agrega a saída das várias regras,  $\hat{y}$ , calculada através da média ponderada dos consequentes das várias regras, é apresentada na seguinte equação:

$$\hat{y} = \frac{\sum_{i=1}^K \beta_i y_i}{\sum_{i=1}^K \beta_i}, \quad (5.11)$$

em que  $\beta_i$  é o grau de activação da regra  $i$ , sendo obtido por:

$$\beta_i = \prod_{j=1}^n \mu_{A_{ij}}(x_j), \quad i = 1, 2, \dots, K, \quad (5.12)$$

e  $\mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0, 1]$  é a função de pertença do conjunto *fuzzy*  $A_{ij}$  no antecedente da regra  $R_i$ .

### ***Fuzzy Clustering***

Para identificar o modelo apresentado em (5.10), é necessário construir uma matriz de regressão  $\mathbf{X}$  e um vector de saída  $\mathbf{y}$  a partir dos dados disponíveis:

$$\mathbf{X}^T = [\mathbf{x}_1, \dots, \mathbf{x}_N], \quad \mathbf{y}^T = [y_1, \dots, y_N]. \quad (5.13)$$

em que  $N \gg n$ , é o número de amostras retiradas ao longo do tempo dos dados utilizados no processo de identificação. O objectivo da identificação é construir a função não-linear desconhecida  $\mathbf{y} = f(\mathbf{X})$  a partir dos dados de entrada/saída observados, em que  $f$  é o modelo Takagi-Sugeno (5.10). O número de regras  $K$ , os conjuntos *fuzzy* dos antecedentes,  $A_{ij}$ , e os parâmetros dos consequentes  $\mathbf{a}_i, b_i$  são determinados através de *fuzzy clustering* no espaço que resulta do produto das entradas e saídas [5]. Assim, o conjunto de dados  $\mathbf{Z}$  a ser classificado é composto pela matriz  $\mathbf{X}$  e pelo vector  $\mathbf{y}$ :

$$\mathbf{Z}^T = [\mathbf{X}, \mathbf{y}]. \quad (5.14)$$

Considerando a matriz  $\mathbf{Z}$  e o número estimado de grupos para classificação  $K$ , o algoritmo de *fuzzy clustering* Gustafson-Kessel [57] é aplicado para calcular a matriz de partição *fuzzy*  $\mathbf{U}$ .

Assim é obtida uma descrição do sistema em termos do comportamento das suas características locais nas regiões dos dados identificadas pelo algoritmo de classificação, em que cada grupo de classificação define uma regra.

Apesar da popularidade do algoritmo *fuzzy c-means* [13], o algoritmo de Gustafson-Kessel ao aplicar uma medida de distância adaptativa melhora a classificação. Assim, o segundo algoritmo procura regiões hiper-elipsóides nos dados de identificação descritos através dos consequentes no modelo Takagi-Sugeno.

No algoritmo de Gustafson-Kessel, os conjuntos *fuzzy* nos antecedentes das regras são obtidos através da matriz de partição  $\mathbf{U}$ , cujo elemento  $\mu_{ik} \in [0, 1]$  é o grau de pertença do conjunto de dados  $\mathbf{z}_k$  no *cluster*  $i$ . Os conjuntos vagos unidimensionais  $A_{ij}$  são obtidos através dos conjuntos vagos multidimensionais definidos na linha  $i$  da matriz de partição *fuzzy* através de projecções no espaço das variáveis de entrada  $x_j$ :

$$\mu_{A_{ij}}(x_{jk}) = \text{proj}_j(\mu_{ik}), \quad (5.15)$$

em que  $\text{proj}$  define o operador projecção [81]. Os conjuntos *fuzzy*  $A_{ij}$  são aproximados através de funções paramétricas adequadas que permitem determinar  $\mu_{A_{ij}}(x_j)$  para cada valor de  $x_j$ . Os parâmetros dos consequentes para cada regra são determinados através da aproximação com mínimos quadrados, como se descreve em seguida.

Definindo  $\theta_i^T = [\mathbf{a}_i^T, b_i]$ , a matriz  $\mathbf{X}_e = [\mathbf{X}, \mathbf{1}]$  e a matriz  $\mathbf{W}_i$  como uma matriz diagonal em  $\mathbb{R}^{N \times N}$  tendo o grau de activação,  $\beta_i(\mathbf{x}_k)$ , como valor da sua diagonal  $k$ . Assumindo que as colunas de  $\mathbf{X}_e$  são linearmente independentes e  $\beta_i(\mathbf{x}_k) > 0$  para  $1 \leq k \leq N$ , a solução do problema de mínimos quadrados,  $\mathbf{y} = \mathbf{X}_e \theta + \epsilon$ , é:

$$\theta_i = [\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e]^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y}. \quad (5.16)$$

Com a determinação dos valores óptimos dos parâmetros dos consequentes  $\mathbf{a}_i$  e  $b_i$ , o procedimento de identificação do modelo *fuzzy* fica completo. No caso de sistemas MIMO, o procedimento descrito deverá ser repetido para cada uma das saídas, obtendo-se assim o modelo de todo o sistema.

### 5.3.1 Identificação do Modelo Inverso *Fuzzy*

Existem dois métodos para obter o modelo inverso de um dado sistema, utilizando técnicas *fuzzy* [15]:

1. Identificação através de dados de entrada/saída,
2. Inversão analítica do modelo original.

O primeiro método constitui a forma mais intuitiva para obter o modelo inverso, uma vez que procura ajustar uma função inversa  $f^{-1}$  aos dados de entrada/saída, utilizando modelos baseados em lógica *fuzzy* [8] ou modelos baseados em redes neuronais [71].

O segundo método só pode ser aplicado em sistemas fuzzy para modelos com uma só entrada [6, 123]. Uma vez que um robô manipulador é um sistema MIMO, este segundo método não é aplicável. Assim, nesta tese é utilizado o método de identificação através de dados de entrada/saída, *direct inverse learning* [123].

No método *direct inverse learning* é necessário realizar um processo de aprendizagem, em que o sistema a modelar é excitado com um sinal de treino e posteriormente o sinal de entrada é reconstruído pelo modelo identificado através do sinal de saída observado. No entanto existem duas desvantagens nesta abordagem. A primeira desvantagem está relacionada com o facto de a dinâmica do sistema poder ser um mapeamento entre várias variáveis de entrada e uma variável de saída, levando a que existam vários valores possíveis de acções de controlo para o mesmo valor da variável de saída do sistema. A segunda

desvantagem deve-se à dificuldade em obter um sinal de treino adequado para o processo de aprendizagem, uma vez que o modelo inverso deverá trabalhar num domínio alargado de amplitudes de entrada e saída, assim como de largura de banda. Assim é impossível garantir que todos os modos do sistema são activados, uma vez que a excitação do sistema é assegurada através das variáveis de entrada.

Como descrito, uma solução para obter o modelo inverso é trocar as variáveis de entrada com as de saída do modelo e posteriormente aplicar o método escolhido para identificação. No entanto, o vector de saída poderá ter que ser alterado [124], pois a inversão do modelo directo poderá não ser correcta. Note-se ainda que o modelo é não-causal e como tal, é necessário realizar um *shift* no vector de saída do modelo inverso (entrada do modelo directo) para garantir a causalidade do modelo que se pretende obter, ver sub-secção 6.4.

### 5.3.2 Identificação do Modelo Inverso Fuzzy do Aparato Experimental

Na presente sub-secção é abordada a identificação do modelo inverso *fuzzy* do aparato experimental utilizado na presente tese e apresentado no anexo B. As técnicas utilizadas foram descritas nas sub-secções anteriores e baseiam-se na modelação *fuzzy*. Assim, o modelo inverso é identificado utilizando os dados de entrada/saída do sistema. Para alcançar tal objectivo, em primeiro lugar é necessário compreender o sistema em estudo para determinar quais as variáveis a observar e assim obter os dados de entrada/saída necessários à identificação. Seguidamente é realizado o processo de identificação do modelo inverso *fuzzy*.

A principal contribuição desta tese ao controlo visual de robôs com a utilização do modelo inverso *fuzzy* como controlador, é o facto de que com esta abordagem não ser necessária a calibração do sistema robô-câmara, como se irá verificar em seguida.

### O Modelo Inverso

No controlo visual cinemático, é necessário obter uma estimação precisa do jacobiano da imagem  $\mathbf{J}_i$ , da transformação entre os referenciais da câmara e do elemento terminal  ${}^c\mathbf{W}_e$ , e do jacobiano do robô no referencial do elemento terminal  ${}^e\mathbf{J}_R$ . O produto das três matrizes anteriores foi definido como Jacobiano global tanto para o controlo visual cinemático baseado em imagem (3.17), em posição (3.27) ou híbrido (como por exemplo (3.34)).

Por forma a obter um jacobiano global que se aproxime à realidade, é necessário conhecer com elevada precisão as matrizes  $\mathbf{J}_i$ ,  ${}^c\mathbf{W}_e$ ,  ${}^e\mathbf{J}_R$ , com todos os problemas de estimação associados e referidos no capítulo 3. É pois necessária uma boa calibração para obter um jacobiano analítico que permita obter uma aproximação precisa à realidade do sistema. Contudo outro problema é o do jacobiano da imagem depender das características do objecto na imagem que se estejam a utilizar: pontos, segmentos de recta, linhas, elipses, [20] e mais recentemente momentos [22].

Mesmo quando existe um modelo perfeito do jacobiano, este pode conter singularidades [53] que podem destabilizar a lei de controlo que esteja a ser utilizada no sistema. Este facto deve-se à necessidade de inversão do jacobiano (3.20), para que seja possível enviar as velocidades de junta ao anel interno do robô, ver figura 3.5. Para ultrapassar as dificuldades existentes com a utilização do Jacobiano global anteriormente referido, Suh e Kim, em [126], propuseram uma nova relação diferencial entre a variação das características do objecto na imagem e a velocidade da câmara. Esta abordagem estima a variação das características do objecto na imagem  $\delta\mathbf{s}$ , para uma dada variação da posição da câmara  $\delta{}^c\mathbf{r}_C$ , através de uma função  $G$ . Esta relação pode ainda ser dividida numa função  $G_1$  que relaciona a posição da câmara com as características do objecto na imagem, e  $F_1$  que relaciona as respectivas variações:

$$\mathbf{s} + \delta\mathbf{s} = G({}^c\mathbf{r}_C + \delta{}^c\mathbf{r}_C) = G_1({}^c\mathbf{r}_C) + F_1({}^c\mathbf{r}_C, \delta{}^c\mathbf{r}_C). \quad (5.17)$$

Considerando somente as variações apresentadas em (5.17):

$$\delta \mathbf{s} = F_1(\mathbf{r}_C, \delta \mathbf{r}_C), \quad (5.18)$$

e que a relação entre a variação da posição da câmara  $\delta \mathbf{r}_C$ , é obtida em função da variação da posição de junta,  $\delta \mathbf{q}$  e a anterior posição do robô  $\mathbf{q}$ :

$$\delta \mathbf{r}_C = F_2(\delta \mathbf{q}, \mathbf{q}). \quad (5.19)$$

As equações (5.18) e (5.19) podem ser invertidas se um mapeamento unívoco for garantido. Considerando que o cálculo da inversa é possível, os modelos inversos das equações referidas são dados pelas equações:

$$\delta \mathbf{r}_C = F_1^{-1}(\delta \mathbf{s}, \mathbf{r}_C) \quad (5.20)$$

e

$$\delta \mathbf{q} = F_2^{-1}(\delta \mathbf{r}_C, \mathbf{q}) \quad (5.21)$$

As equações (5.20) e (5.21) podem ser compostas porque a câmara, no caso em estudo da configuração *eye-in-hand*, se encontra rigidamente ligada ao elemento-terminal do robô. O mesmo é dizer que conhecendo a posição das juntas do robô  $\mathbf{q}$ , a posição da câmara  $\mathbf{r}_C$  pode ser facilmente obtida através da cinemática do robô e da transformação entre os referenciais do elemento terminal e da câmara. Assim, a função inversa  $F^{-1}$  é dada por:

$$\delta \mathbf{q} = F^{-1}(\delta \mathbf{s}, \mathbf{q}) \quad (5.22)$$

Esta relação indica que uma variação na posição de junta do robô  $\delta \mathbf{q}$ , depende da variação das características do objecto na imagem  $\delta \mathbf{s}$  e da posição anterior das juntas do robô  $\mathbf{q}$ . A equação (5.22) pode ser discretizada:

$$\delta \mathbf{q}(k) = F_k^{-1}(\delta \mathbf{s}(k+1), \mathbf{q}(k)). \quad (5.23)$$



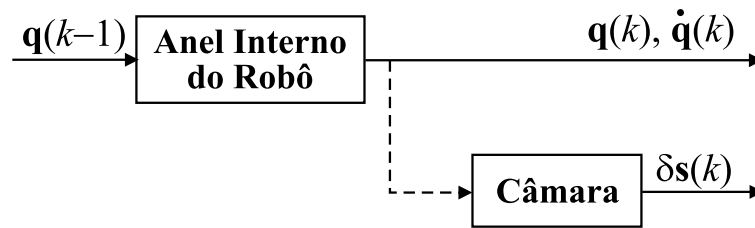


Figura 5.2: Configuração do sistema robô-câmera para identificação do modelo inverso.

A equação (5.23) descreve o que se pretende de um controlador por forma a ser aplicado ao controlo visual cinemático. Este terá que fornecer uma velocidade de junta  $\delta\mathbf{q}(k)$ , ao anel interno do robô, por forma a que este último siga uma desejada variação das características do objecto na imagem  $\delta\mathbf{s}(k+1)$ , a partir de uma determinada posição no espaço de junta do robô  $\mathbf{q}(k)$ .

Através da identificação da função inversa  $F_k^{-1}$ , obtém-se o modelo inverso do sistema robô-câmera necessário ao controlo visual, sendo para tal utilizadas as técnicas *fuzzy* apresentadas anteriormente na presente secção. Esta nova abordagem ao controlo visual de robôs, permite ultrapassar os problemas relacionados com a calibração do sistema e com a utilização do jacobiano analítico.

### O processo de identificação

O aparato experimental é basicamente constituído por um robô e por uma câmara, colocada no elemento terminal do primeiro. O sistema a modelar é portanto denominado robô-câmera e encontra-se esquematizado na figura 5.2. Uma primeira abordagem para obter os dados de entrada/saída necessários para a identificação do modelo consiste em realizar movimentos aleatórios do robô, i.e. a partir de uma posição de junta inicial o robô move-se consecutivamente entre várias posições de junta, obtidas aleatoriamente. Este procedimento deverá terminar quando todo o espaço de junta do robô for percorrido e assim será obtido o mapeamento entre as variáveis de entrada e de saída. Apesar

das várias tentativas realizados sob o procedimento anterior, o esforço foi infrutífero para identificar um modelo inverso do sistema robô-câmara que fosse fiável o suficiente para poder ser utilizado no controlo. A principal razão para este facto deveu-se a que o processo de obtenção dos dados de entrada/saída ter sido realizado em anel fechado, i.e. para movimentar o robô é necessário que este esteja sob a acção de um controlador. Refira-se ainda que o processo de identificação em anel fechado é uma tarefa difícil, especialmente quando aplicada a sistemas não-lineares [123]. Assim, foi necessário encontrar um outro procedimento para obter os dados necessários ao processo de identificação.

Em robótica é usualmente utilizada a técnica de "ensinar" ao robô a trajectória a seguir, *teach-by-showing*, para posterior utilização [125]. A partir de uma posição inicial, definida pelas posições de junta, o robô move-se para uma posição final pré-determinada, utilizando um controlador PID [4] de posição no espaço de junta para seguir uma trajectória também pré-definida entre as posições inicial e final. Durante o seguimento da trajectória são recolhidos os dados de entrada/saída necessários à identificação do modelo inverso. Embora se tenham obtido bons resultados com esta abordagem, como se poderá verificar no capítulo 6, esta tem como desvantagem a necessidade de identificar um novo modelo para uma nova trajectória a realizar pelo sistema robô-câmara. No entanto este procedimento é bastante simples e rápido, sendo possível identificar um novo modelo para uma nova trajectória. A grande vantagem desta abordagem consiste em obviar o problema de identificação em anel fechado, que não se pôde resolver com a primeira abordagem descrita.

Para a identificação do modelo inverso são usados dados de entrada/saída, sendo este obtido através das entradas do modelo directo  $\delta\mathbf{q}(k)$ , das suas saídas  $\delta\mathbf{s}(k+1)$  e do estado do sistema  $\mathbf{q}(k)$ . O vector das saídas é obtido através da equação:

$$\delta\mathbf{s}(k) = \mathbf{s}^* - \mathbf{s}(k) \quad (5.24)$$

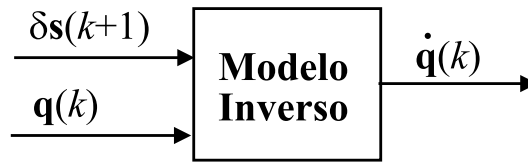


Figura 5.3: Entradas e saídas do modelo inverso a ser identificado.

Note-se agora que o objectivo final é identificar o modelo inverso, apresentado em (5.23). Assim, a modelação *fuzzy* é utilizada para tal fim, como por exemplo em [124]. Na abordagem utilizada, a entrada do sistema (a que corresponde o modelo directo),  $\dot{\mathbf{q}}(k)$ , é utilizada como saída do modelo inverso. O estado  $\mathbf{q}(k)$ , em conjunto com a saída do sistema original  $\delta\mathbf{s}(k+1)$ , são as entradas do modelo inverso, tal como apresentado na figura 5.3. Posteriormente, o modelo identificado será utilizado como controlador no controlo visual cinemático, como será apresentado na secção 5.4.

Será portanto de esperar que o modelo inverso *fuzzy* identificado seja capaz de fornecer ao robô a velocidade de junta  $\dot{\mathbf{q}}(k)$ , encarregue de mover o robô seguindo uma determinada variação das características do objecto na imagem  $\delta\mathbf{s}(k)$ , partindo de uma determinada posição de junta do robô  $\mathbf{q}(k)$ .

## 5.4 Controlo baseado no modelo inverso *Fuzzy*

A forma ideal para controlar um sistema é utilizar como controlador o modelo inverso desse mesmo sistema, numa configuração em anel aberto. Para que os resultados obtidos através deste tipo de controlador possam ser satisfatórios, é necessário que o sistema seja invertível e que o mapeamento conseguido pelo modelo inverso do sistema seja perfeito. Em aplicações reais o controlador baseado no modelo inverso não pode ser directamente aplicado, pois o mapeamento perfeito do sistema nunca é atingido. Ainda há a considerar o facto de que o sistema real não seja imune a perturbações externas, imprevisíveis e complicadas de modelar.

Na presente secção é abordado o controlo baseado no modelo inverso do sistema a controlar. Para a obtenção do modelo são utilizadas técnicas inteligentes, *fuzzy*, que recentemente têm demonstrado bons resultados em sistemas não-lineares [122, 6, 124].

A técnica de controlo baseada no modelo inverso e também a técnica baseada no controlo predictivo baseado no modelo, *Model Based Predictive Control* (MBPC) [16], conseguem abordar a dinâmica de sistemas não-lineares. Contudo, quando o sistema se encontra em regime estacionário, podem ainda surgir erros como consequência de perturbações externas ou ainda devido à discrepância existente entre o sistema real e o modelo identificado para este. Como tal é necessária a inclusão de um mecanismo que compense este tipo de erros, i.e. erros em regime estacionário. A abordagem clássica seria utilizar na lei de controlo uma acção integral [4]. Contudo, este tipo de acção no controlador não é desejável em sistemas altamente não-lineares, pois é necessário afinar vários parâmetros para diferentes zonas de operação do sistema modelado. Outra solução possível seria a utilização do controlo por modelo interno *Internal Model Control* (IMC) [46], que também pode ser aplicado a sistemas não-lineares [37]. O controlo por modelo interno baseia-se na alteração do sinal de referência, através da inclusão do erro entre a saída do modelo e a saída do sistema. Com o controlo não-linear por modelo interno é introduzido no anel de realimentação um filtro linear para o erro, que tem a desvantagem de deteriorar a dinâmica do anel fechado quando o sistema se encontra na presença de não-linearidades.

Tendo em consideração o descrito anteriormente, na presente tese é apresentada a solução baseada num compensador *fuzzy*, também denominada compensação *fuzzy*, que constitui uma extensão ao proposto em [121] e que pretende ser uma boa solução para eliminar erros em regime estacionário. Note-se ainda que para aplicar o controlo por modelo interno, seria necessário conhecer os modelos directo e inverso do sistema em causa. Para tal e para o sistema utilizado na presente tese, os modelos teriam que ser identificados directamente dos dados de entrada/saída. Assim, seria impossível assegurar que o modelo inverso identificado convergiria para o modelo inverso real, i.e. seria impossível assegurar estabilidade e um erro nulo em regime estacionário para o controlo por modelo interno

[46] do sistema não-linear em estudo.

O controlo baseado no modelo inverso *fuzzy* apresentado na presente tese, não é uma alternativa ao controlo por modelo interno, impossível de aplicar neste caso, mas sim uma alternativa aos restantes controladores visuais já apresentados. Os resultados apresentados na secção 6.5 revelarão a eficácia da abordagem proposta.

### 5.4.1 Compensador *Fuzzy*

O objectivo do compensador *fuzzy* é o de compensar o erro em regime estacionário, tendo como base a informação contida no modelo do sistema. Para tal, é então definido um conjunto *fuzzy*, para o erro estacionário, que determina o grau de activação do compensador *fuzzy*. Com a introdução do referido conjunto *fuzzy* é conseguida uma passagem suave da compensação entre os estados de activo e inactivo. Utilizando a derivada local do modelo relativamente à acção de controlo, é possível obter a compensação através do ajuste de um só parâmetro (de uma forma similar ao ganho integral do controlador PID clássico).

A compensação *fuzzy* utiliza explicitamente o modelo não-linear do sistema, que pode ser o modelo clássico não-linear em espaço de estados ou um modelo tipo caixa-preta. No controlo através do modelo inverso, a derivada referida anteriormente é calculada directamente através do modelo inverso identificado, simplificando assim o processo de compensação *fuzzy*. O método é apresentado para sistemas SISO não-lineares, por uma questão de simplicidade, podendo ser facilmente extendido para sistemas MIMO (por exemplo, o caso de um robô manipulador de 2 graus de liberdade).

Para simplificar a notação, nesta secção é conveniente atrasar um passo no tempo o modelo do sistema. O modelo discreto SISO de regressão do sistema a controlar é dado por:

$$y(k) = f(x(k-1)), \quad (5.25)$$

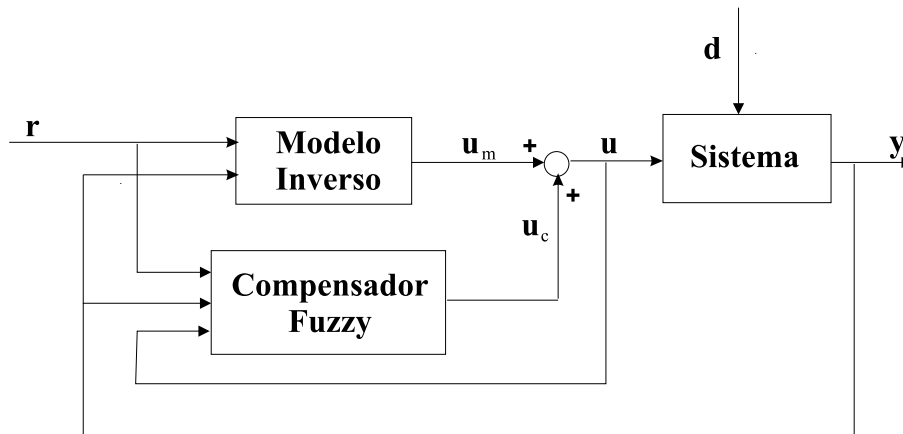


Figura 5.4: O compensador *fuzzy* inserido no controlo por modelo inverso.

em que  $x(k-1)$  é o estado que contém as saídas (com atraso) do modelo inverso, i.e. as entradas e os estados do sistema. O compensador *fuzzy* utiliza uma acção correctiva denominada de  $u_c(k)$ , a qual será adicionada à acção obtida através do controlador baseado no modelo inverso,  $u_m(k)$ , como se pode verificar na figura 5.4. A acção de controlo total a aplicar ao sistema é definida por,

$$u(k) = u_m(k) + u_c(k). \quad (5.26)$$

O controlador apresentado na figura. 5.4 é baseado no modelo inverso do sistema, mas qualquer outro controlador poderia ser aqui utilizado, como por exemplo um controlador preditivo. Por forma a ter em conta o ruído e um pequeno sinal de *offset*, o conjunto *fuzzy*  $SS$  define a região onde a compensação deverá estar activa, como apresentado na figura 5.5. O erro é definido através da expressão  $e(k) = r(k) - y(k)$ , e a função de pertença  $\mu_{SS}(e(k))$  é construída por forma a permitir a compensação do erro estacionário, sempre que o suporte de  $\mu_{SS}(e(k))$  não é nulo. O valor  $B$  que determina a dimensão de  $\text{core}(\mu_{SS})$  deve ser o limite superior do valor absoluto do erro previsível, em regime estacionário. Assim, a compensação *fuzzy* é completamente activada no intervalo  $[-B, B]$ . O suporte de  $\mu_{SS}(e(k))$  deverá ser escolhido de forma a que se realize uma transição suave entre os estados de activação e desactivação da compensação. A suavidade de  $SS$  induz suavidade

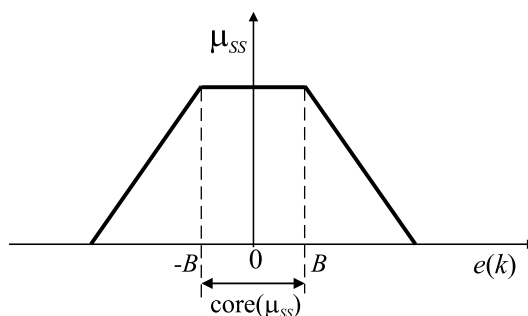


Figura 5.5: Definição da fronteira de  $SS$ , onde a compensação está activa.

na acção do compensador *fuzzy*  $u_c(k)$ , e evita assim mudanças abruptas na transição da acção de controlo  $u(k)$ . A acção do compensador *fuzzy*  $u_c(k)$  no instante  $k$  é dada por

$$u_c(k) = \mu_{SS}(e(k)) \left( \sum_{i=k_0}^{k-1} u_c(i) + K_c e(k) f_u^{-1} \right), \quad (5.27)$$

em que  $\mu_{SS}(e(k))$  é o valor da função de pertinência associada ao erro no instante  $k$ ,  $k_0$  é o instante de activação da compensação *fuzzy*,  $K_c$  é um parâmetro constante a definir e

$$f_u = \left[ \frac{\partial f}{\partial u(k-1)} \right]_{x(k-1)} \quad (5.28)$$

é a derivada parcial da função  $f$  definida em (5.25) relativamente à acção de controlo  $u(k-1)$ , para o estado actual do sistema  $x(k-1)$ .

Comparando (5.27) com a acção integral clássica, surgem dois novos termos:  $\mu_{SS}(e(k))$ , cujo efeito foi anteriormente descrito, e o termo descrito em (5.28), que representa a sensibilidade do modelo quando sujeito a uma variação do sinal de entrada. Em sistemas lineares, este termo é constante sendo incluído na constante  $K_c$ , mas para sistemas altamente não-lineares, a compensação é melhorada ao incluir este termo. Quando a derivada parcial aumenta o sistema torna-se mais sensível a mudanças nas acções de controlo, sendo necessária uma pequena compensação. O caso contrário é também válido. Assim, o inverso de (5.28) deverá ser considerado na acção do compensador (5.27). O parâmetro  $K_c$  será ajustado para as condições de funcionamento do sistema. O seu valor deverá

ser seleccionado de forma a que o erro em regime estacionário diminua o mais depressa possível sem oscilações na resposta do sistema. Estas oscilações podem ocorrer na acção do compensador *fuzzy* se esta for de valor elevado, levando a um novo erro  $e(k+1)$  de sinal oposto ao anterior erro  $e(k)$ .

Quando o modelo do sistema  $f$  está disponível, a derivada parcial definida em (5.28) pode ser calculada sempre que o modelo seja diferenciável. O mesmo ocorre sempre que exista o modelo inverso do sistema, o que acontece na presente tese. Contudo, alguns tipos de modelação caixa-preta resultam em vários modelos lineares que definem o modelo global do sistema. Para este tipo de modelos, a derivada definida em (5.28) não está definida no regime transitório das várias partes lineares do modelo. A solução deste problema baseia-se na definição de uma pseudo-derivada para estes pontos singulares (transições entre troços lineares) que é dada pelo valor médio das derivadas à esquerda e à direita. Estas duas derivadas existem pois provêm dos troços lineares do modelo que aproxima o sistema não-linear. Assim, a derivada para estes pontos pode ser calculada através de,

$$\left[ \frac{\partial f}{\partial u} \right]_{x(k-1)} = \frac{\left[ \frac{\partial f}{\partial u(k-1)} \right]_{(x(k-1))^+} + \left[ \frac{\partial f}{\partial u(k-1)} \right]_{(x(k-1))^-}}{2}. \quad (5.29)$$

Esta aproximação não deteriora o desempenho do controlo, como descrito em [121].



# Capítulo 6

## Resultados Experimentais em Controlo Visual

### 6.1 Introdução

No presente capítulo são apresentados os resultados experimentais que dizem respeito às técnicas de controlo visual de robôs manipuladores apresentadas nos capítulos anteriores, onde se incluem as abordagens clássicas e as propostas nesta tese, por forma a ser possível realizar algumas comparações entre elas. Existem porém casos em que serão apresentadas simulações por forma a confirmar resultados teóricos no controlo visual de seis graus de liberdade. No entanto estes resultados de simulação serão posteriormente objecto de verificação no aparato experimental disponível, como será o caso da aplicação com filtros *fuzzy*.

Na primeira parte deste capítulo são descritas as condições experimentais presentes durante as diversas experiências realizadas no aparato experimental desenvolvido para o efeito. Para harmonizar a apresentação de resultados em todo este capítulo, são definidas as variáveis que melhor descrevem o controlo visual de robôs manipuladores. Sempre que possível são utilizadas, em todas as experiências, as mesmas posições do objecto face ao robô manipulador para assim serem definidas as mesmas tarefas para tipos diferentes de controladores.

Depois da descrição dos aspectos gerais de todas as experiências realizadas, são apresentados em diferentes secções do presente capítulo os resultados experimentais bem como as respectivas análises. De salientar que sempre que necessário, em cada secção, são apresentadas as alterações realizadas às condições experimentais gerais. Após a apresentação dos resultados, estes serão analisados em função do objectivo que se pretende para cada lei de controlo testada, bem como, e sempre que possível, a sua comparação com as restantes leis de controlo descritas na tese.

## 6.2 Condições Experimentais

As leis de controlo visual apresentadas nos capítulos anteriores foram testadas num robô manipulador de dois graus de liberdade, em que no elemento terminal foi colocada uma câmara a olhar um objecto. O aparato experimental desenvolvido para a realização das experiências é apresentado no anexo B. O objecto utilizado nos testes é planar e contém quatro *led's* que podem ser ligados independentemente, permitindo assim obter um número variável de informações visuais, i.e. pontos. Deste conjunto de *led's* apenas um será utilizado na maioria das experiências, o que está no meio dos restantes pontos, pois o robô do aparato experimental tem dois graus de liberdade e as duas coordenadas do ponto na imagem são suficientes para o controlo. Existem situações em que são necessários mais pontos para poder reconstruir a pose do objecto relativamente à câmara, e depois aplicar o controlo visual em posição ou híbrido.

A informação visual utilizada é o centro de gravidade de cada *led*, obtido através da imagem capturada. Com imagens em que se encontram bem definidos os *led's*, que caracterizam o objecto planar, é possível fechar o anel de controlo com sistemas de aquisição comerciais, ver o Anexo B. De referir que a abordagem realizada na presente tese não teve em conta os importantes problemas associados ao tratamento de imagem (inicialização, correspondências e seguimento de características) associadas a objectos realmente complexos, que não é o caso do objecto utilizado. A calibração da matriz de passagem do



Figura 6.1: Imagem correspondente à posição 1.

referencial da câmara para o referencial do elemento terminal do robô foi realizada manualmente.

$${}^e\mathbf{T}_c = \begin{bmatrix} 0 & -1 & 0 & 0.05 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.03 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Foram utilizados os parâmetros de calibração da câmara fornecidos pelo fabricante da mesma, não tendo sido realizadas outras experiências para a estimação dos seus parâmetros de calibração.

$$\mathbf{K} = \begin{bmatrix} 716.1905 & 0 & 0 \\ 0 & 742.9787 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

a que corresponde a distância focal  $f = 0.006$ .

Diversas posições da câmara relativamente a um objecto, sempre na mesma posição, permitem também obter diversas imagens deste. Como a câmara se encontra ligada rigidamente ao elemento terminal do robô, para obter as imagens inicial e desejada para a execução do controlo visual é necessário mover o robô. Assim e em seguida são apresentadas as diversas posições utilizadas na presente tese para a obtenção de resultados:



Figura 6.2: Imagem correspondente à posição 2.

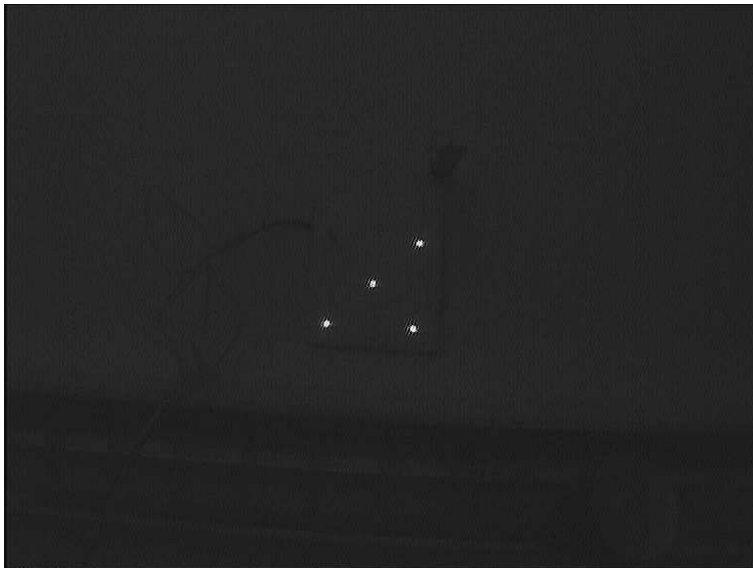


Figura 6.3: Imagem correspondente à posição 3.

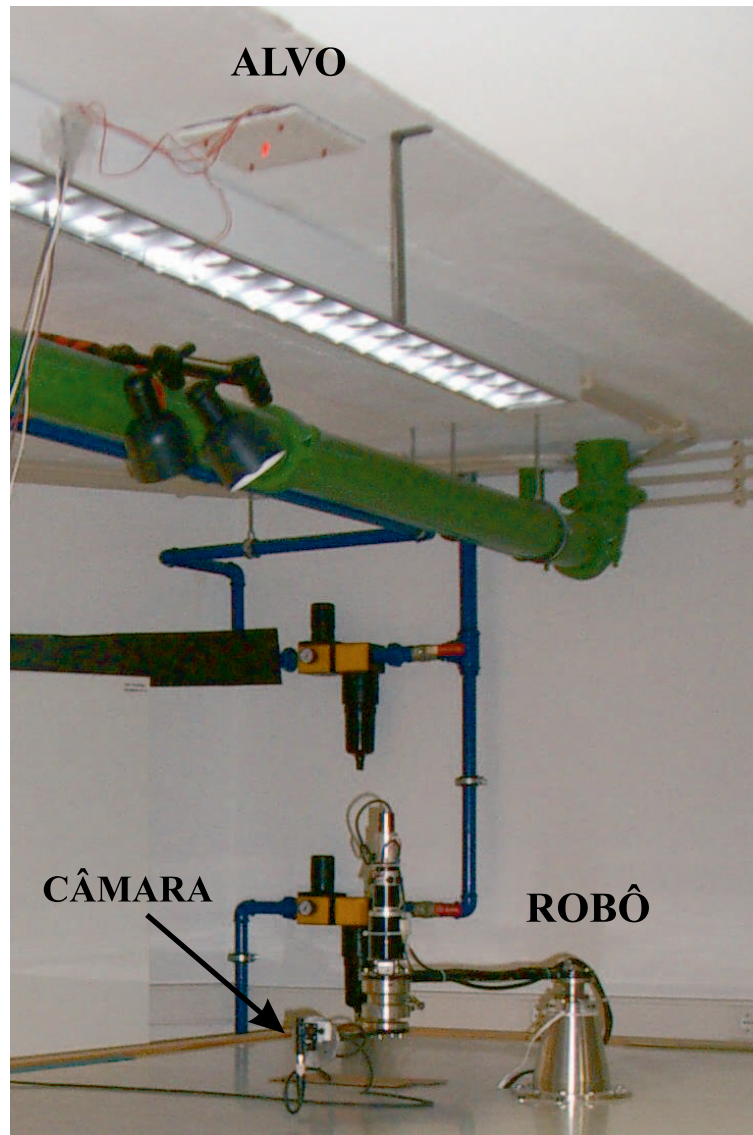


Figura 6.4: Fotografia de conjunto do robô manipulador, com a câmera acoplada, e o objecto (alvo) planar.

- Posição 1: , figura 6.1;
- Posição 2: , figura 6.2;
- Posição 3: , figura 6.3;

As imagens anteriores foram obtidas com a configuração do aparato experimental como descrito na figura 6.4, em que se apresenta uma fotografia de conjunto do robô manipulador, com a câmara acoplada, e o objecto planar. O plano de movimento do robô manipulador e o plano do objecto são paralelos e distam entre si 110 centímetros, sendo esta medida constante em todas as experiências realizadas.

Um facto importante e que condiciona os resultados obtidos, deve-se aos erros inerentes ao cálculo do centróide do(s) pontos na imagem, de aproximadamente 4[*pixel*]. Outro facto deve-se ainda à existência de zonas mortas nos motores da junta do robô. Estes factores levam à existência de erros no posicionamento inicial do robô e ainda aumentam a dificuldade em atingir a posição desejada na imagem.

Por forma a apresentar, na presente tese, os resultados obtidos no aparato experimental de uma forma uniforme são definidas de seguida as variáveis que revelam o comportamento das leis de controlo testadas. Note-se aqui que podem surgir variáveis associadas a uma dada lei de controlo, com interesse em monitorizar para estudar comportamentos particulares, mas directamente fora do âmbito de uma análise global entre leis de controlo. Assim, são definidos os seguintes parâmetros gerais a monitorizar durante execução da lei de controlo:

- a) erro das características do objecto na imagem, durante a execução da lei de controlo;
- b) trajectória das características do objecto na imagem, durante a execução da lei de controlo;
- c) velocidade das juntas do robô, durante a execução da lei de controlo;

## 6.3 Filtros Fuzzy

Na presente secção são apresentados os resultados experimentais obtidos quando da aplicação de filtros *fuzzy* ao controlo visual cinemático baseado na imagem, tanto para o caso do regulador como o caso de planeamento de trajectórias. No caso do regulador apenas é necessário definir a priori as imagens inicial e desejada, enquanto para o planeamento de trajectórias é necessário (também a partir das imagens inicial e desejada) calcular a trajectória *off-line*, como definido na secção 3.6.

Para comparar os resultados obtidos com ou sem a aplicação de filtros *fuzzy* ao controlo visual cinemático baseado na imagem é necessário proceder a duas definições;

- *amplitude global*, a amplitude máxima obtida durante o controlo visual;
- *amplitude local*, a amplitude máxima obtida entre duas imagens consecutivas, durante o controlo visual.

### 6.3.1 Aplicação ao caso do Regulador

Para comprovar os resultados teóricos definidos na secção 5.2 optou-se numa primeira fase por obter resultados para o caso do controlo visual de seis graus de liberdade, através de uma *Toolbox de Visual Servoing* disponibilizada em [18]. Numa segunda fase são apresentados e discutidos os resultados obtidos no aparato experimental de dois graus de liberdade.

#### Resultados de simulação para seis graus de liberdade

O modelo de simulação [18], adaptado pelo autor para a aplicação de filtros *fuzzy*, não é mais que a implementação directa das equações do controlo visual cinemático baseado na imagem em que a câmara está acoplada ao elemento terminal, como definido em (3.9). Aqui o robô manipulador é considerado como um posicionador perfeito e a lei de controlo

visual tem como saídas as velocidades de translação e rotação da câmara, i.e. o tursor de velocidade  $\mathbf{v}$ :

$$\mathbf{v} = -\lambda \cdot \hat{\mathbf{J}}_i^+(x, y, Z) \cdot (\mathbf{s} - \mathbf{s}^*), \quad (6.1)$$

em que  $\hat{\mathbf{J}}_i^+(x, y, Z)$  define a pseudo-inversa do jacobiano da imagem.

No caso de simulação apresentado o objecto é planar e define-se com oito características na imagem, que correspondem a quatro cantos de um quadrado. Como são utilizadas oito características, o jacobiano da imagem tem dimensão  $(8 \times 6)$ , i.e. o jacobiano da imagem definido em (3.11) deve ser repetido quatro vezes de forma a se obterem as oito linhas necessárias.

Quando da utilização de filtros *fuzzy* os ganhos proporcionais da lei de controlo (6.1) foram  $\lambda = 0.25 \cdot \mathbf{I}_c$  e sem a aplicação dos filtros foram  $\lambda = 0.125 \cdot \mathbf{I}_c$ , em que aqui  $c$  define o número de características do objecto na imagem. Uma vez que estes ganhos reflectem a velocidade de decaimento do erro, foram escolhidos por forma a obter aproximadamente 35 segundos para o tempo de estabelecimento.

Na figura 6.5 é apresentado o tursor de velocidade da câmara, resultado da acção da lei de controlo visual (6.1), aquando do movimento da câmara para a posição desejada das características da imagem  $\mathbf{s}^*$ . Verifica-se nos instantes iniciais que, quando os filtros *fuzzy* são aplicados as velocidades da câmara são menores do que a situação em que os mesmos não são aplicados. Verifica-se também para as seis componentes da velocidade da câmara que a sua amplitude global diminui com a aplicação dos filtros *fuzzy*. Analisando o movimento do referencial da câmara, podemos ainda concluir sobre o trabalho requerido ao robô manipulador para mover a câmara para a posição desejada. Uma vez que o trabalho realizado por uma força para mover uma partícula é igual à variação da energia cinética durante o caminho percorrido [9], podemos calcular o trabalho para a situação em análise. A energia cinética de uma partícula depende da sua massa e velocidade, como o robô utilizado é o mesmo nos dois casos conclui-se que apenas existe dependência na velocidade. A energia cinética tem como contribuições as devidas às amplitudes das



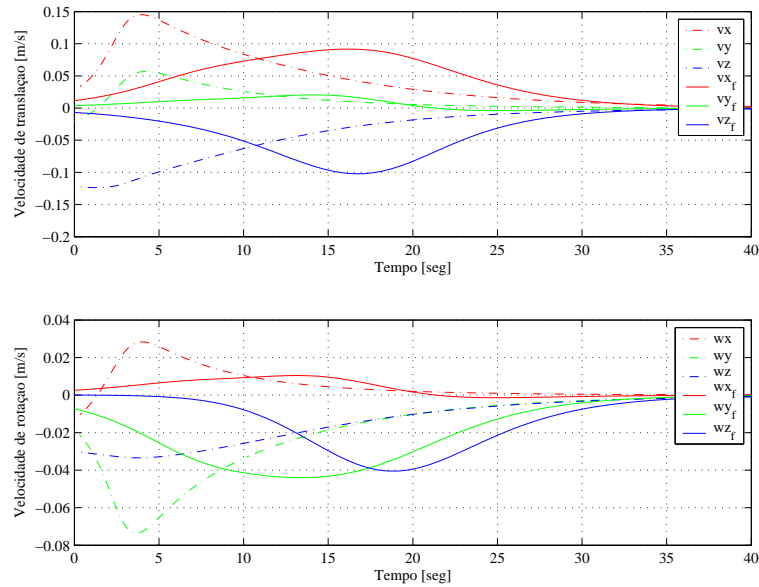


Figura 6.5: Comparação do torsor de velocidade da câmara durante o controlo visual, com e sem a utilização de filtros *fuzzy*. Cima - velocidades de translação. Baixo - velocidades de rotação. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.

velocidades de translação e rotação. Conclui-se portanto que para o caso da simulação apresentada, no caso de serem utilizados os filtros *fuzzy* durante o controlo visual, estes só iriam necessitar de 93% da energia aplicada sem a utilização dos filtros, o que se reflecte num menor consumo de energia do robô para realizar a mesma tarefa.

O comportamento do erro das características do objecto na imagem é apresentado na figura 6.6. O comportamento evidenciado pela aplicação dos filtros *fuzzy* era esperado devido à sua formulação, i.e. as características iniciam o seu movimento lentamente mas atingem a posição desejada ao mesmo tempo, visto que a velocidade aumenta suavemente durante o controlo (ver figura 6.5).

As trajectórias das características do objecto na imagem são apresentadas na figura 6.7. Pode-se constatar através dos resultados da simulação apresentada que os resultados teóricos apresentados na secção 5.2.1 são verificados, i.e. quando são aplicados os filtros *fuzzy* as trajectórias das características na imagem deixam de ser uma linha recta. Saliente-se o facto que ao ser utilizada, como é o caso, uma função de pertença triangular

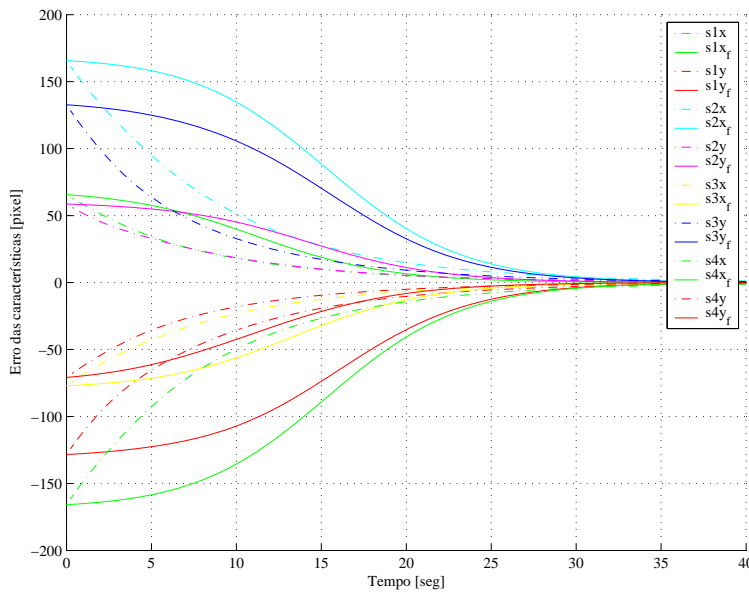


Figura 6.6: Comparação do erro das características do objecto na imagem, durante o controlo visual. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.

a trajectória é quadrática.

### Resultados experimentais no aparato experimental

O aparato experimental utilizado para obter os resultados apresentados de seguida é descrito no anexo B. A lei de controlo visual implementada tem como saídas as velocidades de junta do robô, sendo fixados os ganhos proporcionais para  $\lambda_1 = 0.04$  e  $\lambda_2 = 0.08$  associados às duas características do objecto na imagem. Estes ganhos asseguram a maior rapidez na resposta do robô para realizar a tarefa pré-definida, sem a utilização de filtros *fuzzy*. A experiência realizada irá mover o robô da posição 1 para a posição 3, ver sub-secção 6.2, a que corresponde o movimento entre a posição de junta inicial  $\mathbf{q}_i = [0; 0]$  e a posição de junta final  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$ .

A figura 6.8 apresenta as saídas da lei de controlo visual, quando o robô se move para atingir a posição desejada  $\mathbf{s}^* = (0 \ 0)^T$  no plano da imagem, i.e. as velocidades de junta. Quando são aplicados os filtros *fuzzy* à lei de controlo visual, as velocidades de junta são menores como se pode verificar em ambos os gráficos apresentados na figura 6.8.

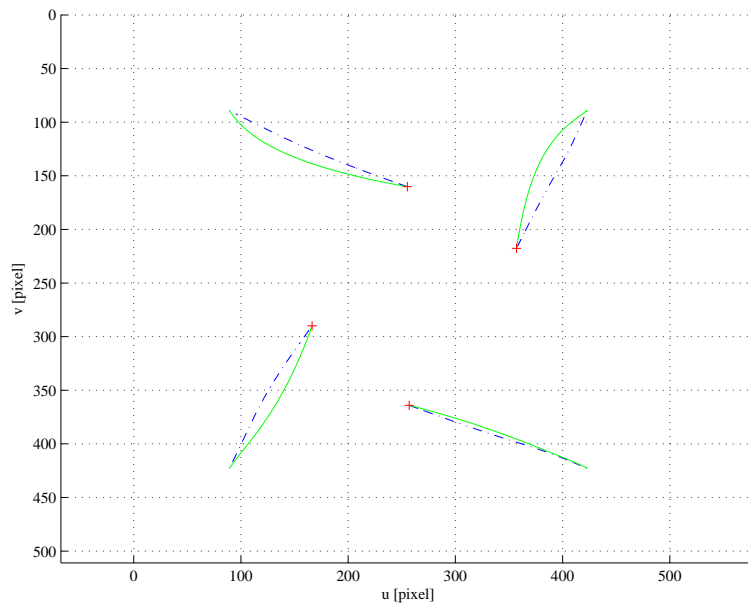


Figura 6.7: Comparação das trajetórias das características do objecto na imagem, durante o controlo visual. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*. (+) posição desejada.

Por forma a manter o tempo de estabelecimento obtido sem a utilização de filtros *fuzzy*, a constante  $\varepsilon$  definida em (5.2) deverá ser grande  $\varepsilon = 450[\text{pixel}]$ , como se pode verificar no gráfico inferior da figura 6.8. Se o objectivo for somente diminuir as velocidades nos instantes iniciais,  $\varepsilon$  deverá ser pequeno, como se pode verificar para o caso em que  $\varepsilon = 40[\text{pixel}]$  através do gráfico superior da figura 6.8. Ambos os resultados anteriores confirmam os obtidos anteriormente em simulação, para o caso geral de seis graus de liberdade.

Conclui-se que, para o caso da utilização de filtros *fuzzy* apresentada no gráfico inferior da figura 6.8 e aplicando a análise da energia necessária para a realização da mesma tarefa, que o robô só iria consumir 98% da energia aplicada sem a utilização dos filtros.

Também se pode verificar, na figura 6.8, que a amplitude global é diminuída quando se aplicam filtros *fuzzy* ao controlo visual com um tempo de amostragem de visão de 20[ms]. Este efeito é bem mais visível quando se aumenta o tempo de amostragem vídeo, no caso da figura 6.9 para 100[ms]. Devido ao aumento do tempo de amostragem, no controlo

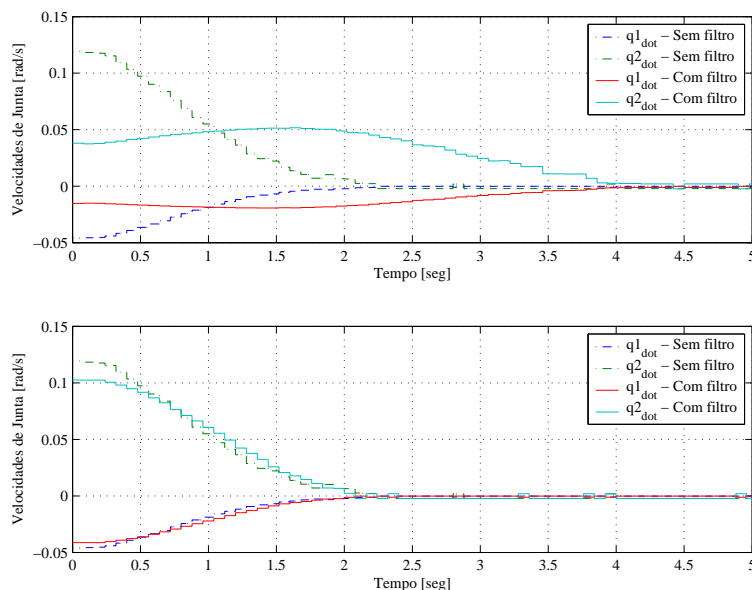


Figura 6.8: Comparação das velocidades de junta durante o controlo visual, com e sem a utilização de filtros *fuzzy*. Tempo de amostragem de visão 20[ms]. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*. Cima -  $\varepsilon$  pequeno. Baixo -  $\varepsilon$  grande.

visual sem a aplicação de filtros *fuzzy* surgem oscilações indesejáveis nas velocidades de junta que são bastante diminuídas quando estes são aplicados à lei de controlo visual, como se verifica na figura 6.9.

O comportamento do erro das características do objecto na imagem durante o controlo visual é apresentado na figura 6.10. O comportamento evidenciado é similar ao verificado nos resultados de simulação apresentados anteriormente e está de acordo com os gráficos apresentados na figura 6.8. A figura 6.11 apresenta a trajectória das características do objecto na imagem, no plano da imagem. A conclusão da análise teórica apresentada na secção 5.2.1 é verificada, pois neste caso a trajectória não é uma linha recta. No entanto quando não se aplicaram os filtros *fuzzy* a trajectória em linha recta não foi atingida, como observado nos resultados em simulação. Esta situação é devida ao facto da calibração do sistema robô-câmara não ser perfeita.

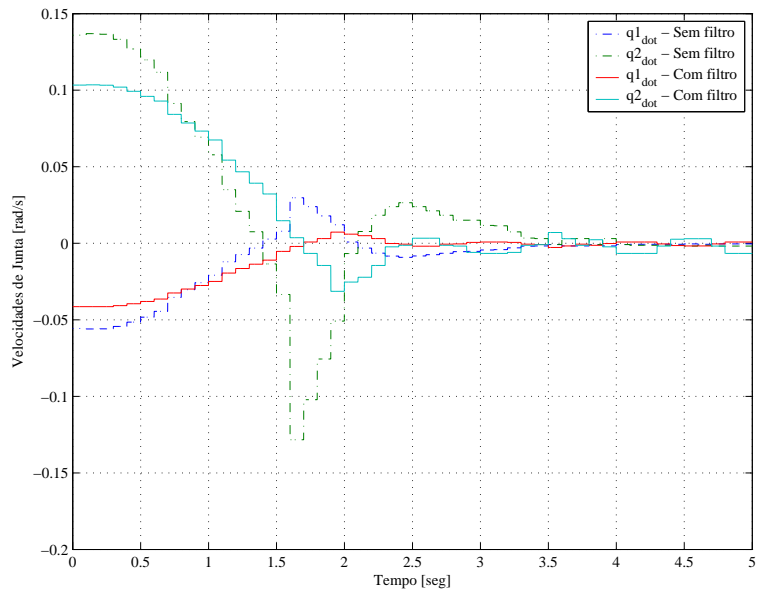


Figura 6.9: Comparação das velocidades de junta durante o controlo visual, com e sem a utilização de filtros *fuzzy*. Tempo de amostragem de visão 100[ms]. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.

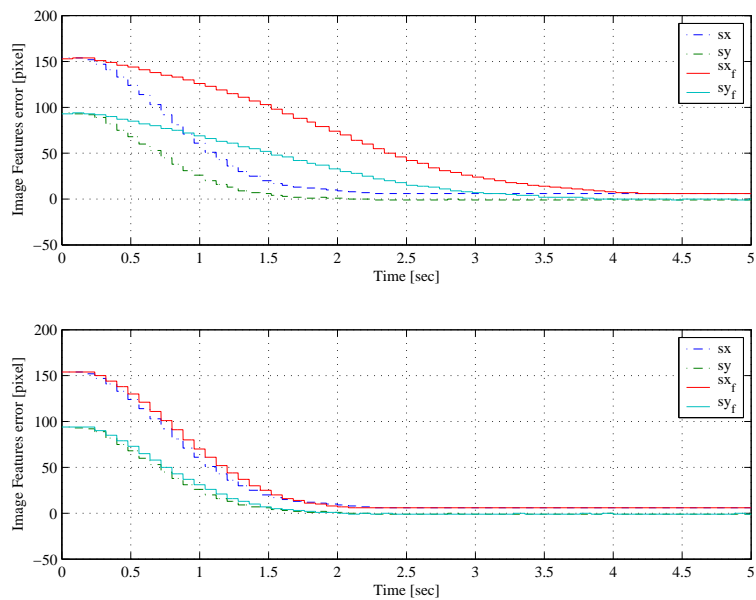


Figura 6.10: Comparação do erro das características do objecto na imagem, com e sem a utilização de filtros *fuzzy*. Tempo de amostragem de visão 20[ms]. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*. Cima -  $\epsilon$  pequeno. Baixo -  $\epsilon$  grande.

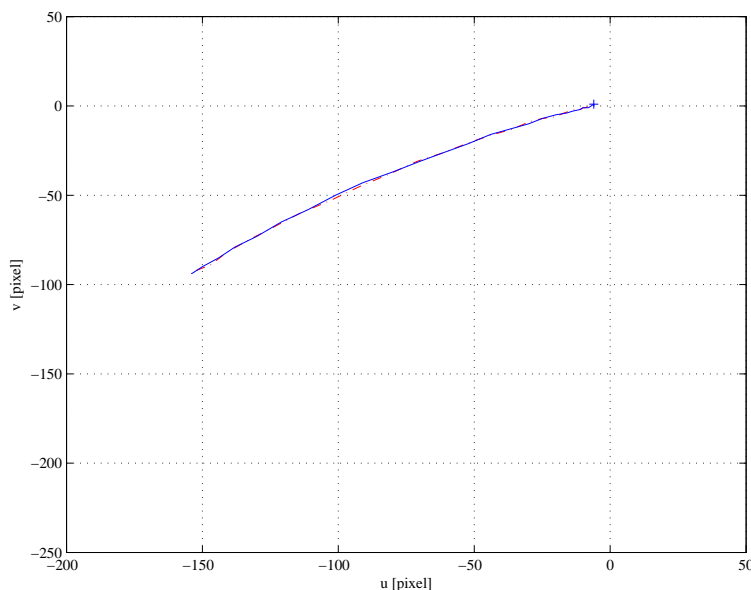


Figura 6.11: Comparação das trajetórias das características do objecto na imagem. Tempo de amostragem vídeo 20[ms]. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*. (+) posição desejada.

### 6.3.2 Aplicação ao caso do Planeamento de Trajectórias

Na presente sub-secção são apresentados e comparados, os resultados obtidos no controlo visual cinemático com planeamento de trajectórias quando se aplicam os filtros *fuzzy* ao erro das características do objecto na imagem.

No caso das experiências apresentadas nesta sub-secção foi planeada a trajectória apresentada na figura 6.15, entre a posição 2 e a posição 3, ver sub-secção 6.2, que corresponde ao movimento do robô entre a posição de junta inicial  $\mathbf{q}_i = [-\frac{\pi}{2}; \frac{\pi}{2}]$  e a posição de junta final  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$ . Nas experiências foi utilizado o ganho  $\lambda = 2$ , referido em (5.9). Na figura 6.12 é apresentada a saída da lei de controlo visual cinemático, as velocidades de junta, durante o seguimento realizado pelo robô à trajectória  $\mathbf{s}^p$ , definida no plano da imagem. Verifica-se que com a aplicação dos filtros *fuzzy* a amplitude local das velocidades de junta diminui. Esta diminuição contribuiu para que ao fim de 10 segundos, i.e. quando se atinge a posição desejada das características do objecto na imagem figura 6.14, o robô consumisse 97% da energia aplicada sem a utilização dos filtros. Verifica-se também que

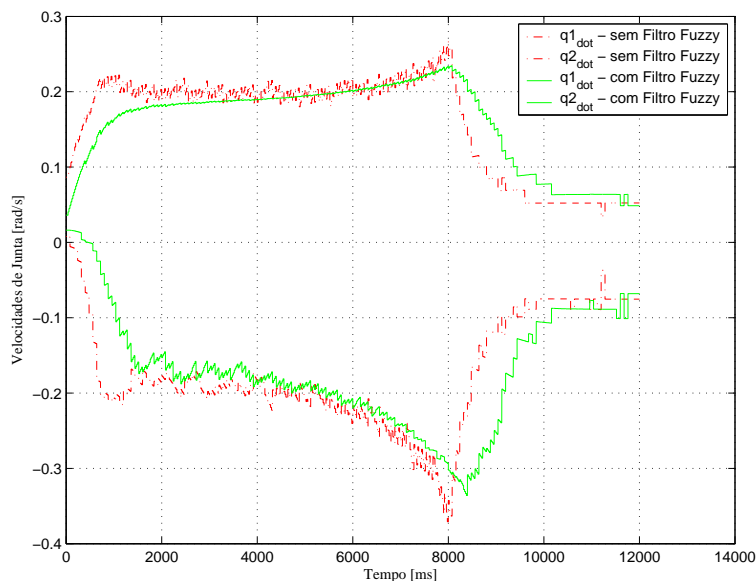


Figura 6.12: Comparação das velocidades de junta durante o controlo visual com trajectória planeada, com e sem a utilização de filtros *fuzzy*. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.

nos instantes iniciais as velocidades de junta diminuem com a utilização dos filtros *fuzzy*, tal como sucedia no caso do regulador. A diminuição das velocidades nos instantes iniciais tem a desvantagem de aumentar o erro entre as características do objecto na imagem actuais e as planeadas, como se pode verificar na figura 6.13, levando também ao aumento do tempo de estabelecimento em um segundo. No entanto, o erro entre as características do objecto na imagem desejadas e as actuais diminui exponencialmente, como se pode verificar na figura 6.14.

Na figura 6.15 são apresentadas as trajectórias no plano da imagem. Tendo como base a trajectória planeada verifica-se que o controlador visual cinemático, com planeamento de trajectória, quando se aplicam, ou não, os filtros *fuzzy* leva o robô para a vizinhança da posição desejada na imagem  $\mathbf{s}^* = (0 \ 0)^T$ . Saliente-se que a posição inicial das trajectórias não coincide, devido ao erro existente na detecção do centróide e ainda ao erro de posicionamento do robô.

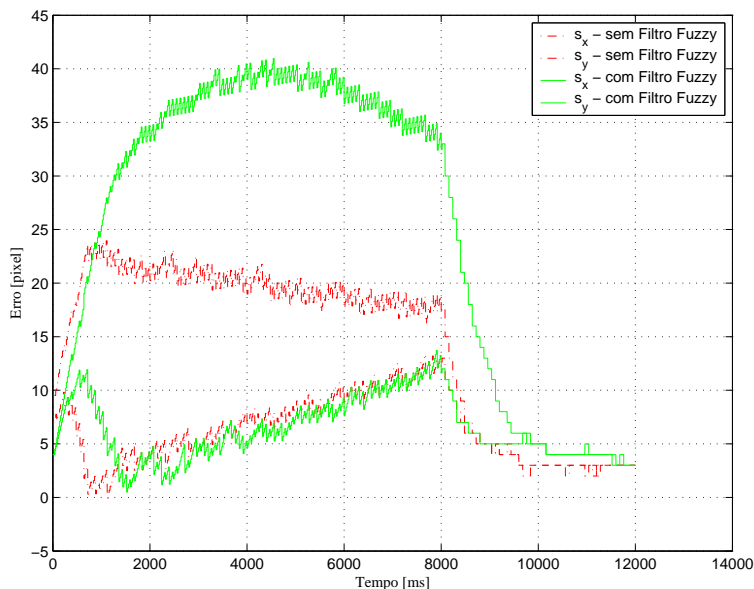


Figura 6.13: Comparação do erro das características do objecto na imagem, entre a posição actual e a trajectória planeada, com e sem a utilização de filtros *fuzzy*. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.

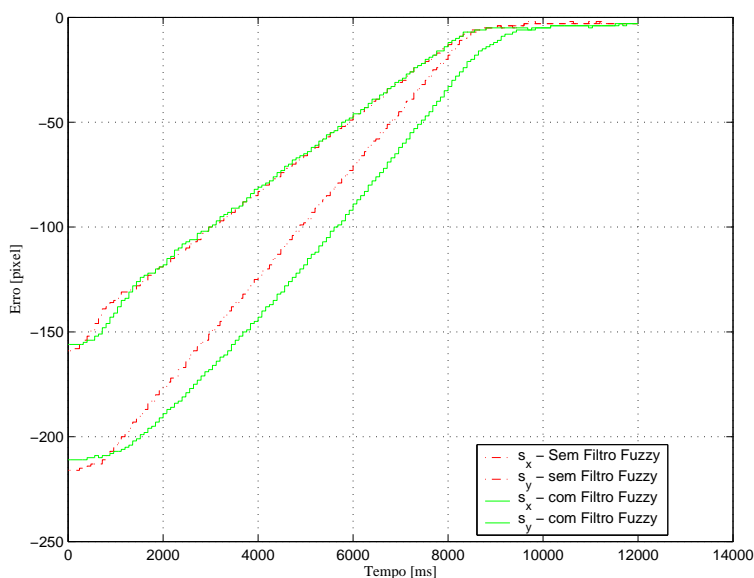


Figura 6.14: Comparação do erro das características do objecto na imagem, entre a posição actual e a desejada, com e sem a utilização de filtros *fuzzy*. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*.



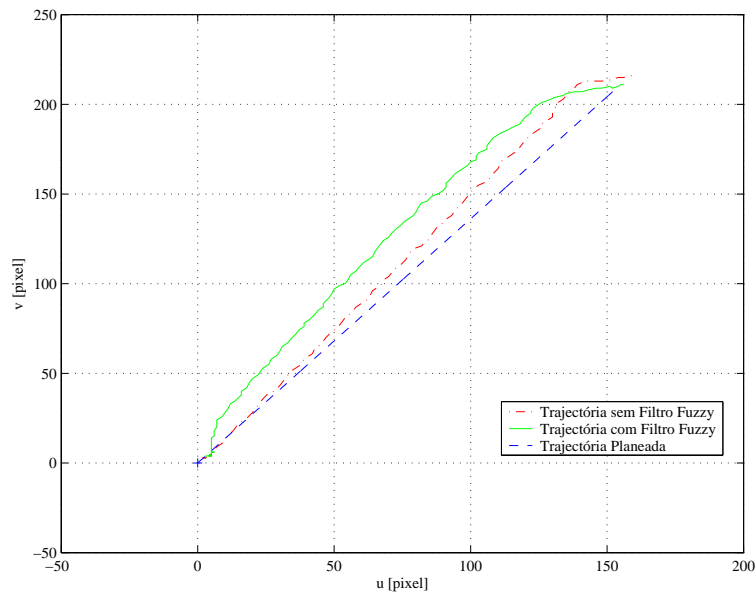


Figura 6.15: Comparação entre a trajetória planeada e as trajetórias com e sem a utilização de filtros *fuzzy*, com e sem a utilização de filtros *fuzzy*. Cheio - com filtros *fuzzy*. Traço-ponto - sem filtros *fuzzy*. Traço interrompido - trajetória planeada

### 6.3.3 Conclusões da Aplicação de Filtros *Fuzzy*

A aplicação de filtros *fuzzy* ao controlo visual cinemático como foi realizada na secção 6.3, permitiu comprovar a sua aplicabilidade e os resultados teóricos desejados e previstos na secção 5.2. Assim, os filtros *fuzzy* permitem melhorar o desempenho das leis de controlo visual cinemático baseadas na imagem.

No caso do regulador, ao serem utilizados filtros *fuzzy*, o tursor de velocidade da câmara e as velocidades de junta do robô sofrem uma diminuição nos instantes iniciais do controlo visual, quando comparados com a sua não utilização. Conseguiu-se também diminuir as oscilações indesejáveis nas velocidades de junta e da câmara, devido ao controlo visual ser inerentemente discreto. Como também seria de esperar os melhores resultados foram obtidos quando o tempo de amostragem vídeo é baixo, como verificado experimentalmente. Os resultados foram verificados em simulação para o caso genérico de seis graus de liberdade e no aparato experimental utilizado na presente tese. Foi também verificado que a trajetória das características do objecto no plano da imagem deixaria de ser linear

e passaria a ser quadrática, quando um função de pertinência triangular fosse utilizada.

Quando os filtros *fuzzy* são aplicados ao caso de planeamento de trajectórias, verificou-se a diminuição das velocidades de junta fornecidas pelo controlador. Neste caso a margem de melhoria é menor quando comparada com o caso do regulador, pois o erro entre as características do objecto na imagem entre dois instantes de tempo  $k$  consecutivos é também menor. No entanto a amplitude local das velocidades diminuiu bem como a sua amplitude nos instantes iniciais.

## 6.4 Modelação Fuzzy

Na presente secção são apresentados e discutidos os resultados obtidos no processo de identificação do modelo inverso *fuzzy* do aparato experimental, conforme descrito na sub-secção 5.3.1. Seguindo a metodologia então definida é em primeiro lugar necessário definir a trajectória sobre a qual os dados de entrada/saída, necessários à identificação do modelo inverso, serão observados e guardados. Posteriormente esses mesmos dados serão utilizados no processo de identificação *off-line*.

A trajectória escolhida move o robô, em linha recta, entre a posição de junta inicial  $\mathbf{q}_i = [-\frac{\pi}{2}; \frac{\pi}{2}]$  e a posição de junta final  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$  durante oito segundos. Estas posições de junta referem-se às posições 2 e 3 definidas na sub-secção 6.2. As velocidades inicial e final das juntas do robô são nulas, para a trajectória referida. As medições das variáveis necessárias à identificação do modelo são obtidas directamente do aparato experimental na configuração *eye-in-hand*, quando este se encontra a realizar o movimento com um controlador PID de posição de junta.

O conjunto de pontos observados e guardados para identificar o modelo inverso *fuzzy* contém 575 elementos, com o tempo de amostragem de visão 0.02[seg]. A figura 6.16 apresenta os dados de entrada, i.e. as posições de junta  $q_1(k)$  e  $q_2(k)$ , e a variação das características do objecto na imagem  $\delta s_x(k)$  e  $\delta s_y(k)$ , usadas para a identificação.

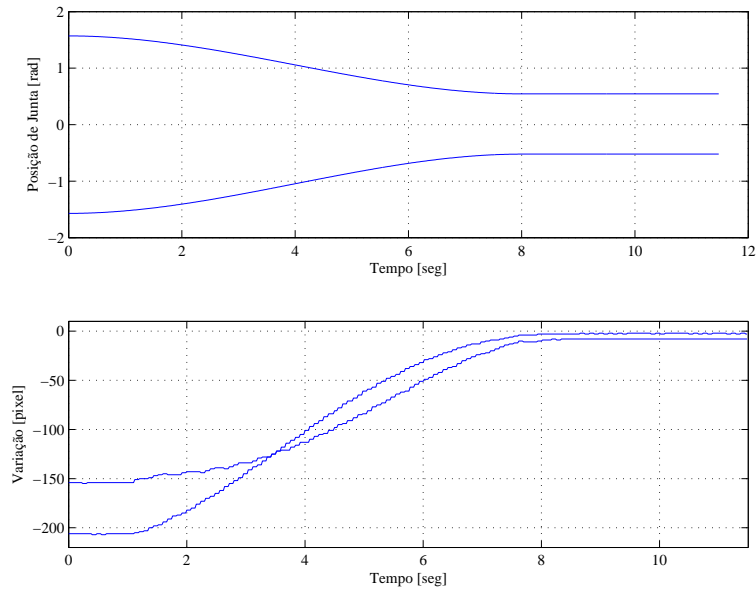


Figura 6.16: Dados de entrada/saída para a identificação do modelo inverso. Cima: posições de junta. Baixo: variação das características do objecto na imagem.

Note-se ainda que para identificar o modelo inverso de um sistema, não se pode simplesmente trocar as entradas com as saídas e as saídas com as entradas. Uma vez que o modelo inverso, como descrito em (5.23), é um sistema não-causal, a saída do modelo inverso deverá ser atrasada um passo no tempo [124].

Os resultados da validação do modelo inverso *fuzzy* obtido são apresentados na figura 6.17, nomeadamente a saída do modelo, velocidades de junta  $\delta\mathbf{q}$ . Com esta abordagem são identificados dois modelos, um para cada velocidade de junta, uma vez que a modelação *fuzzy* foi apresentada na secção 5.3 para sistemas MISO. Considerando o critério de performance *variance accounted for* (VAF), os modelos identificados obtiveram VAFs de 99,01% e 87,95% para a velocidade da junta 1 e velocidade da junta 2, respectivamente. Quando VAF toma o valor de 100%, existe uma correspondência perfeita entre o modelo identificado e o sistema real. Através dos resultados apresentados verifica-se que o modelo inverso *fuzzy* para a velocidade da junta 1 é preciso, mas o nível de precisão diminui no modelo inverso *fuzzy* para a velocidade da junta 2. No entanto os modelos identificados são suficientes para obter um controlador preciso, como se irá verificar na secção 6.5.

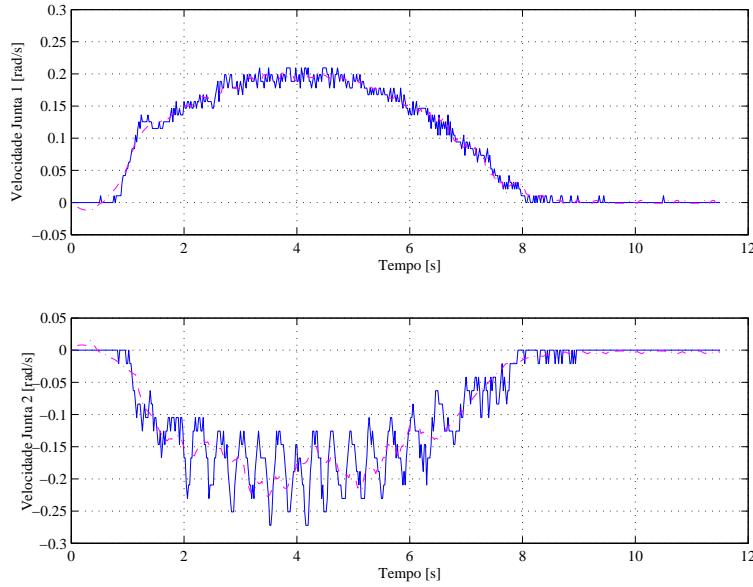


Figura 6.17: Validação do modelo inverso *fuzzy* (velocidades de junta). Cheio – valor real do dado de saída. Traço-ponto – saída do modelo inverso *fuzzy*.

Relativamente aos parâmetros dos modelos inverso *fuzzy*, quatro regras (*clusters*) revelaram-se suficientes para a obtenção dos bons resultados descritos no parágrafo anterior. Assim o modelo inverso identificado *fuzzy* tem oito regras no total, quatro para cada saída,  $\delta q_1$  e  $\delta q_2$ .

Em seguida são apresentadas as regras para o modelo inverso *fuzzy* associado à velocidade de junta  $\delta q_1$ :

1. **If**  $\delta q_1(k-1)$  is  $A_{11}$  **and**  $q_1$  is  $A_{12}$  **and**  $q_2$  is  $A_{13}$   
**and**  $\delta s_x$  is  $A_{14}$  **and**  $\delta s_y$  is  $A_{15}$  **then**

$$\delta q_1(k) = 3.68 \cdot 10^{-1} \delta q_1(k-1) + 1.17 q_1 + \\ + 0 \cdot q_2 - 5.10 \cdot 10^{-3} \delta s_x + 2.91 \cdot 10^{-3} \delta s_y + 5.65 \cdot 10^{-1} \quad (6.2)$$

2. **If**  $\delta q_1(k-1)$  is  $A_{21}$  **and**  $q_1$  is  $A_{22}$  **and**  $q_2$  is  $A_{23}$   
**and**  $\delta s_x$  is  $A_{24}$  **and**  $\delta s_y$  is  $A_{25}$  **then**

$$\delta q_1(k) = 8.88 \cdot 10^{-1} \delta q_1(k-1) + 2.61 \cdot 10^{-1} q_1 + \\ + 0 \cdot q_2 - 4.36 \cdot 10^{-6} \delta s_x - 8.72 \cdot 10^{-4} \delta s_y + 2.26 \cdot 10^{-1} \quad (6.3)$$

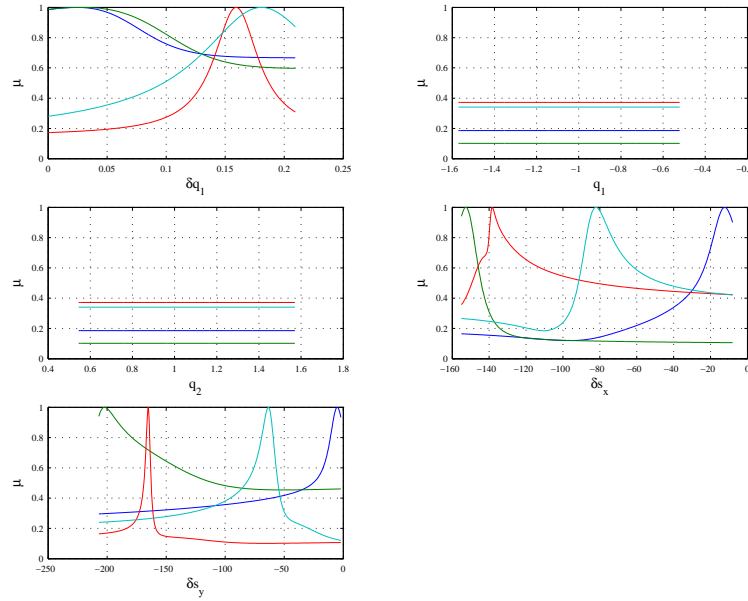


Figura 6.18: Funções de pertinência  $A_{ij}$  obtidas para a saída  $\delta q_1$ .

3. **If**  $\delta q_1(k-1)$  is  $A_{31}$  **and**  $q_1$  is  $A_{32}$  **and**  $q_2$  is  $A_{33}$   
**and**  $\delta s_x$  is  $A_{34}$  **and**  $\delta s_y$  is  $A_{35}$  **then**

$$\begin{aligned} \delta q_1(k) = & -9.55 \cdot 10^{-2} \delta q_1(k-1) - 8.57 \cdot 10^{-2} q_1 + \\ & + 0 \cdot q_2 - 3.53 \cdot 10^{-3} \delta s_x + 2.66 \cdot 10^{-3} \delta s_y + 3.74 \cdot 10^{-3} \end{aligned} \quad (6.4)$$

4. **If**  $\delta q_1(k-1)$  is  $A_{41}$  **and**  $q_1$  is  $A_{42}$  **and**  $q_2$  is  $A_{43}$   
**and**  $\delta s_x$  is  $A_{44}$  **and**  $\delta s_y$  is  $A_{45}$  **then**

$$\begin{aligned} \delta q_1(k) = & 1.06 \cdot 10^{-1} \delta q_1(k-1) + 1.27 \cdot 10^{-1} q_1 + \\ & + 0 \cdot q_2 - 3.83 \cdot 10^{-3} \delta s_x + 1.47 \cdot 10^{-3} \delta s_y + 2.42 \cdot 10^{-2} \end{aligned} \quad (6.5)$$

As funções de pertinência obtidas através da projecção dos *clusters* para a saída do modelo  $\delta q_1$ , são apresentadas na figura 6.18.

As regras para o modelo inverso *fuzzy* associado à velocidade de junta  $\delta q_2$ , são as seguintes:

1. **If**  $\delta q_2(k-1)$  is  $A_{11}$  **and**  $q_1$  is  $A_{12}$  **and**  $q_2$  is  $A_{13}$   
**and**  $\delta s_x$  is  $A_{14}$  **and**  $\delta s_y$  is  $A_{15}$  **then**

$$\begin{aligned} \delta q_2(k) = & 8.84 \cdot 10^{-1} \delta q_2(k-1) + 5.81 \cdot 10^{-1} q_1 + \\ & + 0 \cdot q_2 + 3.08 \cdot 10^{-3} \delta s_x - 4.09 \cdot 10^{-3} \delta s_y + 5.14 \cdot 10^{-1} \end{aligned} \quad (6.6)$$

2. **If**  $\delta q_2(k-1)$  is  $A_{21}$  **and**  $q_1$  is  $A_{22}$  **and**  $q_2$  is  $A_{23}$   
**and**  $\delta s_x$  is  $A_{24}$  **and**  $\delta s_y$  is  $A_{25}$  **then**

$$\begin{aligned} \delta q_2(k) = & 8.16 \cdot 10^{-1} \delta q_2(k-1) - 6.62 \cdot 10^{-1} q_1 + \\ & + 0 \cdot q_2 + 5.25 \cdot 10^{-3} \delta s_x - 1.12 \cdot 10^{-3} \delta s_y - 2.32 \cdot 10^{-1} \end{aligned} \quad (6.7)$$

3. **If**  $\delta q_2(k-1)$  is  $A_{31}$  **and**  $q_1$  is  $A_{32}$  **and**  $q_2$  is  $A_{33}$   
**and**  $\delta s_x$  is  $A_{34}$  **and**  $\delta s_y$  is  $A_{35}$  **then**

$$\begin{aligned} \delta q_2(k) = & 7.20 \cdot 10^{-1} \delta q_2(k-1) - 2.87 \cdot 10^{-1} q_1 + \\ & + 0 \cdot q_2 - 3.59 \cdot 10^{-3} \delta s_x + 1.50 \cdot 10^{-3} \delta s_y - 6.92 \cdot 10^{-1} \end{aligned} \quad (6.8)$$

4. **If**  $\delta q_2(k-1)$  is  $A_{41}$  **and**  $q_1$  is  $A_{42}$  **and**  $q_2$  is  $A_{43}$   
**and**  $\delta s_x$  is  $A_{44}$  **and**  $\delta s_y$  is  $A_{45}$  **then**

$$\begin{aligned} \delta q_2(k) = & 7.28 \cdot 10^{-1} \delta q_2(k-1) - 7.93 \cdot 10^{-2} q_1 + \\ & + 0 \cdot q_2 + 8.46 \cdot 10^{-4} \delta s_x + 1.46 \cdot 10^{-3} \delta s_y - 3.22 \cdot 10^{-2} \end{aligned} \quad (6.9)$$

As funções de pertença obtidas através da projecção dos *clusters* para a saída do modelo  $\delta q_2$ , são apresentadas na figura 6.19.

## 6.5 Controlo Fuzzy

Na presente secção são apresentados e discutidos os resultados da aplicação do controlo baseado no modelo inverso *fuzzy*, utilizando compensação *fuzzy*. Os resultados obtidos por esta abordagem serão ainda comparados com o resultado da aplicação do controlo

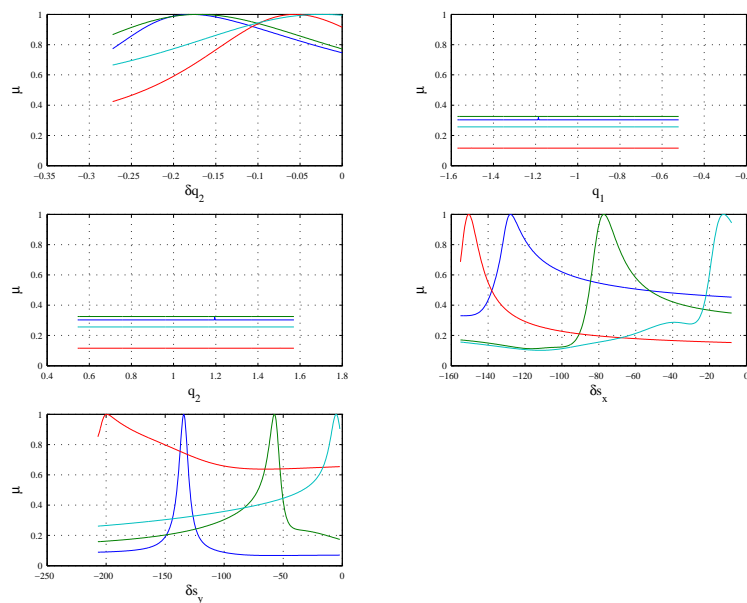


Figura 6.19: Membership functions  $A_{ij}$  obtained for the output  $\delta q_2$ .

visual cinemático baseado na imagem, com planeamento da trajectória. Ambas as implementações foram realizadas no aparato experimental, com a configuração *eye-in-hand*, utilizado nesta tese.

A trajectória linear definida no plano da imagem para o controlo visual cinemático baseado na imagem, tem como posição inicial e posição final respectivamente as posições 2 e 3 definidas na sub-secção 6.2, a que correspondem as posições no espaço de junta  $\mathbf{q}_i = [-\frac{\pi}{2}; \frac{\pi}{2}]$ ,  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$  e no plano da imagem,  $\mathbf{s} = [154; 213]$  e  $\mathbf{s}^* = [0; 0]$ . Que coincidem com os pontos iniciais e finais da trajectória também linear utilizada na identificação do modelo inverso *fuzzy*. Ambas as trajectórias têm a duração de oito segundos.

A comparação entre o controlo baseado no modelo inverso *fuzzy* e o controlo visual cinemático baseado na imagem é apresentada na figura 6.20. Nesta figura pode-se verificar que ambos os controladores atingem a vizinhança da posição desejada, no que respeita às características do objecto na imagem. Esta vizinhança é definida num intervalo de 10 pixels em torno da posição desejada, que corresponde à zona morta dos actuadores das juntas. Todavia e como se pode verificar na figura 6.22, a trajectória devida ao controlo

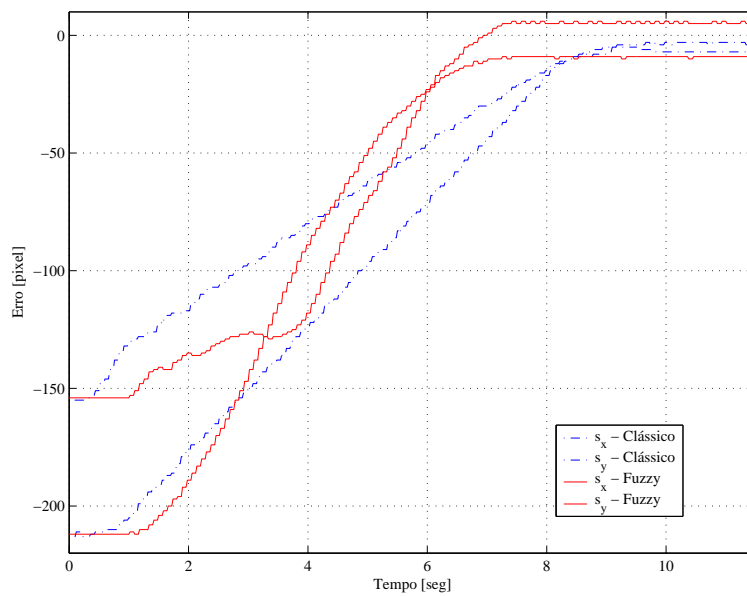


Figura 6.20: Erro das características do objecto na imagem,  $\mathbf{e}$ , durante o controlo. Cheio – controlo *fuzzy*. Traço-ponto – controlo visual cinemático baseado na imagem.

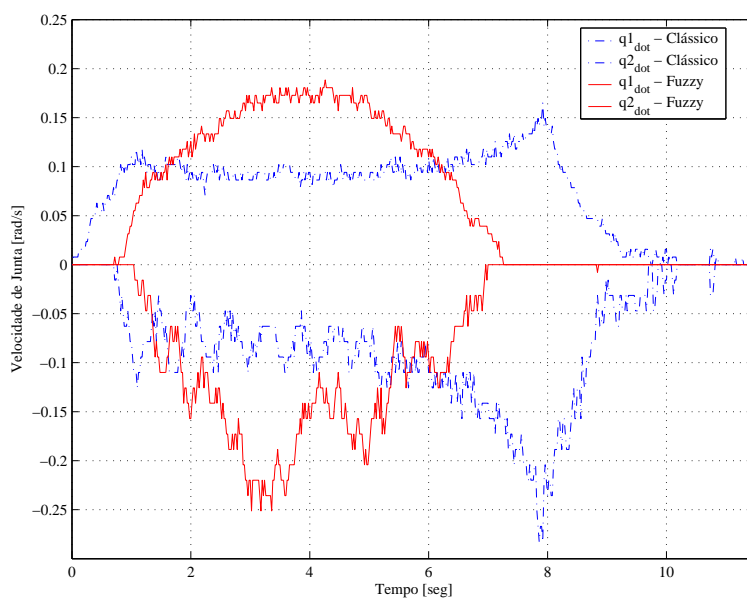


Figura 6.21: Velocidade das juntas do robô,  $\dot{\mathbf{q}}$ , durante o controlo. Cheio – controlo *fuzzy*. Traço-ponto – controlo visual cinemático baseado na imagem.



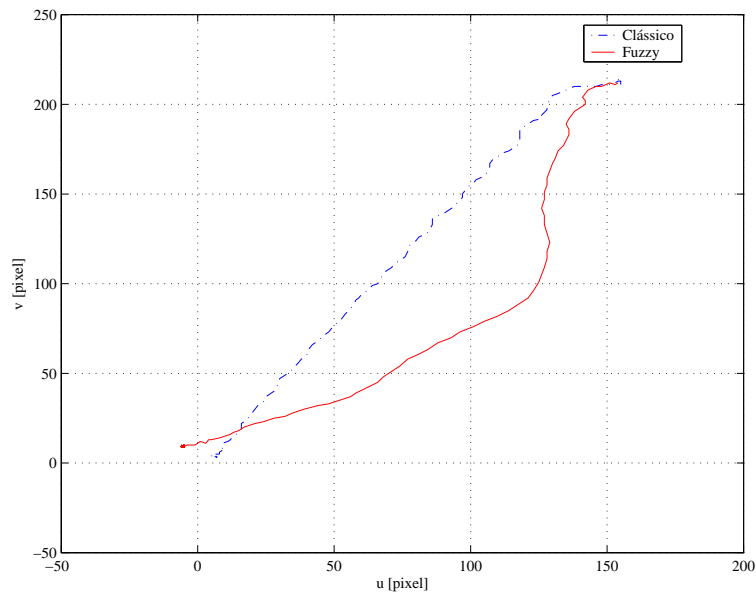


Figura 6.22: Trajectória das características do objecto na imagem  $s$ , no plano da imagem. Cheio – controlo *fuzzy*. Traço-ponto – controlo visual cinematográfico baseado na imagem.

baseado no modelo inverso *fuzzy* não é uma linha recta e também não corresponde à obtida pelo controlo visual clássico baseado na imagem com trajectória planeada. Tal facto deve-se a que a trajectória utilizada para a identificação do modelo, não ser linear no plano da imagem, mas sim no espaço de junta do robô, como descrito na secção 6.4.

Para melhor comparar as das abordagens é necessário verificar a evolução das velocidades de junta ao longo da trajectória, para concluir acerca da sua suavidade. Assim, na figura 6.21 são apresentadas as referidas velocidades, onde se pode verificar que a aplicação do controlo visual cinematográfico baseado na imagem apresenta maiores oscilações na execução da mesma trajectória. Ainda se pode verificar que no final da trajectória planeada, oito segundos, o controlo cinematográfico baseado na imagem ainda não atingiu o regime estacionário. A posição desejada só é atingida ao fim de 10 segundos. Para aproximar o tempo de estabelecimento dos dois controladores em torno dos oito segundos, bastaria aumentar o ganho proporcional do controlador cinematográfico baseado na imagem. Todavia esta alteração iria provocar maiores oscilações nas velocidades de junta do robô, o que seria de todo indesejável do ponto de vista de controlo.

Os resultados experimentais obtidos nesta secção, figuras 6.20 e 6.22 mostram que o controlo baseado no modelo inverso *fuzzy* permite alcançar a posição desejada no plano da imagem, tal como a abordagem clássica do controlo visual cinemático baseado na imagem. Todavia, o comportamento das velocidades de junta e o tempo necessário para atingir o regime estacionário, levam à conclusão que a performance do controlador baseado no modelo *fuzzy* é melhor. Conclui-se também que com este tipo de controlo os problemas clássicos associados ao jacobiano e à calibração são ultrapassados, pois não é necessário obter e inverter o jacobiano analítico, nem tão pouco calibrar o sistema robô-câmara.

## 6.6 Controlo Visual Dinâmico

Na presente secção são apresentados os resultados experimentais da aplicação das leis de controlo definidas no capítulo 4, i.e. a extensão ao controlador proposto por Kelly [76] e o controlo visual dinâmico baseado em posição. Os resultados obtidos por ambos os controladores são comparados entre si e com o controlo visual cinemático baseado na imagem. A tarefa a realizar por ambos os controladores é movimentar o robô entre as posições 1 e 3 definidas na sub-secção 6.2. Estas posições correspondem, no espaço de junta  $\mathbf{q}_i = [0; 0]$ ,  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$  e no plano da imagem,  $\mathbf{s} = [-143; -27]$  e  $\mathbf{s}^* = [0; 0]$ . Saliente-se que estes resultados podem ser comparados com os obtidos pelo controlo visual cinemático baseado na imagem, ver sub-secção 6.7.1.

Uma vez que o robô do aparato experimental se move num plano horizontal, é possível simplificar ambos os controladores descritos no capítulo 4. Isto é, não existem forças gravíticas a perturbar o sistema e nas equações (4.2) e (4.9), os respectivos termos são nulos.

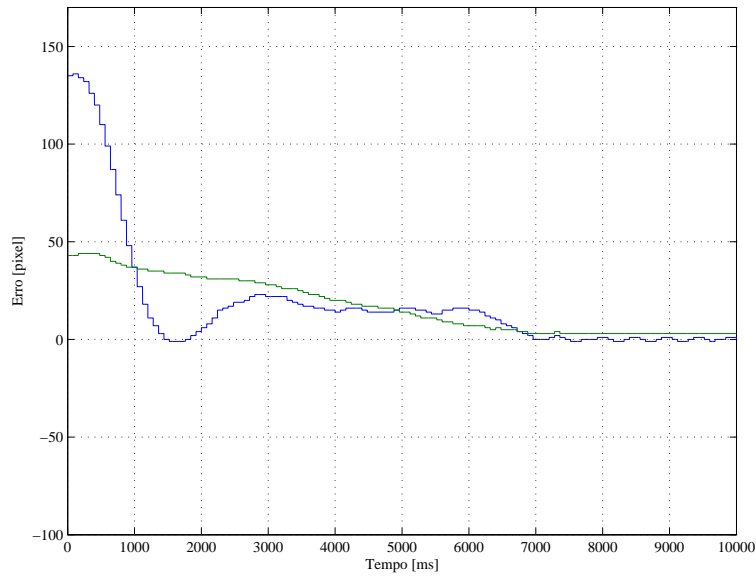


Figura 6.23: Erro das características do objecto na imagem  $s$ , no plano da imagem, com a extensão do controlador dinâmico - Kelly.

### 6.6.1 Extensão do Controlador Dinâmico - Kelly

A extensão ao controlador proposto por Kelly [76], foi especialmente projectada para um robô planar de dois graus de liberdade, sendo que apenas um ponto do alvo é necessário ao controlo. Na experiência apresentada nesta sub-secção, a matriz de rotação é calculada utilizando a expressão:

$${}^c\mathbf{R}_o = ({}^0\mathbf{R}_e \cdot {}^e\mathbf{R}_c)^{-1} \cdot {}^0\mathbf{R}_o$$

em que  ${}^0\mathbf{R}_e$  é calculada utilizando a cinemática directa do robô, e  ${}^e\mathbf{R}_c$  foi definida na sub-secção 6.2. Este cálculo é possível pois  ${}^0\mathbf{R}_o$  foi medido manualmente e é definido por:

$${}^0\mathbf{R}_o = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Na experiência apresentada foram escolhidos os ganhos  $\mathbf{K}_p = 0.115$  e  $\mathbf{K}_d = 2$  por forma a minimizar o sobre-impulso da resposta. Ao aumentar  $\mathbf{K}_p$  a resposta é mais rápida mas com maior sobre-impulso. Esta escolha melhora ainda a trajectória na imagem, uma vez que o sobre-impulso não é elevado.

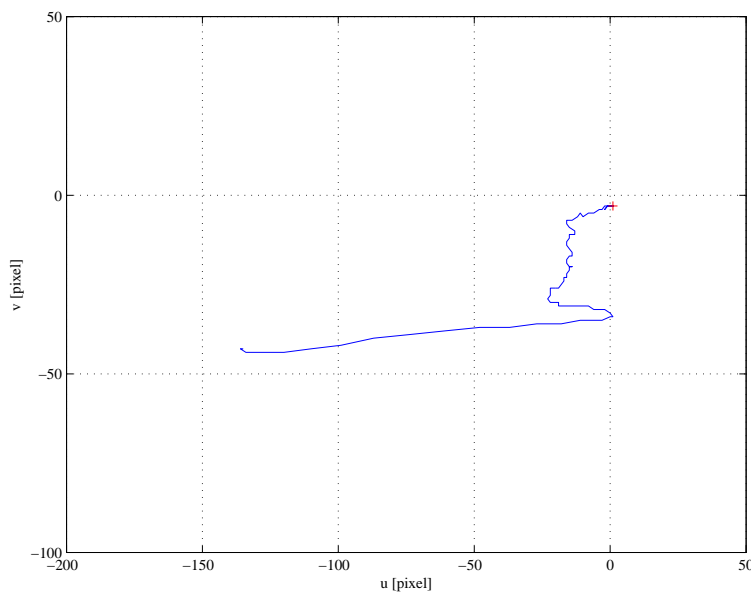


Figura 6.24: Trajectória das características do objecto na imagem  $s$ , no plano da imagem, com a extensão do controlador dinâmico - Kelly. '+' posição final.

Na figura 6.23 é apresentado o erro das características do objecto na imagem  $s$ , no plano da imagem, em que não se verifica sobre-impulso. A correspondente trajectória na imagem é apresentada na figura 6.24, sendo que o erro perto da posição final é de três pixels, em valor absoluto.

### 6.6.2 Controlo Visual Dinâmico, *eye-in-hand*, baseado em posição

O controlo visual dinâmico apresentado na presente tese é verificado experimentalmente na presente sub-secção, através dos resultados apresentados, considerando o erro da pose do objecto relativamente à câmara e a trajectória dos pontos na imagem (necessários para a reconstrução *on-line* da pose).

A pose do objecto relativamente à câmara, a que corresponde a posição 3 definida na sub-secção 6.2, é dada pelo vector:

$$\mathbf{s}_{3D}^* = [ 0.092 \quad -0.039 \quad 1.1 \quad 0 \quad 0 \quad 1.5 ]^T$$

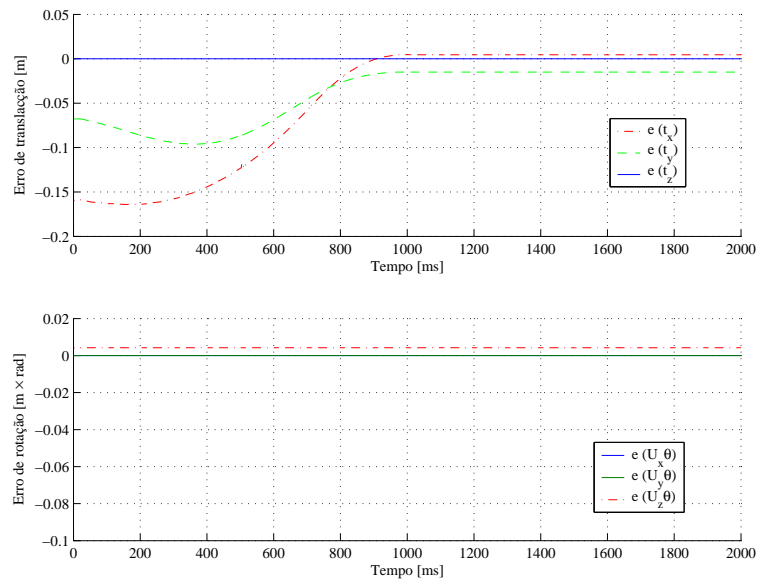


Figura 6.25: Erro das características do objecto na imagem  $s$ , no plano da imagem, com o controlador visual dinâmico baseado em posição.

Uma vez que o eixo óptico da câmara é paralelo ao eixo  $Z$  do robô e perpendicular ao plano do objecto e do movimento do robô, a projecção perspectiva fraca é uma boa aproximação à projecção perspectiva total. Assim o algoritmo de Dementhon & Davis, descrito na sub-secção 2.4.2, é utilizado para estimar a pose do objecto relativamente à câmara.

As características especiais do aparato experimental, i.e. o plano da imagem ser paralelo ao plano do movimento do robô, permitem simplificar o algoritmo de estimação da pose. A terceira coordenada do vector de erro (3.22) é constante durante o movimento do robô, pois a distância entre os dois planos é constante. A quarta e quinta coordenadas do mesmo vector de erro são nulas, pois não existe movimento de rotação segundo esses eixos.

Através da figura 6.25 verifica-se que o movimento de rotação é praticamente nulo, para a tarefa especificada, pois a sexta coordenada do vector de erro se manteve constante durante o movimento. Verifica-se portanto que o principal movimento foi devido ao movimento de translação no plano do robô.

Na figura 6.25 é apresentado o erro das características do objecto na imagem  $s$ , no plano

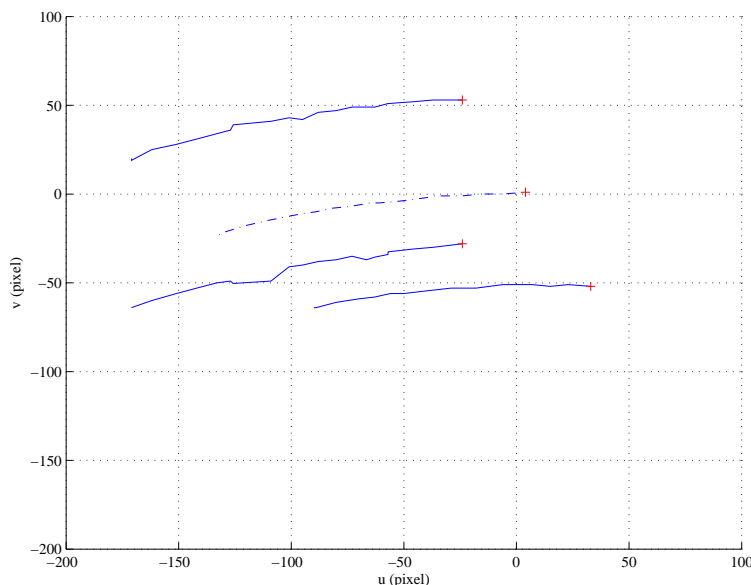


Figura 6.26: Trajectória dos pontos necessários à reconstrução da pose, com o controlador visual dinâmico baseado em posição. '+' posição final.

da imagem. A correspondente trajectória na imagem dos quatro pontos utilizados para reconstruir a rotação e translação é apresentada na figura 6.26, sendo que o erro perto da posição final é de cinco pixels em valor absoluto.

### 6.6.3 Conclusões sobre o Controlo Visual Dinâmico

Nesta secção foram apresentados os resultados obtidos com os controladores apresentados no capítulo 4. Estes dois controladores constituem a contribuição ao controlo visual dinâmico de robôs manipuladores. A primeira contribuição foi a extensão ao controlador de Kelly [76], para a configuração *eye-in-hand*. A segunda contribuição foi o controlador visual dinâmico baseado em posição.

É oportuno nesta sub-secção constatar que a extensão ao controlador de Kelly [76], permite controlar o robô sem a utilização dos parâmetros intrínsecos da câmara e da profundidade  $Z$ , o que não sucede com o controlador proposto por Kelly [77]. No entanto a abordagem proposta nesta tese apenas é aplicável a um robô de dois graus de liberdade, em contraste

com o caso mais geral do controlador proposto por Kelly [77]. O controlador visual dinâmico baseado em posição, proposto nesta tese, tem a particularidade de ser também aplicável a um robô com mais de dois graus de liberdade.

Ambos os controladores propostos têm como principal vantagem, relativamente ao controlo visual cinemático, o facto de o jacobiano não ter que ser invertido. Apenas é utilizado o jacobiano transposto em ambas as leis de controlo. A escolha entre os dois controladores propostos depende da aplicação em estudo e das características do objecto na imagem disponíveis. Uma desvantagem da aplicação do controlo visual dinâmico aos robôs actualmente existentes no mercado, é o facto de que nestes robôs não estar disponível o acesso aos binários dos robôs para que se possa actuar directamente nos motores.

## 6.7 Abordagens Clássicas

### 6.7.1 Controlo Visual Cinemático Baseado na Imagem

Na presente sub-secção são apresentados os resultados obtidos experimentalmente durante o controlo visual cinemático baseado na imagem do robô manipulador presente no aparato experimental, anexo B. A tarefa a realizar pelo controlador consiste em movimentar o robô entre as posições 1 e 3 definidas na sub-secção 6.2. Estas posições correspondem as posições no espaço de junta  $\mathbf{q}_i = [0; 0]$ ,  $\mathbf{q}_f = [-\frac{\pi}{6}; \frac{\pi}{6}]$  e no plano da imagem,  $\mathbf{s} = [-143; -27]$  e  $\mathbf{s}^* = [0; 0]$ . Os resultados obtidos podem ainda ser comparados com os obtidos durante o controlo visual dinâmico (extensão do controlador Kelly [76]).

Em [55], foram apresentados os resultados obtidos, durante o controlo visual baseado na imagem, com a utilização de vários tipos de controladores PID e com diferentes números de características do objecto na imagem, neste caso pontos. Na presente tese não se irão apresentar todos os resultados, apenas os obtidos pelo controlador proporcional (3.20) com ganho  $\lambda = 0.95$  e com um ponto como característica do objecto na imagem. Saliente-se

que com a introdução de um controlador PID a equação (3.20) fica:

$$\dot{\mathbf{q}} = -\hat{\mathbf{J}}_{2D}^{-1} \cdot \left( \lambda \cdot (\mathbf{s} - \mathbf{s}^*) + \mathbf{K}_d \frac{d(\mathbf{s} - \mathbf{s}^*)}{dt} + \mathbf{K}_i \cdot \int (\mathbf{s} - \mathbf{s}^*) dt \right) \quad (6.10)$$

Na figura 6.27 é apresentado o erro das características do objecto na imagem, durante o controlo efectuado pelo controlador visual cinemático baseado na imagem. Verifica-se que existe um decaimento exponencial do erro, após um sobre-impulso inicial. Depois do sobre-impulso a trajectória das características no plano da imagem aproxima-se a uma linha recta, figura 6.28, como é característica deste tipo de controlador. As velocidades de junta do robô têm o comportamento esperado, figura 6.29, pois para o caso do regulador a velocidade inicial é alta e diminui até ao valor zero, quando perto da posição desejada. Note-se no gráfico que, quando no regime estacionário, a velocidade não é nula pois o erro das características do objecto na imagem também não é nulo. Assim a lei de controlo visual indica uma velocidade a seguir não nula, mas que o robô não consegue seguir devido às características deste (não-linearidades). O erro verificado nas características da imagem é na ordem dos 5 pixels, em valor absoluto, o que vem em consonância com os resultados obtidos como todos os outros controladores.

### 6.7.2 Controlo Visual Cinemático Híbrido

Na presente sub-secção é apresentado o resultado obtido para a aplicação do controlo visual híbrido ao robô planar existente no aparato experimental. Para tal foi utilizada uma *toolbox de visual servoing* [51, 52] desenvolvida para simular as leis de controlo antes da passagem para a plataforma experimental. A *toolbox* contém também o modelo do robô planar. Saliente-se aqui o facto de que esta permite a utilização dos algoritmos desenvolvidos na fase de simulação directamente no *software* de controlo do aparato experimental, como descrito no anexo B.

A lei de controlo visual híbrido simulada na *toolbox de visual servoing* foi a desenvolvida por Malis,  $2\frac{1}{2}D$  *visual servoing* [91]. Por forma a ser possível obter resultados para este



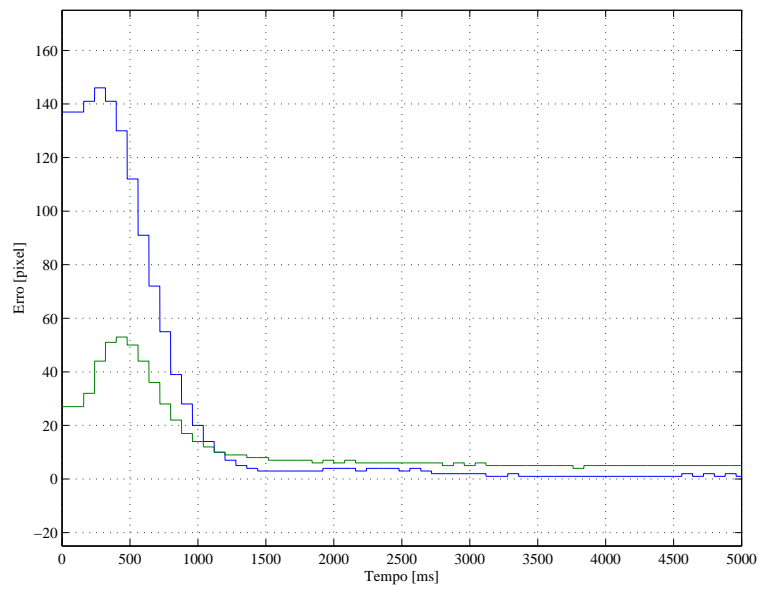


Figura 6.27: Erro das características do objecto na imagem  $s$ , no plano da imagem, com o controlador visual baseado na imagem.

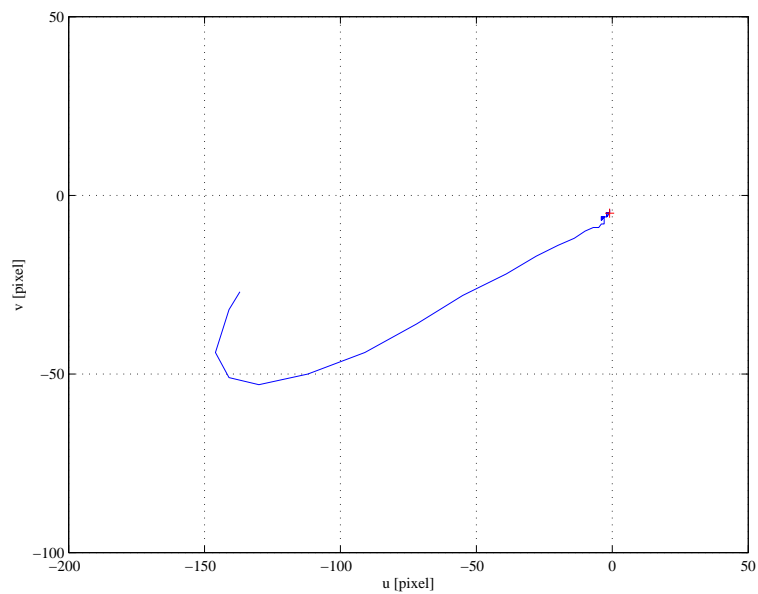


Figura 6.28: Trajectória das características do objecto na imagem  $s$ , no plano da imagem, com o controlador visual baseado na imagem. '+' posição final.

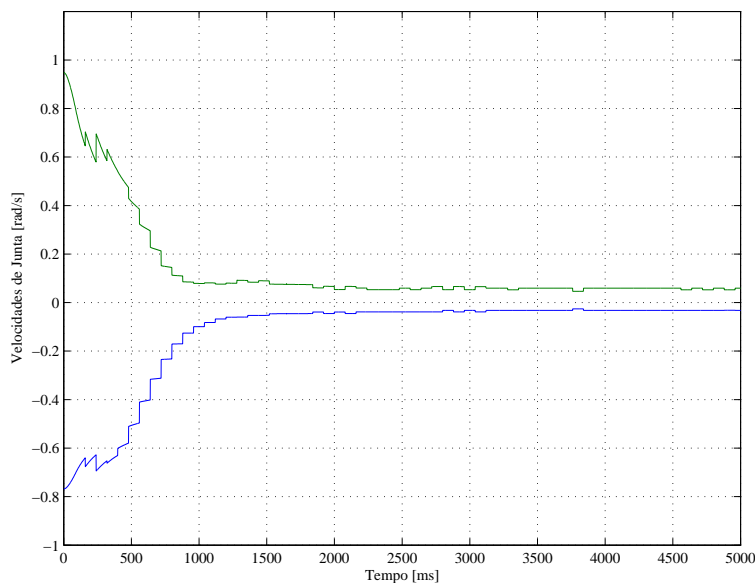


Figura 6.29: Velocidades das juntas dos robô, com o controlador visual baseado na imagem.

tipo de controlador foi necessário alterar a configuração do sistema robô-câmara, i.e. a câmara está a olhar para a frente e colocada no seguimento do segundo elo do robô. Assim a profundidade  $Z$  deixa de ser constante, como o foi na configuração utilizada no aparato experimental. Saliente-se que a câmara utilizada é a mesma das experiências anteriores.

Em [51] foram apresentados os resultados obtidos para o controlo visual híbrido  $2\frac{1}{2}D$ , quando a estimação da rotação e da translacção foi obtida com base na matriz de homografia e fundamental. Os resultados apresentados nesta tese dizem respeito aos obtidos pela matriz de homografia, visto que o objecto utilizado é planar. Uma vez que o robô não tem punho não é necessário controlar o movimento rotação, sendo que o vector de erro se resume às três primeiras coordenadas de (3.35), i.e. as coordenadas do ponto e a relação de profundidades.

A posição desejada dos pontos na imagem em coordenadas da imagem é definida por:

$$\mathbf{s}^* = \begin{pmatrix} -13 & -156 & -13 & 156 & 295 & -160 & 295 & 160 \end{pmatrix}^T$$

sendo que a posição inicial é definida por:

$$\mathbf{s}_i = ( -53 \quad -163 \quad -56 \quad 163 \quad 258 \quad -163 \quad 258 \quad 163 )^T$$

Verifica-se que utilizando a lei de controlo híbrido  $2\frac{1}{2}D$ , o robô atinge a posição desejada com um decaimento exponencial das características do objecto na imagem, como se pode verificar na figura 6.30. Na figura 6.31 pode-se confirmar uma trajectória linear dos quatro pontos utilizados para estimar a homografia, pois apenas pode existir movimento na direcção  $u$  da imagem, devido à configuração da câmara. O erro verificado na direcção  $v$  deve-se a que a simulação da captura da imagem tem um erro aleatório associado de no máximo 10 pixels, com distribuição normal. Na figura 6.32 pode verificar-se o andamento das velocidades de junta requeridas ao robô, i.e. saídas da lei de controlo visual. Nota-se que quando o robô se aproxima da posição desejada as velocidades tomam valores oscilatórios, uma vez que a estimação da matriz de homografia e também da matriz fundamental piora a sua performance. A forma de obviar este problema é, quando o robô se aproximar da posição desejada, i.e. dentro de uma vizinhança pré-definida, comutar o controlo para o controlo visual cinemático baseado na imagem. Assim evita-se a instabilidade na estimação da matriz de homografia ou fundamental, junto da posição desejada. Note-se que como o robô está próximo da posição desejada, o controlo visual cinemático baseado na imagem pode ser utilizado com garantias de convergência.

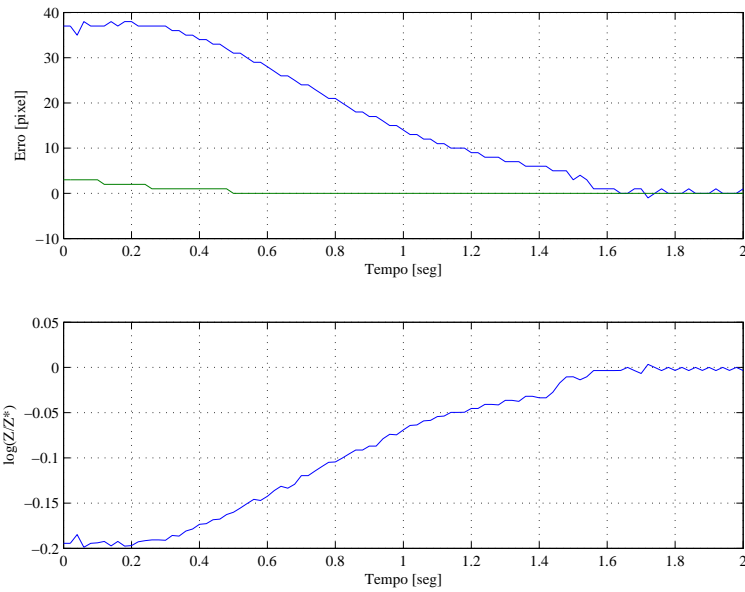


Figura 6.30: Erro das características do objecto na imagem  $\mathbf{s}$ , no plano da imagem, com o controlador visual híbrido ( $2\frac{1}{2}D$ ).

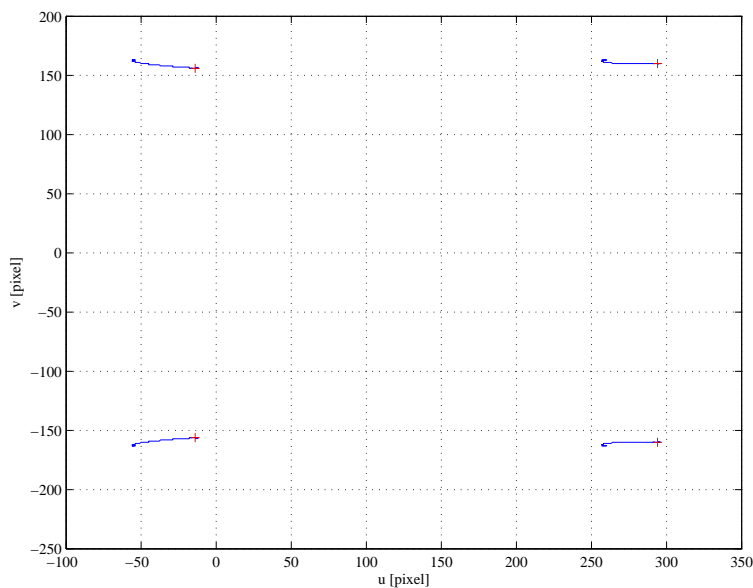


Figura 6.31: Trajectória das características do objecto na imagem  $\mathbf{s}$ , no plano da imagem, com o controlador visual híbrido ( $2\frac{1}{2}D$ ). '+' posição final.

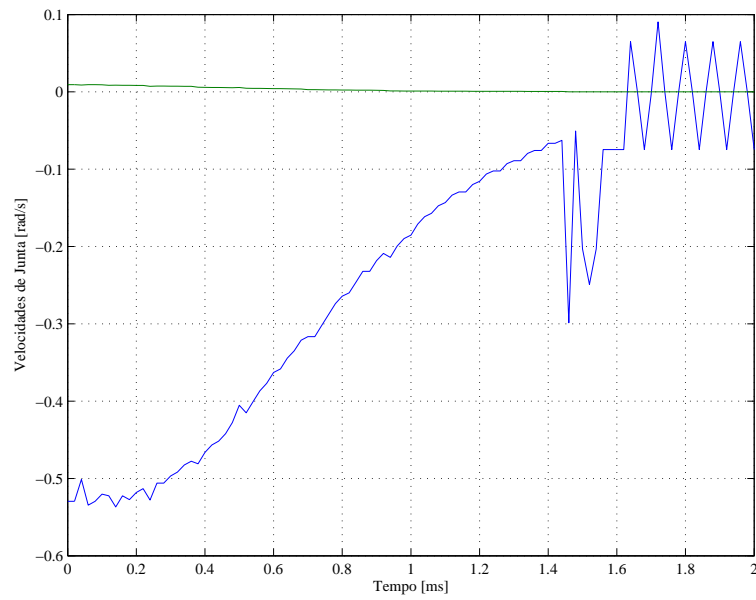


Figura 6.32: Velocidades das juntas dos robô, com o controlador visual híbrido ( $2\frac{1}{2}D$ ).



# Capítulo 7

## Conclusões

O presente capítulo apresenta uma breve sùmula do trabalho e das contribuições apresentadas ao longo desta tese. São ainda apresentados e discutidos alguns aspectos relevantes sobre os temas abordados na presente tese, passíveis de desenvolvimentos futuros.

### 7.1 Sumário e Contribuições

A presente tese abordou o controlo visual de robôs manipuladores, sendo propostas várias soluções validadas num aparato experimental desenvolvido para o efeito.

No capítulo um foi introduzido o tema e o porquê da utilização do controlo visual de robôs. No capítulo dois foi apresentada a teoria e algoritmos de visão por computador, necessários para a execução das soluções de controlo visual propostas na tese. Particularmente importante é o algoritmo de estimação da pose de um objecto relativamente à câmara, e os algoritmos de estimação das matrizes fundamental e homografia, para posterior estimação do movimento da câmara.

O estado da arte sobre o controlo visual de robôs manipuladores foi apresentado no capítulo três. Em primeiro lugar foram abordados os algoritmos clássicos baseados em imagem e posição, também designados por 2D e 3D respectivamente, para em seguida

apresentar as leis de controlo híbridas. Actualmente tem-se verificado um grande interesse por estas últimas, pois conseguem conjugar vários controladores e retirar o maior partido de cada um deles para realizar uma dada tarefa. O controlo visual dinâmico é também apresentado, assim como as principais contribuições existentes na literatura. O modelo de interacção entre o robô e as características do objecto da imagem, i.e. o jacobiano da imagem, pode também ser estimado conforme apresentado na secção 3.5. A solução baseada no planeamento *off-line* da trajectória na imagem foi também apresentada.

Em seguida são apresentadas as principais contribuições da presente tese, todas verificadas experimentalmente:

1. **Extensão do controlador visual dinâmico de Kelly.** Na sub-secção 4.3.1 foi desenvolvida a extensão ao controlador visual dinâmico proposto por Kelly. Este novo controlador tem a particularidade de ser desenvolvido para um robô manipulador planar com dois graus de liberdade na configuração *eye-in-hand*.
2. **Desenvolvimento e implementação de um controlador visual dinâmico baseado em posição.** Na sub-secção 4.3.2 foi apresentado o controlador visual dinâmico baseado em posição, em que a pose do objecto relativamente à câmara foi utilizada como característica do objecto. Esta característica foi estimada experimentalmente utilizando o algoritmo de Dementhon & Davis.
3. **Análise de estabilidade do controlador visual dinâmico baseado em posição.** Na secção 4.4 foi demonstrada a estabilidade assintótica da lei de controlo visual dinâmico baseado em posição, sendo que a mesma análise se aplica também à extensão do controlador visual dinâmico de Kelly.
4. **Aplicação de filtros *fuzzy* ao controlo visual cinemático.** Na secção 5.2 foram definidos os filtros *fuzzy* e a motivação para a sua utilização em controlo visual. Estes filtros melhoram o desempenho do controlo visual, ao diminuir as velocidades de junta nos instantes iniciais e também ao atenuar o carácter oscilatório destas quando o tempo de amostragem de visão é elevado.



5. **Desenvolvimento de um novo modelo de interação robô-câmara.** Na secção 5.3 foi apresentada a metodologia para a identificação do modelo do sistema robô-câmara. Para tal foram identificadas as entradas e saídas do sistema para posterior modelação *fuzzy*, a partir dos dados de entrada/saída observados.
6. **Desenvolvimento e implementação de um novo controlador baseado no modelo inverso *fuzzy*.** Na secção 5.4 foi utilizado o modelo inverso *fuzzy*, identificado pela metodologia proposta nesta tese, para o controlo do sistema robô-câmara. Foi ainda utilizado no anel de controlo um compensador *fuzzy* para compensar eventuais discrepâncias entre o modelo identificado e o real.
7. **Desenvolvimento de um aparato experimental para controlo visual.** No anexo B foi apresentado o aparato experimental desenvolvido, tendo como base um robô manipulador planar de dois graus de liberdade. A este robô foi acoplada uma câmara CCD no elemento terminal. Para o controlo do robô foi desenvolvido *software* específico, baseado na plataforma *MatLab Simulink xPC*. Para a obtenção da informação visual foi também desenvolvido *software* específico, baseado na placa de aquisição Matrox Meteor II.

No capítulo 6, foram apresentados os resultados da implementação no aparato experimental dos controladores e filtros apresentados nos capítulos 4 e 5 e sub-secção 3.3.1. O controlo visual híbrido,  $2\frac{1}{2}$  D, foi implementado na plataforma de simulação desenvolvida durante a tese. Esta plataforma foi também utilizada na simulação de todos os tipos de controlo visual de robôs manipuladores apresentados na presente tese.

## 7.2 Discussão

A presente tese abordou vários algoritmos para controlo visual de robôs manipuladores. Tendo como base os algoritmos apresentados e desenvolvidos, é importante reflectir sobre

algumas questões que surgiram no decorrer dos trabalhos da tese:

- *Qual a importância da informação visual na lei de controlo visual?*

A dependência das leis de controlo na informação visual sobre o objecto a manipular foi verificada na presente tese, através do estado da arte e nas experiências realizadas. Para cada informação visual é construído o modelo de interacção analítico que relaciona o movimento da câmara com o movimento das características do objecto na imagem, i.e. o jacobiano da imagem. A informação visual do objecto a extrair da imagem é de importância fulcral em controlo visual, pois é responsável pelo jacobiano da imagem e logo pelo comportamento da lei de controlo. Para além da construção do jacobiano da imagem é ainda importante extrair as características do objecto em imagens sucessivas e em ambientes de trabalho complexos, i.e. não-estruturados. Como premissa desta tese, todos os trabalhos desenvolvidos foram direccionados para os problemas associados ao controlo. Este facto foi desde o início tido em conta, nomeadamente no desenvolvimento do aparato experimental e na escolha de um alvo planar constituído por LED's, de forma a facilitar o processamento da imagem capturada.

- *Controlo visual dinâmico ou controlo visual cinemático?*

Foi abordada na tese o controlo do robô directamente com binários vindos da lei de controlo visual, i.e. o controlo visual dinâmico, e o controlo visual cinemático em que é fornecido ao robô a velocidade de junta a seguir durante o controlo. Esta segunda abordagem necessita da utilização de um anel interior para o controlo da velocidade de junta. Através dos resultados obtidos na presente tese, verificou-se através das experiências realizadas que ao controlo visual dinâmico está associado um melhor comportamento na resposta, tendo-se ainda demonstrado a estabilidade assintótica da lei de controlo. No entanto, apenas se verificou na bibliografia, experiências com robôs manipuladores, no máximo com dois graus de liberdade. Este facto é justificado pelo facto de os robôs manipuladores industriais disponíveis na

mercado terem na sua maioria os controladores fechados, i.e. não permitem o acesso aos binários, sugerindo também a barreira à expansão do controlo visual dinâmico. Assim a maioria do controlo visual de robôs manipuladores é cinemático, tendo a vantagem de separar completamente a lei de controlo de visão da cinemática do robô, desde que a lei de controlo em velocidade esteja baseada na velocidade do elemento terminal.

- *O porquê da utilização de modelos de interacção estimados?*

Os modelos de interacção analíticos têm a particularidade de depender das características do objecto na imagem utilizadas e da calibração do sistema robô-câmara. Esta última dependência tem vindo a ser esbatida nos últimos anos, devido à identificação de valores limite para o erro associado à calibração da câmara. A estimação do modelo de interacção tem vindo a ser utilizada no controlo visual para resolver as dependências anteriores e tornar a lei de controlo independente da calibração do sistema. Esta estimação pode ser realizada através de modelos ou de técnicas de aprendizagem, como a utilizada na presente tese. Com a utilização de modelos de interacção estimados, a utilização de um conjunto de pontos como características do objecto na imagem é suficiente, por exemplo os cantos do objecto, sendo estes relativamente fáceis de extrair da imagem. Assim, a procura de novas características do objecto na imagem com o intuito de melhorar o desempenho da lei de controlo, como tem sido realizado com os modelos analíticos, não seria necessária.

## 7.3 Trabalho Futuro

Várias direcções podem ser definidas para trabalho futuro, relativamente ao trabalho desenvolvido na presente tese.

Uma das limitações do aparato experimental utilizado é o número de graus de liberdade do robô manipulador. Logo, vários trabalhos podem ser realizados na implementação das

várias contribuições presentes na tese, com a utilização de um robô com seis graus de liberdade. Eventualmente, o trabalho de maior interesse seria mesmo a implementação das leis de controlo visual dinâmico, uma vez que apenas foram encontrados na bibliografia trabalhos para robôs manipuladores com dois graus de liberdade, com a excepção de [61]. Todas as restantes contribuições também devem ser implementadas num robô com seis graus de liberdade.

A estimação do modelo de interacção utilizando modelação *fuzzy* tem a desvantagem de ser realizada *off-line* e para uma trajectória pré-definida. A evolução que se impõe nesta metodologia é a estimação *on-line* do referido modelo utilizando técnicas adaptativas *fuzzy*, nomeadamente *Adaptative Fuzzy Control* [10].

Durante os trabalhos da tese foi ainda verificado que as várias arquitecturas de controlo visual, têm características e especificidades próprias. Actualmente, tem crescido o interesse em construir uma lei de controlo híbrida capaz de aumentar a região de estabilidade global e a convergência para a solução desejada, através da comutação entre os controladores quando um deles tende a divergir, i.e. encontra um mínimo local ou o objecto sai da imagem. Pode ainda ser útil, escolher uma determinada característica da imagem para realizar uma tarefa e a meio do controlo ser necessário mudar de característica, por exemplo devido a oclusões ou instabilidade no sistema de controlo. Assim uma direcção futura de elevado interesse prático é a de aplicar a teoria do controlo de sistemas híbridos, para obter uma metodologia capaz de mudar de controlador e/ou de informação visual automaticamente durante o controlo, de forma a ser possível provar a estabilidade assimpótica global.

Um campo importante no controlo visual de robôs, e à partida não abordado nesta tese, é relativo à utilização de objectos complexos em imagens que podem também ser complexas. Um campo de aplicação importante é a robótica móvel em meios urbanos, em que as imagens estão em constante mutação. A inclusão de controlo visual neste tipo de ambiente de trabalho, é um campo de investigação importante e que importa considerar na investigação sobre controlo visual de robôs.

# Bibliografia

- [1] H. Araújo, R. Carceroni, and C. Brown, *A fully projective formulation to improve the accuracy of lowe's pose-estimation algorithm*, Computer Vision and Image Understanding **70** (1998), no. 2, 227–238.
- [2] X. Armangué and J. Salvi, *A comparative review of camera calibrating methods with accuracy evaluation*, Pattern Recognition **35** (2002), no. 7, 1617–1635.
- [3] ———, *Overall view regarding fundamental matrix estimation*, Image and Vision Computing **21** (2003), no. 2, 205–220.
- [4] K.J. Aström and T. Hägglund, *Pid controllers: Theory, design, and tuning*, Instrument Society of America, Research Triangle Park, USA, 1995.
- [5] R. Babuška, *Fuzzy modeling for control*, Kluwer Academic Publishers, Boston, 1998.
- [6] R. Babuška, J.M. Sousa, and H.B. Verbruggen, *Inverse fuzzy model based predictive control*, Advances in Fuzzy Control (D. Driankov and R. Palm, eds.), Physica-Verlag, Heidelberg, New York, 1998, pp. 129–154.
- [7] L. F. baptista, *Controlo adaptativo de posição e força de robôs manipuladores*, Tese de Doutoramento, Instituto Superior Técnico, Lisboa, Portugal, 2000.
- [8] C. Batur, A. Srinivasan, and Chien-Chung Chan, *Inverse fuzzy model controllers*, Proceedings of the American Control Conference (San Francisco, USA), 1993, pp. 772–776.

- [9] F. Beer and E. Johnston, *Vector mechanics for engineers - dynamics, 2nd s.i. metric edition*, McGraw-Hill, 1990.
- [10] D. Bellomo, D. Naso, and R. Babuška, *Parameter convergence in adaptative fuzzy control*, Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (Setúbal, Portugal), August 2004, pp. 135–142.
- [11] F. Bensalah, *Estimation du mouvement par vision active*, These de l'Université de Rennes I, 1996.
- [12] A. Bernardino and J. Santos-Victor, *Binocular visual tracking: Integration of perception and control*, IEEE Transactions on Robotics and Automation **15** (1999), no. 6, 1080–1094.
- [13] J. Bezdek, *Pattern recognition with fuzzy objective functions*, Plenum Press, New York, 1981.
- [14] J. Bezdek, J. Keller, R. Krisnapuram, and M. Pal (Editors), *Fuzzy models and algorithms for pattern recognition and image processing*, Kluwer Academic Publishers, 1999.
- [15] L. Boullart, A. Krijgsman, and R.A. Vingerhoeds (eds.), *Application of artificial intelligence in process control*, Pergamon Press, Oxford, 1992.
- [16] E. F. Camacho and C. Bordons, *Model predictive control in the process industry*, Springer Verlag, London, 1995.
- [17] J. Carusone and G. D'Eleuterio, *The "feature cmac": a neural-network based vision system for robotic control*, Proceedings of the IEEE International Conference on Robotics and Automation, 1998, pp. 2959–2964.
- [18] E. Cervera, *Visual servoing toolbox*, EURON Summer School on Visual Servoing (Castello , Spain), 2002.

- [19] E. Cervera and P. Martinet, *Visual servoing with indirect image control and a predictable camera trajectory*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Kyongju, Korea), 1999, pp. 381–386.
- [20] F. Chaumette, *La relation vision-commande: théorie et application á des tâches robotiques*, These de l'Université de Rennes I, 1990.
- [21] ———, *Potential problems of stability and convergence in image-based and position-based visual servoing*, The Confluence of Vision and Control, D. Kriegman, G. Hager, A.S. Morse (eds.), LNCIS Series (1998), 66–78.
- [22] ———, *Image moments: a general and useful set of features for visual servoing*, IEEE Transactions on Robotics and Automation **20** (2004), no. 4, 713–723.
- [23] F. Chaumette, P. Rives, and B. Espiau, *Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing*, Proceedings of the IEEE Int. Conf. on Robotics and Automation (Sacramento, USA), 1991, pp. 2248–2253.
- [24] C. Cheah, S. Kawamura, and S. Arimoto, *Visual servoing of robot manipulator without camera calibration*, Proceedings of the 5th International Workshop on Advanced Motion Control (Coimbra, Portugal), 1998, pp. 688–693.
- [25] G. Chesi and K. Hashimoto, *A self-calibrating technique for visual servoing*, Proceedings of the 41st IEEE Conference on Decision and Control (Las Vegas, USA), 2002, pp. 2878–2883.
- [26] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, *Keeping features in the camera's field of view: a visual servoing strategy*, Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems (South Bend, USA), 2002.

- [27] A. Comport, E. Marchand, and F. Chaumette, *A real-time tracker for markerless augmented reality*, Proceedings of the ACM/IEEE International Symposium on Mixed and Augmented Reality (Tokyo, Japan), 2003, pp. 36–45.
- [28] P. Corke and M. Good, *Dynamic effects in visual closed-loop systems*, IEEE Transactions on Robotics and Automation **12** (1996), no. 5, 671–683.
- [29] P. Corke and S. Hutchinson, *A new hybrid image-based visual servo control scheme*, Proceedings of the 39th IEEE Conference on Decision and Control (Sidney, Australia), 2000, pp. 2521–2526.
- [30] ———, *A new partitioned approach to image-based visual servo control*, IEEE Transactions on Robotics and Automation **17** (2001), no. 4, 507–515.
- [31] A. Crétual and F. Chaumette, *Visual servoing based on image motion*, International Journal of Robotics Research **20** (2001), no. 11, 857–877.
- [32] K. Deguchi, *Optimal motion control for image-based visual servoing by decoupling translation and rotation*, Proceedings of the IEEE International Conference on Intelligent Robots and Systems, 1998, pp. 705–711.
- [33] D. Dementhon and L.S. Davis, *Model-based object pose in 25 lines of code*, International Journal of Computer Vision **15** (1995), no. 1/2, 123–141.
- [34] J. Denavit and R.S. Hartenberg, *A kinematic notation for lower-pair mechanisms based on matrices*, ASME Journal Applied Mechanics **22** (1955), 215–221.
- [35] M. Dhome, M. Richetin, J. Lapresté, and G. Rives, *Determination of the attitude of 3-d objects from a single perspective view*, IEEE Transactions on Pattern Analysis and Machine Intelligence **47** (1989), 33–44.
- [36] D. Driankov, H. Hellendoorn, and M. Reinfrank, *An introduction to fuzzy control, 2nd edition*, Springer-Verlag, Berlin Heidelberg, 1996.



- [37] C. G. Economou, M. Morari, and B. O. Palsson, *Internal model control. extension to nonlinear systems*, Ind. Eng. Chem. Process Des. Dev. **25** (1986), 403–411.
- [38] B. Espiau, F. Chaumette, and P. Rives, *A new approach to visual servoing in robotics*, IEEE Transactions on Robotics and Automation **8** (1992), no. 3, 313–326.
- [39] O. Faugeras, *Three dimensional computer vision, a geometric viewpoint*, MIT Press, Cambridge, Massachusetts, USA, 1993.
- [40] O. Faugeras and F. Lustman, *Motion and structure from motion in a piecewise planar environment*, International Journal of Pattern Recognition and Artificial Intelligence **2** (1988), no. 3, 485–508.
- [41] J. T. Feddema and O. R. Michell, *Vision guided servoing with feature-based trajectory generation*, IEEE Transactions on Robotics and Automation **5** (1989), no. 5, 691–700.
- [42] M. Fujita, A. Maruyama, M. Watanabe, and H. Kawaai, *Inverse optimal  $h_\infty$  disturbance attenuation for planar manipulators with the eye-in-hand system*, Proceedings of the 39th Conference on Decision and Control (Sidney, Australia), 2000, pp. 3945–3950.
- [43] N. Gans, P. Corke, and S. Hutchinson, *Performance tests of partitioned approaches to visual servo control*, Proceedings of the IEEE International Conference on Robotics and Automation (Washington, U.S.A.), 2002, pp. 1616–1623.
- [44] N. Gans and S. Hutchinson, *A switching approach to visual servo control*, Proceedings of the 2002 IEEE International Symposium on Intelligent Control (Vancouver, Canada), 2002, pp. 770–776.
- [45] ———, *An asymptotically stable switched system visual controller for eye in hand robots*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (Las Vegas, USA), 2003, pp. 735–742.

- [46] C. E. Garcia and M. Morari, *Internal model control: a unifying review and some new results*, Ind. Eng. Chem. Process Des. Dev. **21** (1982), 308–323.
- [47] V. Gengenbach, H.-H. Nagel, M. Tonko, and K. Schafer, *Multi-cameras visual servoing*, Proceedings of the IEEE International Conference on Robotics and Automation (Minneapolis, U.S.A.), 1996, pp. 1320–1325.
- [48] P.J. Sequeira Gonçalves, L.F. Mendonça, J.M. Sousa, and J.R. Caldas Pinto, *Fuzzy model based control applied to image-based visual servoing*, Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics (Setúbal, Portugal), August 2004, pp. 143–150.
- [49] ———, *Fuzzy model based control applied to path planning visual servoing*, Lecture Notes in Computer Science **3287** (2004), 224–231.
- [50] ———, *Image recognition applied to robot control using fuzzy modeling*, Lecture Notes in Computer Science **3211** (2004), 253–260.
- [51] P.J. Sequeira Gonçalves and J.R. Caldas Pinto, *2  $\frac{1}{2}$  d visual servoing applied to a planar robot*, Proceedings of the 5th Portuguese Conference on Control (Aveiro, Portugal), September 2002, pp. 183–188.
- [52] ———, *Comparing visual servoing architectures for a planar robot*, Proceedings of the 10th Mediterranean Conference on Control and Automation (Lisbon, Portugal), 2002.
- [53] ———, *Camera configurations of a visual servoing setup, for a 2 dof planar robot*, Proceedings of the 7th International IFAC Symposium on Robot Control (Wroclaw, Poland), September 2003, pp. 181–187.
- [54] ———, *Dynamic visual servoing of robotic manipulators*, Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (Lisbon, Portugal), 2003, pp. 560–565.

- [55] ———, *An experimental testbed for visual servo control of robotic manipulators*, Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (Lisbon, Portugal), 2003, pp. 377–382.
- [56] ———, *Dynamic position-based visual servoing of robotic manipulators*, Proceedings of the 35th International Symposium on Robotics (Paris, France), 2004.
- [57] D. E. Gustafson and W. C. Kessel, *Fuzzy clustering with a fuzzy covariance matrix*, Proceedings of the IEEE Conference on Decision and Control (San Diego, USA), 1979, pp. 761–766.
- [58] R. Hartley, *In defense of the 8-point algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 6, 580–593.
- [59] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, Cambridge, United Kingdom, 2000.
- [60] K. Hashimoto, *A review on vision-based control of robot manipulators*, Advanced Robotics **17** (2003), no. 10, 969–991.
- [61] K. Hashimoto, T. Ebine, and H. Kimura, *Visual servoing with hand-eye manipulator - optimal control approach*, IEEE Transactions on Robotics and Automation **12** (1997), no. 5, 766–774.
- [62] K. Hashimoto, T. Kimoto, M. Kawabata, and H. Kimura, *Manipulator control with image-based visual servo*, Proceedings of the IEEE International Conference on Robotics and Automation (Sacramento, USA), 1991, pp. 2267–2272.
- [63] K. Hashimoto and T. Noritsugu, *Performance and sensitivity in visual servoing*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), 1998, pp. 2321–2326.

- [64] ———, *Potential switching control in visual servo*, Proceedings of the IEEE International Conference on Robotics and Automation (San Francisco, USA), 2000, pp. 2765–2770.
- [65] J. Hill and W. Park, *Real time control of a robot with a mobile camera*, 9th International Symposium on Industrial Robots, 1979, pp. 233–246.
- [66] R. Horaud, B. Conio, O. Leboulleux, and B. Lacolle, *An analytic solution for the perspective 4-point problem*, Computer Vision, Graphics and Image Processing **18** (1989), no. 2, 176–186.
- [67] R. Horaud, F. Dornaika, B. Lamiroy, and S. Christy, *Object pose: The link between weak perspective, paraperspective, and full perspective*, International Journal of Computer Vision **22** (1997), no. 2, 173–189.
- [68] B. Horn, *Robot vision*, MIT Press, 1986.
- [69] K. Hosoda and M. Asada, *Versatile visual servoing without knowledge of true jacobian*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1994, pp. 186–193.
- [70] K. Hosoda, K. Igarashi, and M. Asada, *Adaptive hybrid control for visual servoing and force servoing in an unknown environment*, IEEE Robotics and Automation Magazine **5** (1998), no. 4, 39–43.
- [71] K.J. Hunt, D. Sbarbaro, R. Zbikowki, and P.J. Gawthrop, *Neural networks for control systems – a survey*, IEEE Transactions on Neural Networks **28** (1992), 1083–1112.
- [72] S. Hutchinson, G. Hager, and P. Corke, *A tutorial on visual servo control*, IEEE Transactions on Robotics and Automation **12** (1996), no. 5, 651–670.

- [73] M. Jägersand, O. Fuentes, and R. Nelson, *Experimental evaluation of uncalibrated visual servoing for precision manipulation*, Proceedings of the IEEE International Conference on Robotics and Automation, 1994, pp. 186–193.
- [74] J.S.R. Jang, *Anfis: Adaptive-network-based fuzzy inference system*, IEEE Transactions on Systems, Man, and Cybernetics **3** (1993), no. 23, 665–685.
- [75] W. Jang and Z. Bien, *Feature-based visual servoing of an eye-in-hand robot with improved tracking performance*, Proceedings of the IEEE International Conference on Robotics and Automation (Sacramento, USA), 1991, pp. 2254–2260.
- [76] R. Kelly, *Robust asymptotically visual servoing of planar robots*, IEEE Transactions on Robotics and Automation **12** (1996), no. 5, 759–766.
- [77] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, *Stable visual servoing of camera-in-hand robotic systems*, IEEE/ASME Transactions on Mechatronics **5** (2000), no. 1, 39–48.
- [78] R. Kelly, J. Favela, J. Ibarra, and D. Bassi, *Asymptotically stable visual servoing of manipulators via neural networks*, Journal of Robotic Systems **17** (2000), no. 12, 659–669.
- [79] D. Khadraoui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette, *Visual servoing in robotics scheme using a camera/laser-stripe sensor*, IEEE Transactions on Robotics and Automation **12** (1996), no. 5, 743–750.
- [80] D. Kragić and H. Christensen, *Using a redundant coarsely calibrated vision system for 3d grasping*, Proceedings of the Computational Intelligence for Modelling, Control and Automation, 1999, pp. 91–97.
- [81] R. Kruse, J. Gebhardt, and F. Klawonn, *Foundations of fuzzy systems*, John Wiley and Sons, Chichester, 1994.

- [82] E. Lefeberg, R. Kelly, R. Ortega, and H. Nijmeijer, *Adaptative and filtered visual servoing of planar robots*, Proceedings of the Fourth IFAC Symposium on Nonlinear Control Systems Design (Enschede, The Netherlands), 1998, pp. 563–568.
- [83] H. Longuet-Hoggins, *A computer algorithm for reconstructing a scene from two projections*, Nature **293** (1981), no. 10, 133–135.
- [84] D. Lowe, *Three-dimensional object recognition from single two-dimensional images*, Artificial Intelligence **31** (1987), no. 3, 441–450.
- [85] C. Lu, G. Hager, and E. Mjolsness, *Fast and globally convergent pose estimation from video images*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 6, 610–622.
- [86] Q. Luong and O. Faugeras, *The fundamental matrix: theory, algorithms, and stability analysis*, International Journal of Computer Vision **17** (1996), 43–75.
- [87] E. Malis, *Contributions à la modélisation et à la commande en asservissement visuel*, These de l’Université de Rennes I, 1998.
- [88] ———, *Visual servoing invariant to changes in camera intrinsic parameters*, IEEE Transactions on Robotics and Automation **20** (2004), no. 1, 72–81.
- [89] E. Malis and F. Chaumette, *Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods*, IEEE Transactions on Robotics and Automation **18** (2002), no. 2, 176–186.
- [90] E. Malis, F. Chaumette, and S. Boudet, *Positioning a coarse-calibrated camera with respect to an unknown planar object by  $2\frac{1}{2}d$  visual servoing*, IFAC Symposium on Robot Control, SYROCO ’97, 1997, pp. 517–527.
- [91] ———,  *$2\frac{1}{2}d$  visual servoing*, IEEE Transactions on Robotics and Automation **15** (1999), no. 2, 238–250.

- [92] ———, *Multi-cameras visual servoing*, Proceedings of the IEEE International Conference on Robotics and Automation (San Francisco, U.S.A.), 2000, pp. 3183–3188.
- [93] E. Mamdani, *Application of fuzzy logic to approximate reasoning using linguistic systems*, IEEE Transactions on Computers **26** (1977), no. 12, 1182–1191.
- [94] N. Mansard and F. Chaumette, *Tasks sequencing for visual servoing*, Proceedings of the IEEE International Conference on Intelligent Robots and Systems (Sendai, Japan), 2004, pp. 992–997.
- [95] P. Martinet, N. Daucher, J. Gallice, and M. Dhome, *Robot control using monocular pose estimation*, Workshop on New Trends in Image-based Visual Servoing, IROS'97 (Grenoble, France), 1997, pp. 1–12.
- [96] L. F. Mendonça, J. M. Sousa, and J. M. G. Sá da Costa, *Optimization problems in multivariable fuzzy predictive control*, International Journal of Approximate Reasoning **36** (2004), no. 3, 199–221.
- [97] Y. Mezouar and F. Chaumette, *Path planning for robust image-based control*, IEEE Transactions on Robotics and Automation **18** (2002), no. 4, 534–549.
- [98] H. Michel and P. Rives, *Singularities in the determination of the situation from the perspective view of three points*, Rapport de Recherche n°1850, INRIA, 1993.
- [99] W. Miller, *Real-time application of neural networks for sensor-based control of robots with vision*, IEEE Transactions on Systems Man and Cybernetics **19** (1989), no. 4, 825–831.
- [100] D. Oberkampf, D. DeMenthon, and L. Davis, *Iterative pose estimation using coplanar feature points*, CVGIP: Image Understanding **63** (1996), no. 3.
- [101] J. Peipmeier, G. McMurray, and H. Lipkin, *A dynamic jacobian estimation method for uncalibrated visual servoing*, Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 1999, pp. 944–949.

- [102] ———, *A dynamic quasi-newton method for uncalibrated visual servoing*, Proceedings of the IEEE International Conference on Robotics and Automation (Detroit, USA), 1999, pp. 1595–1600.
- [103] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical recipes in c*, 2nd ed., Cambridge University Press, 1992.
- [104] J. Qian and J. Su, *Online estimation of image jacobian matrix by kalman-bucy filter for uncalibrated stereo vision feedback*, Proceedings of the International Conference on Robotics and Automation (Washington, USA), 2002, pp. 562–567.
- [105] J. Reis, M. Ramalho, J. Caldas Pinto, and J.M.G. Sá da Costa, *Dynamical characterization of flexible robot links using machine vision*, Proceedings of the 5th Ibero American Symposium on Pattern Recognition (Lisbon, Portugal), 2000, pp. 487–492.
- [106] A. Remazeilles, F. Chaumette, and P. Gros, *Robot motion control from a visual memory*, Proceedings of the IEEE International Conference on Robotics and Automation (New Orleans, U.S.A.), 2004, pp. 4695–4700.
- [107] Z. Roth, B. Mooring, and B. Ravani, *An overview of robot calibration*, IEEE Journal of Robotics and Automation **3** (1987), no. 5, 377–385.
- [108] J. Santos-Victor, *Visual servoing of cellular robots*, Proceedings of the European Control Conference (Porto, Portugal), 2001.
- [109] J. Santos-Victor, R. Carelli, and S. Van de Zwaan, *Nonlinear visual control of remote cellular robots*, Proceedings of the 10th Mediterranean Conference on Control and Automation (Lisbon, Portugal), 2002.
- [110] J. Santos-Victor and G. Sandini, *Visual behaviors for docking*, Computer Vision and Image Understanding **67** (1997), no. 3, 223–238.



- [111] C. Scheering and B. Kersting, *Uncalibrated hand-eye coordination with a redundant camera system*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), 1998, pp. 2953–2958.
- [112] D.C. Schuurman and D.W. Capson, *Robust direct visual servo using network-synchronized cameras*, IEEE Transactions on Robotics and Automation **20** (2004), no. 2, 319–334.
- [113] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*, Springer-Verlag, London, 2000.
- [114] M. Setnes, R. Babuška, and H. B. Verbruggen, *Transparent fuzzy modeling*, International Journal of Human-Computer Studies **49** (1998), no. 2, 159–179.
- [115] L. Shapiro and G. Stockman, *Computer vision*, Prentice-Hall, 2001.
- [116] Y. Shirai and H. Inoue, *Guiding a robot by visual feedback in assembly tasks*, Pattern Recognition **5** (1973), 99–108.
- [117] J. Slotine and W. Li, *Applied nonlinear control*, Prentice-Hall, 1991.
- [118] C. Smith and N. Papanikolopoulos, *Computation of shape through controlled active exploration*, Proceedings of the IEEE International Conference on Robotics and Automation (San Diego, U.S.A.), 1994, pp. 2516–2521.
- [119] M. Sonka, V. Hlavac, and R. Boyle, *Image processing, analysis and machine vision, 2nd edition*, PWS Publishing, 1999.
- [120] J. M. Sousa and M. Setnes, *Fuzzy predictive filters in model predictive control*, IEEE Transactions on Industrial Electronics **46** (1999), no. 6, 1225–1232.
- [121] J.M. Sousa, R. Babuška, P. Bruin, and H.B. Verbruggen, *Fuzzy compensation of steady-state errors in nonlinear control systems*, Proceedings of 4<sup>th</sup> European Congress on Fuzzy and Intelligent Technologies, EUFIT'96 (Aachen, Germany), September 1996, pp. 946–950.

- [122] J.M. Sousa, R. Babuška, and H.B. Verbruggen, *Branch-and-bound optimization in fuzzy predictive control: An application to an air conditioning system*, *Control Engineering Practice* **5** (1997), no. 10, 1395–1406.
- [123] J.M. Sousa and U. Kaymak, *Fuzzy decision making in modeling and control*, World Scientific Publishing Co., Singapore, 2002.
- [124] J.M. Sousa, C. Silva, and J. Sá da Costa, *Fuzzy active noise modeling and control*, *International Journal of Approximate Reasoning* **33** (2003), 51–70.
- [125] M. Spong and M. Vidyasagar, *Robot dynamics and control*, John Wiley and Sons, Singapore, 1989.
- [126] I.H. Suh and T.W. Kim, *Fuzzy membership function based neural networks with applications to the visual servoing of robot manipulators*, *IEEE Transactions on Fuzzy Systems* **2** (1994), no. 3, 203–220.
- [127] ———, *A visual servoing algorithm using fuzzy logics and fuzzy-neural networks*, *Mechatronics* **10** (2000), 1–18.
- [128] H. Sutanto, R. Sharma, and V. Varma, *Image based autodocking without calibration*, *Proceedings of the International Conference Robotics and Automation* (Albuquerque, USA), 1997, pp. 974–979.
- [129] O. Tahri and F. Chaumette, *Application of moment invariants to visual servoing*, *Proceedings of the IEEE International Conference on Robotics and Automation* (Taipeh, Taiwan), 2003, pp. 4276–4281.
- [130] T. Takagi and M. Sugeno, *Fuzzy identification of systems and its applications to modelling and control*, *IEEE Transactions on Systems, Man, and Cybernetics* **15** (1985), 116–132.
- [131] E. Trucco and A. Verri, *Introductory techniques for 3-d computer vision*, Prentice-Hall, New Jersey, 1998.

- [132] W.J. Wilson, C.C. Williams Hulls, and G.S. Bell, *Relative end-effector control using cartesian position based visual servoing*, IEEE Transactions on Robotics and Automation **12** (1996), no. 5, 684–696.
- [133] ———, *Robust image processing and position based visual servoing*, Proceedings of the IEEE International Conference on Robotics and Automation, Workshop "Robust Vision for Vision-Based Control of Motion" (Leuven, Belgium), 1998, pp. 1–24.
- [134] L. Zadeh, *Fuzzy sets*, Information and Control **8** (1965), no. 3, 338–353.
- [135] ———, *Outline of a new approach to the analysis of complex systems and decision processes*, IEEE Transactions on Systems, Man and Cybernetics **3** (1973), 28–44.
- [136] E. Zergeroglu, D. Dawson, M. Queiroz, and S. Nagarkatti, *Robust visual-servo control of robot manipulators in the presence of uncertainty*, Proceedings of the 38th Conference on Decision and Control (Phoenix, U.S.A.), 1999, pp. 4137–4142.
- [137] Z. Zhang and A. Hanson, *3d reconstruction based on homography mapping*, Proceedings of the ARPA Image Understanding Workshop (Palm Springs, USA), 1996.



# Apêndice A

## Modelação de Robôs Manipuladores

No presente anexo é realizada uma abordagem sucinta às ferramentas utilizadas na modelação de robôs manipuladores, com referência às equações do robô manipulador presente no aparato experimental, anexo B. Assim, na primeira secção deste anexo é abordado o movimento do corpo rígido, com especial ênfase à mudança de coordenadas e ao tissor de velocidade. Na segunda e terceira secções são abordadas a cinemática e a dinâmica de robôs manipuladores, respectivamente. Na quarta secção é apresentado o controlo de velocidade do robô manipulador, utilizada na tese.

### A.1 Movimento do Corpo Rígido

#### A.1.1 Mudança de Coordenadas

Considerando um ponto  $\mathbf{p}$  no espaço cartesiano e ainda dois sistemas de eixos coordenados  $\mathcal{F}_c$  e  $\mathcal{F}_o$ , como se pode verificar na figura A.1, define-se a matriz de transformação entre os referenciais  $\mathcal{F}_c$  e  $\mathcal{F}_o$ ,  ${}^c\mathbf{T}_o$ , através da respectiva matriz de rotação,  ${}^c\mathbf{R}_o$ , e o vector de translacção,  ${}^c\mathbf{t}_o$  (posição da origem do referencial  $\mathcal{F}_o$  relativamente à origem do referencial

$\mathcal{F}_c$ , em coordenadas do referencial  $\mathcal{F}_c$ ):

$${}^c\mathbf{T}_o = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (\text{A.1})$$

O ponto  $\mathbf{p}$  pode ser escrito no referencial  $\mathcal{F}_c$ , utilizando a expressão:

$${}^c\mathbf{p} = {}^c\mathbf{R}_o \cdot {}^o\mathbf{p} + {}^c\mathbf{t}_o \quad (\text{A.2})$$

Uma matriz de rotação genérica  $\mathbf{R}$ , pode-se escrever segundo a fórmula de Rodriguez, em função de um versor,  $\mathbf{u}$ , segundo o qual é realizada a rotação e um ângulo,  $\theta$ , que quantifica a respectiva rotação:

$$\mathbf{R} = \mathbf{I}_3 + \sin(\theta) \cdot \mathbf{S}(\mathbf{u}) + (1 - \cos(\theta)) \cdot \mathbf{S}^2(\mathbf{u}) \quad (\text{A.3})$$

em que a matriz anti-simétrica associada ao vector  $\mathbf{u}$ , é definida por:

$$\mathbf{S}(\mathbf{u}) = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix} \quad (\text{A.4})$$

Por outro lado, conhecido  $\mathbf{R}$ , o ângulo de rotação  $\theta$ , é dado por [87] :

$$\theta = \arccos\left(\frac{1}{2} \text{Trace}(\mathbf{R})\right) \quad (\text{A.5})$$

e o vector segundo é realizada a rotação  $\mathbf{u}$  multiplicado pelo ângulo de rotação, é dado por [87] :

$$\mathbf{u}\theta = \frac{1}{2 \cdot \sin(\theta)} \begin{pmatrix} r_{32} - r_{23} \\ r_{31} - r_{13} \\ r_{21} - r_{12} \end{pmatrix}, \text{ se } \begin{cases} 0 \leq \theta < \pi \\ \pi < \theta < 2\pi \end{cases} \quad (\text{A.6})$$

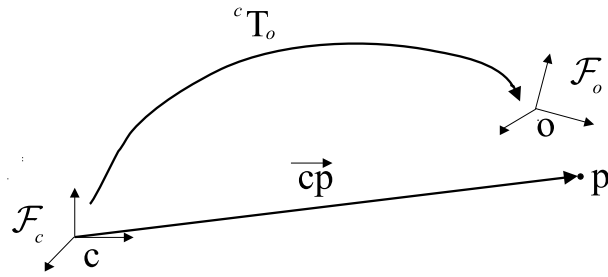


Figura A.1: Matriz de transformação entre dois referenciais

### A.1.2 Torsor de Velocidade

Considerando que na figura A.1, o referencial  $\mathcal{F}_c$  está associado a uma câmara e o referencial  $\mathcal{F}_o$  está associado a um objecto no espaço, define-se torsor de velocidade (*velocity screw*), expresso no referencial  $\mathcal{F}_c$ , entre a câmara e a cena em que o objecto se encontra enquadrado, como:

$${}^c\mathbf{v} = \begin{pmatrix} \mathbf{v}^T & \boldsymbol{\omega}^T \end{pmatrix}^T \quad (\text{A.7})$$

em que  $\boldsymbol{\omega}$  corresponde à velocidade de rotação definida por  $\mathbf{S}(\boldsymbol{\omega}) = \mathbf{R} \cdot \dot{\mathbf{R}}^T$  e  $\mathbf{v}$  corresponde à velocidade de translacção:  $\mathbf{v}(\mathbf{p}) = \mathbf{v}(\mathbf{c}) + \mathbf{S}(\boldsymbol{\omega}) \cdot \overrightarrow{\mathbf{c}\mathbf{p}}$ . Este torsor de velocidade pode decompor-se em duas componentes, que dizem respeito ao torsor de velocidade da câmara,  $\mathbf{v}_c$ , e ao torsor de velocidade do objecto,  $\mathbf{v}_o$ :

$${}^c\mathbf{v} = \mathbf{v}_c - \mathbf{v}_o \quad (\text{A.8})$$

por forma a expressar o torsor de velocidade  $\mathbf{v}$  num outro referencial, por exemplo o referencial  $\mathcal{F}_o$ , é necessário proceder à seguinte transformação:

$${}^o\mathbf{v} = \begin{bmatrix} {}^o\mathbf{R}_c & \mathbf{S}({}^o\mathbf{t}_c) \cdot {}^o\mathbf{R}_c \\ \mathbf{0}_3 & {}^o\mathbf{R}_c \end{bmatrix} \cdot {}^c\mathbf{v} = {}^o\mathbf{W}_c \cdot {}^c\mathbf{v} \quad (\text{A.9})$$

## A.2 Cinemática de Robôs Manipuladores

A cinemática de robôs manipuladores estuda a geometria do seu movimento, sem considerar os binários ou forças que provocam esse movimento. A cinemática pode ainda ser dividida em cinemática de posição (directa e inversa) e cinemática diferencial, em que a primeira relaciona as posições de junta do robô e do espaço cartesiano onde este se encontra, e a segunda relaciona as respectivas velocidades.

Um manipulador é constituído por uma estrutura mecânica que pode ser modelada por uma cadeia articulada aberta, com vários corpos rígidos, designados por elos, ligados entre si por juntas. Considera-se que cada junta é constituída por um grau de liberdade, de rotação ou translacção. Assim, é necessário definir um conjunto de parâmetros cinemáticos para um robô manipulador de  $n$  graus de liberdade. Nomeadamente as coordenadas de posição das juntas do robô,  $\mathbf{q}$ :

$$\mathbf{q} = ( \mathbf{q}_1 \quad \mathbf{q}_2 \quad \dots \quad \mathbf{q}_n )^T \quad (\text{A.10})$$

as velocidades das juntas do robô,  $\dot{\mathbf{q}}$ :

$$\dot{\mathbf{q}} = ( \dot{\mathbf{q}}_1 \quad \dot{\mathbf{q}}_2 \quad \dots \quad \dot{\mathbf{q}}_n )^T \quad (\text{A.11})$$

e as coordenadas do elemento terminal do robô manipulador,  $\mathbf{r}_e$ :

$$\mathbf{r}_e = ( \mathbf{X} \quad \mathbf{Y} \quad \mathbf{Z} \quad \phi \quad \theta \quad \psi )^T \quad (\text{A.12})$$



### A.2.1 Cinemática Directa de Robôs Manipuladores

Para um robô manipulador com uma determinada estrutura mecânica, cada grau de liberdade está associado aos parâmetros das juntas e dos elos. O método sistemático para descrever as ligações da cadeia articulada aberta foi apresentado por Denavit e Hartenberg (DH) [34]. Este método resulta numa matriz de transformação homogênea  ${}^{i-1}\mathbf{T}_i$  que representa um sistema de coordenadas adjunto a cada ligação  $i$  relativamente ao sistema de coordenadas da ligação anterior  $i - 1$ . Desta forma é obtida a rotação e translacção entre os referenciais  $i$  e  $i - 1$ . Tendo definido o referencial do mundo "0", referencial de inércia do sistema dinâmico, é possível obter através do método de DH a posição e orientação do elemento terminal  $\mathbf{r}_e$  a partir das coordenadas de junta do robô,  $\mathbf{q}$ , i.e.:

$${}^0\mathbf{T}_e = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \cdot \dots \cdot {}^{n-1}\mathbf{T}_n \quad (\text{A.13})$$

A partir das equações (A.1) e (A.12) é possível escrever:

$${}^0\mathbf{T}_e = \begin{pmatrix} \mathbf{n}_x & \mathbf{o}_x & \mathbf{a}_x & \mathbf{X} \\ \mathbf{n}_y & \mathbf{o}_y & \mathbf{a}_y & \mathbf{Y} \\ \mathbf{n}_z & \mathbf{o}_z & \mathbf{a}_z & \mathbf{Z} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{pmatrix} \quad (\text{A.14})$$

e obter os ângulos de orientação, ângulos de Euler, definidos em (A.12), utilizando as expressões definidas em [113]:

$$\begin{aligned} \phi &= \text{Atan2}(\mathbf{a}_y, \mathbf{a}_x) \\ \theta &= \text{Atan2}\left(\sqrt{\mathbf{a}_x^2 + \mathbf{a}_y^2}, \mathbf{a}_z\right) \\ \psi &= \text{Atan2}(\mathbf{o}_z, -\mathbf{n}_z) \end{aligned} \quad (\text{A.15})$$

A cinemática directa de robôs manipuladores é então formalmente definida pela equação:

$${}^0\mathbf{r}_e = \kappa(\mathbf{q}) \quad (\text{A.16})$$

Para o caso do robô manipulador de dois graus de liberdade presente no aparato experimental, anexo B, a cinemática directa do manipulador, em que se pretende obter as coordenadas do espaço cartesiano em função das coordenadas do espaço de junta, é descrita pelas seguintes equações:

$$\begin{aligned} X &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ Y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ \psi &= \theta_1 + \theta_2 \end{aligned} \quad (\text{A.17})$$

em que  $\mathbf{q} = (\theta_1 \ \theta_2)^T$ ,  $l_1 = 0.32[m]$ ,  $l_2 = 0.265[m]$ .

### A.2.2 Cinemática Inversa de Robôs Manipuladores

Quando se pretende obter as coordenadas de junta a partir das coordenadas cartesianas do elemento terminal, é necessário utilizar as equações da cinemática directa e resolver o problema inverso. A cinemática inversa de robôs manipuladores é então formalmente definida pela equação:

$$\mathbf{q} = \kappa^{-1}({}^0\mathbf{r}_e) \quad (\text{A.18})$$

No caso do robô do aparato experimental as equações (A.17), são directamente utilizadas para obter a cinemática inversa:

$$\begin{aligned} \theta_1 &= \arccos \left( \frac{X^2 + Y^2 - l_1^2 - l_2^2}{2 \cdot l_1 \cdot l_2} \right) \\ \theta_2 &= \arctan \left( \frac{Y}{X} \right) - \arctan \left( \frac{l_2 \cdot \sin(\theta_2)}{l_1 + l_2 \cdot \cos(\theta_2)} \right) \end{aligned} \quad (\text{A.19})$$

### A.2.3 Cinemática Diferencial de Robôs Manipuladores

A cinemática diferencial de robôs manipuladores relaciona a velocidade das coordenadas de junta  $\dot{\mathbf{q}}$  com a velocidade do elemento terminal,  $\dot{\mathbf{r}}_e$ :

$$\dot{\mathbf{r}}_e = \left( \dot{X} \quad \dot{Y} \quad \dot{Z} \quad \dot{\phi} \quad \dot{\theta} \quad \dot{\psi} \right)^T \quad (\text{A.20})$$

A cinemática diferencial é definida por uma matriz, designada por jacobiana do robô,  $\mathbf{J}_R$ :

$$\dot{\mathbf{r}}_e = \frac{\partial \kappa(\mathbf{q})}{\partial \mathbf{q}} \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial X}{\partial q_1} & \frac{\partial X}{\partial q_2} & \cdots & \frac{\partial X}{\partial q_n} \\ \frac{\partial Y}{\partial q_1} & \frac{\partial Y}{\partial q_2} & \cdots & \frac{\partial Y}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \cdots & \frac{\partial \psi}{\partial q_n} \end{bmatrix} = \mathbf{J}_R \cdot \dot{\mathbf{q}} \quad (\text{A.21})$$

No caso do robô do aparato experimental, o jacobiano do robô relativamente ao referencial do mundo, é dado por:

$${}^0\dot{\mathbf{r}}_e = \begin{pmatrix} -l_1 \sin(\theta_1) - l_2 \sin(\theta_1 + \theta_2) & -l_2 \sin(\theta_1 + \theta_2) \\ l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) \end{pmatrix} \cdot \dot{\mathbf{q}} = {}^0\mathbf{J}_R \cdot \dot{\mathbf{q}} \quad (\text{A.22})$$

O jacobiano do robô no referencial do elemento terminal é dado por:

$${}^0\dot{\mathbf{r}}_e = {}^e\mathbf{W}_0 \cdot {}^0\mathbf{J}_R \cdot \dot{\mathbf{q}} \quad (\text{A.23})$$

em que a matriz  ${}^e\mathbf{W}_0$ , definida em (A.9), é calculada através da cinemática directa do robô. Para o robô do aparato experimental, o jacobiano do robô relativamente ao referencial do elemento terminal, é dado por:

$${}^e\mathbf{J}_R = \begin{bmatrix} l_1 \cdot \sin(\theta_2) & 0 \\ l_1 \cdot \cos(\theta_1) + l_2 & l_2 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \quad (\text{A.24})$$

### A.3 Dinâmica de Robôs Manipuladores

O modelo da dinâmica de robôs manipuladores descreve a relação entre os actuadores de junta e o movimento da estrutura mecânica que define o robô. Este modelo tem um papel fundamental na simulação do movimento, na análise da estrutura mecânica do robô e no projecto de algoritmos de controlo. O modelo apresentado nesta secção refer-se a robôs rígidos, i.e. sem flexibilidade nas juntas e elos.

No caso geral de um robô manipulador de  $n$  graus de liberdade, é apresentado o respectivo modelo matemático (A.25), com a ausência de termos de atrito e outras perturbações, conforme descrito em [113]:

$$\mathbf{M}(\mathbf{q}) \cdot \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \cdot \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (\text{A.25})$$

onde  $\mathbf{q}$  é o vector ( $n \times 1$ ) que define a posição das juntas do robô manipulador,  $\boldsymbol{\tau}$  é o vector ( $n \times 1$ ) que define os binários das juntas,  $\mathbf{M}$  é a matriz ( $n \times n$ ) que define a matriz de massas,  $\mathbf{C}$  é o vector ( $n \times 1$ ) que define as forças centrífugas e de coriolis,  $\mathbf{g}$  é o vector ( $n \times 1$ ) que define as forças gravíticas. O modelo do robô manipulador, como apresentado em (A.25) tem a seguinte propriedade notável [113]:

$$\dot{\mathbf{q}}^T \left[ \frac{1}{2} \cdot \dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right] \cdot \dot{\mathbf{q}} = 0, \forall \mathbf{q}, \dot{\mathbf{q}} \in \mathbb{R}^n \quad (\text{A.26})$$

Para o robô do aparato experimental, as matrizes da equação (A.25) são dadas por:

$$\mathbf{M}(\mathbf{q}) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \quad (\text{A.27})$$

$$M_{11} = (l_1 - l_{cg1})^2 \cdot m_1 + (2 \cdot l_1 \cdot \cos(\theta_2) \cdot (l_2 - l_{cg2}) + l_1^2 + (l_2 - l_{cg2})^2) \cdot m_2 + I_1 + I_2 + J_1$$

$$M_{12} = (l_1 \cdot \cos(\theta_2) * (l_2 - l_{cg2}) + (l_2 - l_{cg2})^2) \cdot m_2 + I_2$$

$$M_{12} = M_{12}$$

$$M_{12} = (l_2 - l_{cg2})^2 \cdot m_2 + I_2 + J_2$$

Note-se que os termos gravíticos  $\mathbf{g}(\mathbf{q})$ , são nulos e:

$l_{cg1} = 0.163(m)$  é o centro de massa do elo 1;

$l_{cg2} = 0.112(m)$  é o centro de massa do elo 2;

$I_1 = 0.12(Kg \cdot m^2)$  é o momento de inércia do elo 1;

$I_2 = 0.015(Kg \cdot m^2)$  é o momento de inércia do elo 2;

$J_1 = 0.093(Kg \cdot m^2)$  é o momento de inércia da junta 1;

$J_2 = 0.024(Kg \cdot m^2)$  é o momento de inércia da junta 2;

$m_1 = 5(Kg)$  é a massa do elo e junta 1;

$m_2 = 1.2(Kg)$  é a massa do elo e junta 2.

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -2 \cdot m_2 \cdot l_1 \cdot (l_2 - l_{cg2}) \cdot \sin(\theta_2) \cdot q_1 \cdot q_2 - \\ -m_2 \cdot l_1 \cdot (l_2 - l_{cg2}) \cdot \sin(\theta_2) \cdot q_2^2 \\ m_2 \cdot l_1 \cdot (l_2 - l_{cg2}) \cdot \sin(\theta_2) \cdot q_1^2 \end{bmatrix} \quad (A.28)$$

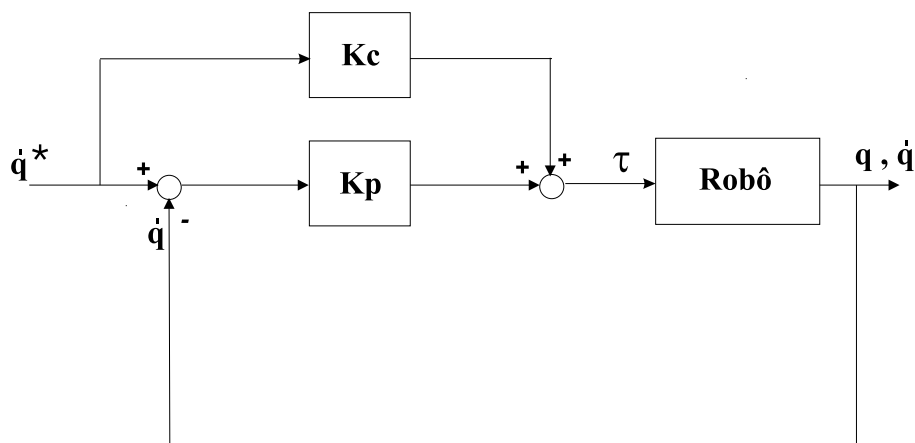


Figura A.2: Controlo em velocidade do robô planar de dois graus de liberdade

## A.4 Controlo em Velocidade de Robôs Manipuladores

Na presente secção é apresentado o diagrama de blocos do controlo em velocidade, utilizado nas experiências com controlo visual cinemático do robô existente no aparato experimental. Foi utilizado um controlo proporcional com uma acção *feed-forward* para compensar as forças de coriolis. Na parte proporcional foram utilizados os ganhos  $K_{p1} = 0.2$  e  $K_{p2} = 0.1$ , enquanto na parte *feed-forward* foi utilizado um ganho constante  $K_{c1} = 0.1$  e  $K_{c2} = 0.08$ , para cada uma das juntas.

# Apêndice B

## Aparato Experimental

O aparato experimental utilizado na presente tese pode ser dividido em dois sub-sistemas, o sistema de visão e o sistema robótico. O sistema de visão desenvolvido permite a aquisição e processamento de imagens à frequência de 50 [Hz]. Este sistema extrai as características do objecto na imagem, em coordenadas da imagem, e envia-as para o sistema do robô também a uma cadência de 50 [Hz]. O sistema robótico controla um robô manipulador planar com dois graus de liberdade com a informação visual vinda do sistema de visão. Na figura B.1 é possível visualizar uma fotografia do robô com a câmara acoplada ao elemento terminal, o alvo planar e o sistema para o controlo do robô (amplificadores de potência e o computador para o controlo, xPC *Target*).

### B.1 Sistema de Visão

#### B.1.1 Hardware

O sistema de visão realiza a aquisição e o processamento de imagens num computador pessoal com um processador Pentium IV da Intel a 1.7 GHz, utilizando para tal uma placa de aquisição de imagem Matrox Meteor II e uma câmara CCD Costar. Na figura B.2 é apresentado o computador pessoal encarregue do sistema de visão e que envia a informação

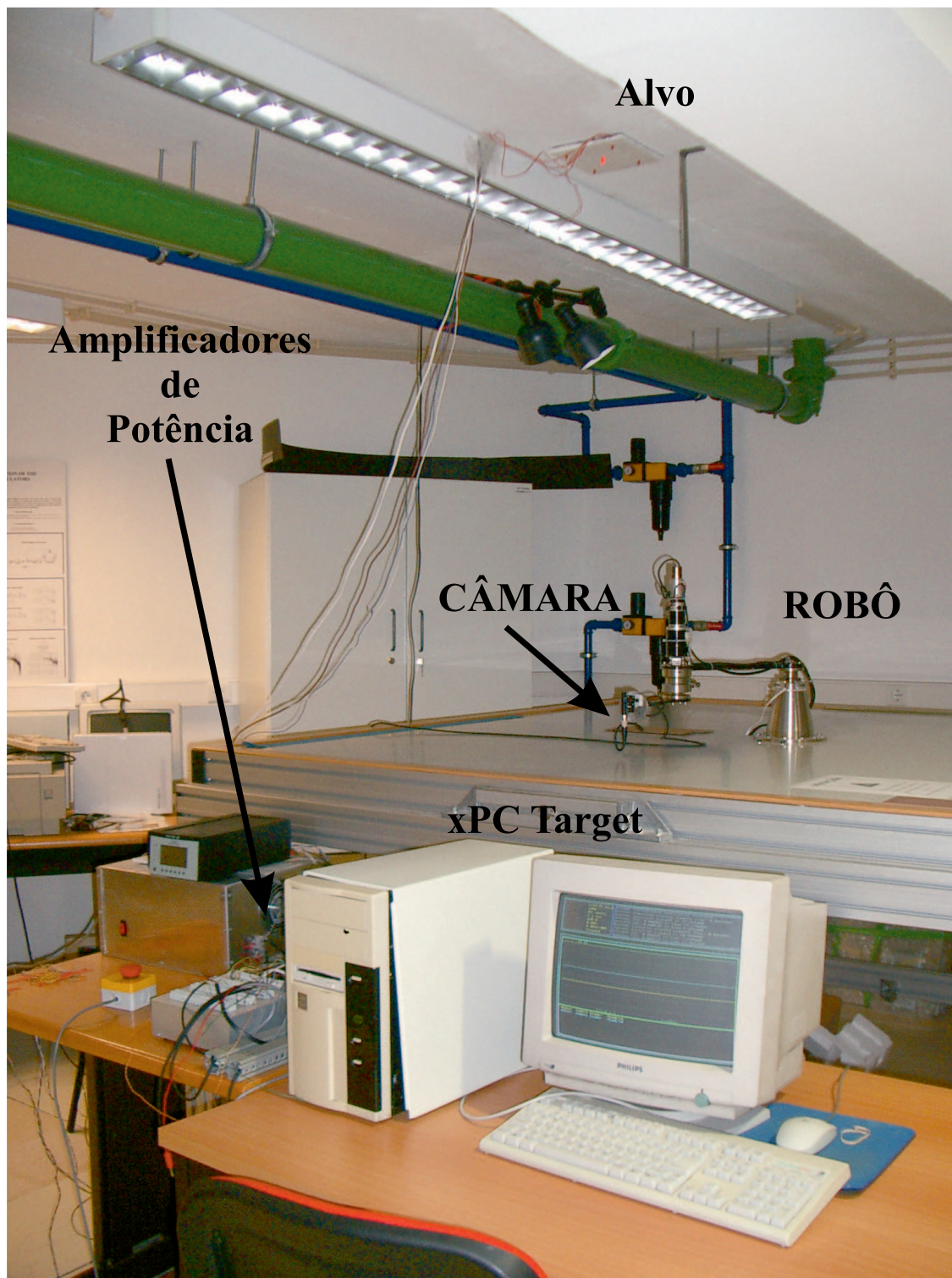


Figura B.1: Fotografia de conjunto do aparato experimental.





Figura B.2: Computador de Visão.

visual através da porta série RS-232. A câmara CCD é fixa ao elemento terminal do robô de forma a que o eixo óptico desta fique perpendicular ao plano do movimento do robô. O alvo planar é paralelo ao plano do movimento do robô e situa-se acima deste. Através dos constrangimentos descritos no aparato experimental, a distância  $Z$  entre o plano do robô e o do objecto é constante, podendo ser fixa durante o cálculo do jacobiano de imagem no controlo visual. O alvo planar consiste em vários LED's (*light emitting diode*), podendo estes serem ligados independentemente e assim variar o número de características do objecto na imagem, durante as experiências a realizar. Na maioria das experiências realizadas na presente tese é apenas necessário a utilização de um LED para controlar o robô, pois este apenas possui dois graus de liberdade. Assim, poupa-se no tempo de processamento e aquisição de imagem, o que permite aumentar o tempo de amostragem mínimo de visão para 20 [ms], i.e. para uma frequência máxima de 50 [Hz].

Após a extracção das características do objecto na imagem, através do *software* descrito na secção B.1.2, é efectuado o envio desses dados através da porta série RS-232. Este envio é realizado a uma velocidade de 115200 bits/segundo. Assim, é garantido que todo o processo de aquisição, processamento e envio de dados para o computador dedicado ao controlo do robô é realizado em menos de 20 [ms], i.e. quando uma nova imagem é adquirida, a informação visual da anterior imagem adquirida já foi enviada através da porta série.

### B.1.2 Software

O sistema de visão realiza as tarefas de aquisição, processamento e envio de informação visual através de um *software* desenvolvido em Microsoft Visual C++ para o sistema operativo Windows, utilizando a livreria de funções próprias de aquisição de imagem, fornecidas com a placa Matrox Meteor II.

Em controlo visual, o tempo de aquisição e processamento de imagens é um factor de extrema importância, uma vez que a informação visual obtida é utilizada directamente para fechar o anel de controlo. Extrair informações de uma imagem é um processo que à partida envolve erros, pois as imagens reais incluem ruído. Para atenuar o erro associado à imagem adquirida foram utilizados LED's, por forma a minimizar os erros na fase de binarização da imagem pelo método de Otsu [115]. Após o processo de binarização, é realizado o cálculo dos centróide(s) do(s) ponto(s) que definem o objecto na imagem. Em seguida são apresentadas as técnicas utilizadas para adquirir imagens a 50Hz e extrair informação visual de imagens sucessivas.

#### Aquisição de imagens a 50Hz

O sistema é capaz de adquirir 50 imagens por segundo, utilizando a técnica desenvolvida em [105]. O sistema video PAL é conhecido por ter uma cadência video de 25 imagens por

segundo. Contudo o sinal video é entrelaçado, i.e. as linhas pares e as linhas ímpares da imagem são codificadas em dois blocos separados. Estes blocos são amostrados imediatamente antes de cada um ser enviado para o sinal video. O mesmo é dizer que cada bloco é uma imagem com metade da resolução vertical, podendo portanto o processamento ser realizado a 50 imagens por segundo num dos blocos. Os dois blocos foram considerados como duas imagens separadas para assim obter uma frequência de visão de 50[Hz]. Ao calcular o centróide de um LED na imagem, um erro na sua coordenada vertical irá surgir, devido a que só metade das linhas estar a ser considerada no cálculo. Para obviar este facto é realizada uma multiplicação por dois na coordenada vertical do centróide, para assim obter uma estimação dessa mesma coordenada.

A técnica apresentada foi implementada utilizando as ferramentas específicas do *software* da placa de aquisição Matrox Meteor II, capazes de definir instruções de captura de imagem com prioridade máxima sobre o Windows, i.e. capazes de parar o processamento do computador quando as instruções de aquisição são chamadas.

Para que seja possível cumprir o tempo de amostragem de visão requerido de 50 Hz, as rotinas para realizar as correspondências entre imagens sucessivas devem ser simples e computacionalmente não muito pesadas. Em seguida é apresentada a heurística aplicada ao aparato experimental.

### **Correspondências entre imagens sucessivas**

Quando se trabalha numa sequência de imagens, é sempre necessário resolver o seguinte problema: Dado um conjunto de pontos numa imagem, qual a sua localização na imagem seguinte? Em [131] são apresentadas várias técnicas para resolver o problema das correspondências. No entanto os métodos exigem um tempo de cálculo elevado e portanto não são facilmente aplicáveis em tempo real. Quando se utiliza apenas um ponto na imagem, que é o caso da maioria das experiências realizadas na presente tese, a solução para a

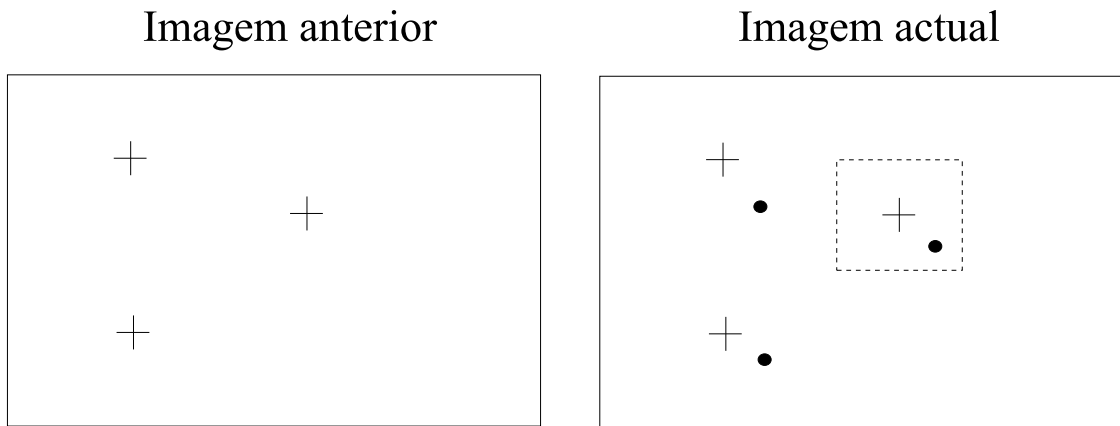


Figura B.3: Correspondências entre imagens sucessivas. '+' , centróides na imagem anterior. '●', centróides na imagem actual. Traço interrompido, janela de procura.

correspondência é trivial. Em seguida é apresentada a heurística utilizada na presente tese para resolver o problema de várias correspondências.

Na fase inicial, após a captura da primeira imagem e antes do início do controlo, são identificados e numerados os vários LED's na imagem através de um processo de etiquetagem [115]. Posteriormente é calculado o centróide de cada LED. Ainda antes do início do controlo visual, i.e. da aquisição de imagens, é definido o tamanho da janela de procura do LED na imagem seguinte. Na presente tese foi utilizada uma janela de  $(20 \times 20)$  pixels, ver figura B.3. Para cada nova imagem capturada é necessário definir a janela de procura, que tem como centro o centróide do LED calculado na imagem anterior. Em cada janela definida na nova imagem capturada é realizado o cálculo do centróide do LED. Assim, são necessárias tantas janelas quantos os LED's utilizados no alvo.

Para o caso em que são utilizados quatro LED's, foi verificado experimentalmente que a frequência de amostragem de visão passou a ser de 10Hz, em contraste com os 50Hz verificados com a utilização de um LED.

## B.2 Sistema Robótico

O sistema robótico consiste num robô manipulador planar com dois graus de liberdade, que se movimenta num plano horizontal, os respectivos amplificadores de potência, um computador pessoal (*xPC Target*) com um processador Intel Pentium a 200MHz e as respectivas placas de interface ISA *ServoToGo*. Ambas as juntas do robô são actuadas por *Harmonic Drive Actuators - HDSH-14*. Cada actuador é portanto constituído por um motor eléctrico D.C., uma harmonic drive, um encoder e um taquímetro. Os amplificadores de potência foram configurados para funcionarem no modo de corrente. Neste modo de funcionamento o sinal de controlo para os motores é uma tensão na gama  $\pm 10[\text{V}]$ , com intensidades de corrente que variam entre  $(-2 : 2)[\text{V}]$ . Na figura B.4 é apresentado o robô com a câmara acoplada ao elemento terminal. Os amplificadores de potência, a fonte de alimentação e a placa de ligações entre os motores e o computador *xPC Target*, são apresentados na figura B.5.

## B.3 Interligação dos Sistemas

A figura B.6 apresenta o esquema utilizado para interligar os dois sistemas referidos anteriormente, de visão e robótico. Os algoritmos de controlo do robô manipulador foram desenvolvidos num computador, designado por *Host-PC*, utilizando ferramentas disponibilizadas no programa *MatLab*, com a utilização das seguintes *toolboxes*: *Simulink*, *xPC*, *DSP Blockset*, *Fuzzy Logic Toolbox*. O algoritmo de controlo, ver figura B.7 é compilado e em seguida criado um ficheiro executável, a ser transferido para o computador *xPC Target*. Esta transferência é realizada através do protocolo TCP/IP, via ligação *Ethernet*. Os algoritmos de controlo são então executados no computador *xPC Target*, durante um determinado tempo e posteriormente os resultados das leituras dos sensores e de todas as variáveis internas que se pretendam monitorizar são transferidas para o *Host-PC*, também via ligação *Ethernet*. Note-se ainda que durante a execução do algoritmo de controlo, o

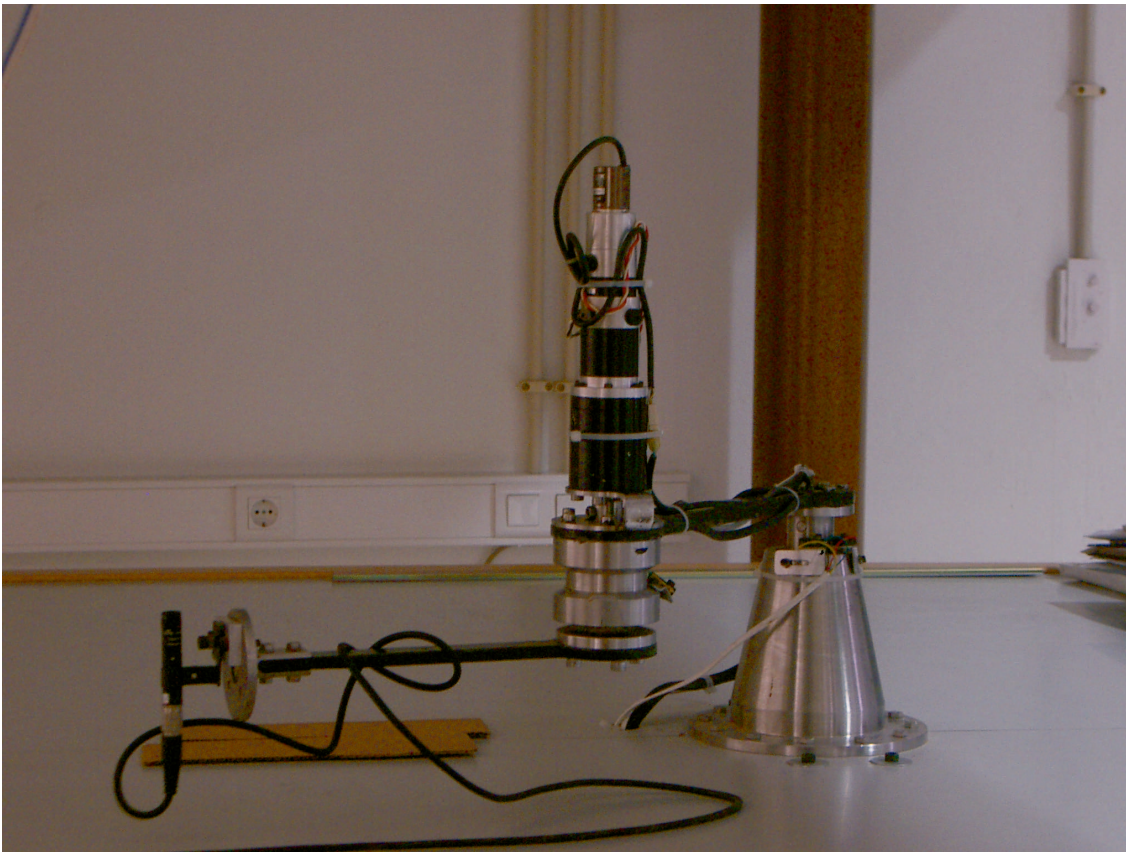


Figura B.4: Robô planar com a câmara acoplada ao elemento terminal.

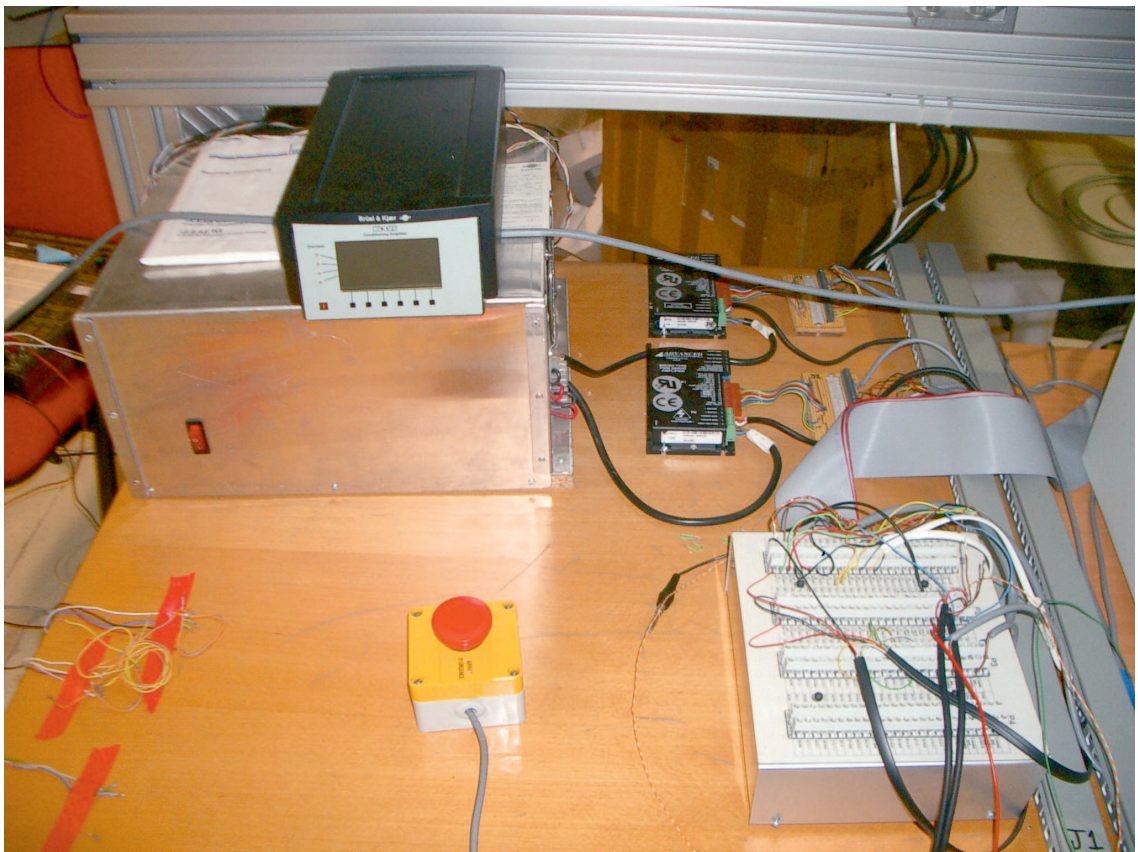


Figura B.5: Vista dos amplificadores de potência, fonte de alimentação e placa de ligações entre o computador, amplificadores de potência e actuadores.

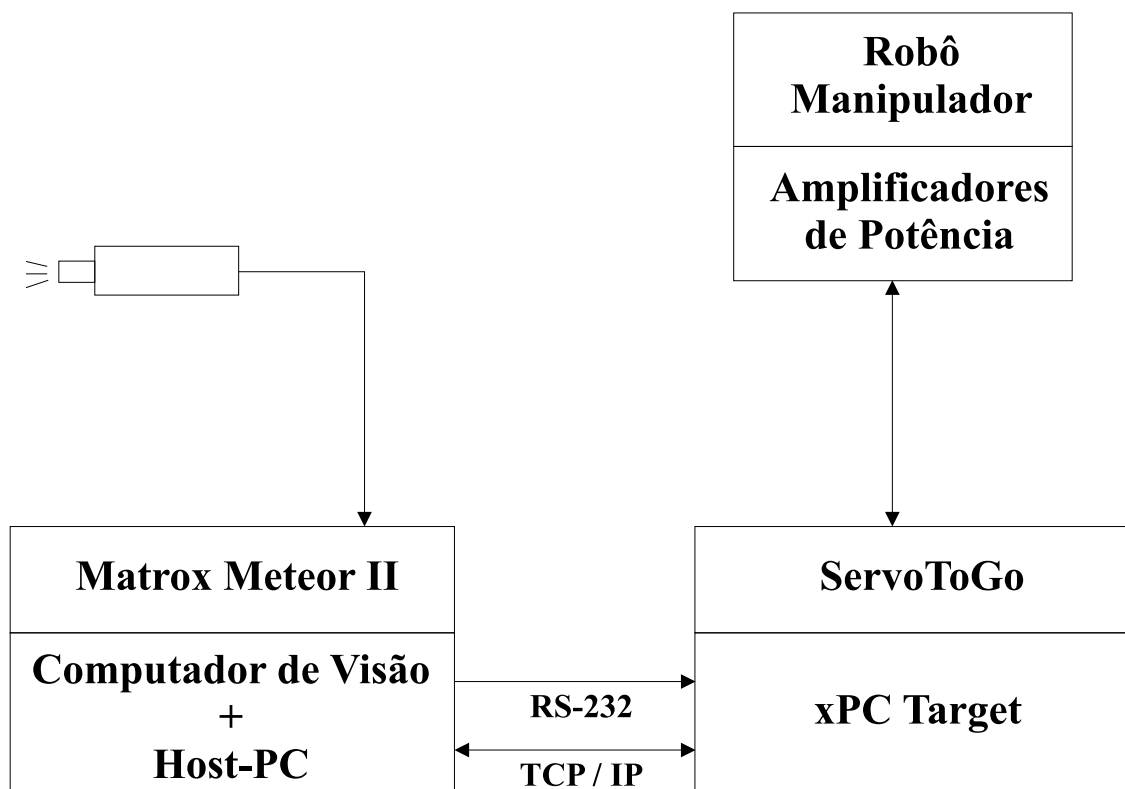


Figura B.6: Interligação dos Sistemas de Visão e Robótico.

computador de visão, envia para o computador *xPC Target* a informação visual necessária para fechar o anel de controlo, através da porta série RS-232.

A ligação entre o computador *xPC Target* e os amplificadores de potência e encoders é realizada através da placa ISA *ServoToGo*. Os *drivers* que permitem controlar esta interligação foram desenvolvidos de raiz para a plataforma *MatLab Simulink xPC* no Departamento de Engenharia Mecânica do Instituto Superior Técnico.

Na figura B.7 é apresentada a implementação da lei de controlo visual cinemático baseada na imagem, em que se verifica a versatilidade da programação, através de blocos funcionais. Nas figuras B.8 e B.9 são apresentados o conteúdos dos blocos referentes ao cálculo da matriz jacobiana global e à implementação da lei de controlo em velocidade do robô manipulador planar, ver apêndice A.4. Refira-se ainda que na lei de controlo em velocidade, a velocidade das juntas do robô é estimada através dos valores lidos nos



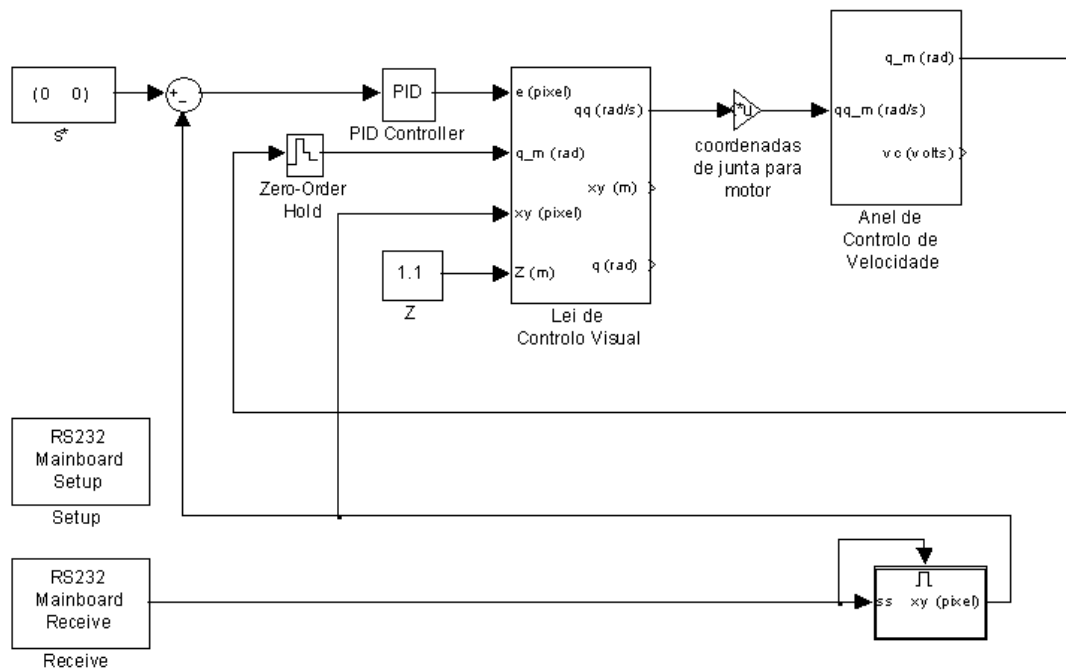


Figura B.7: Implementação da lei de controlo visual em *MatLab Simulink xPC*.

respectivos encoders. Tal facto é devido a que a leitura realizada pelo taquímetro tem muito ruído, tornando assim impossível a sua utilização para fechar o anel de controlo. Assim, as velocidades de junta são estimadas utilizando os anteriores cinco valores lidos pelo encoder [7].

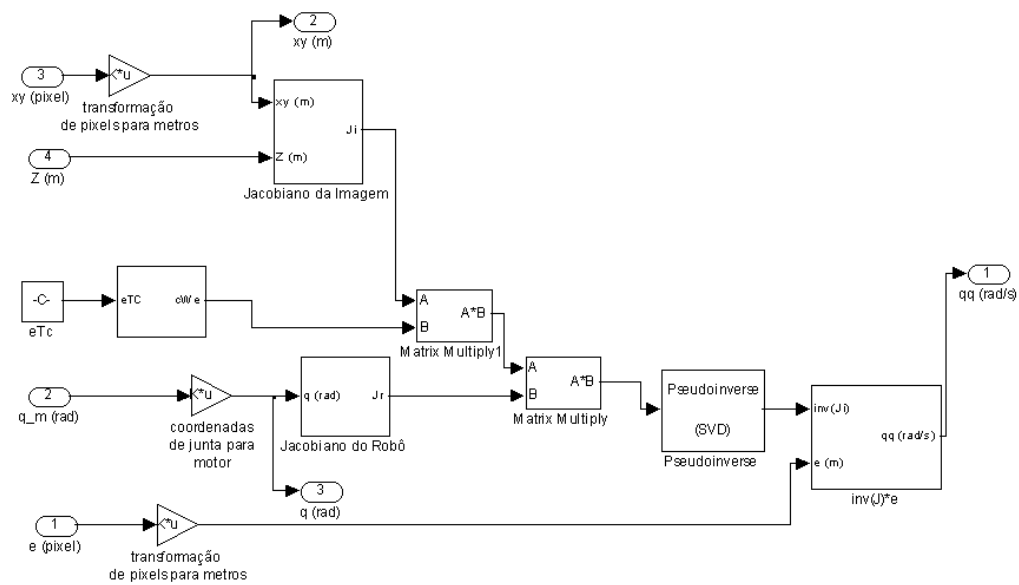


Figura B.8: Implementação do cálculo do jacobiano global para a lei de controlo visual em *MatLab Simulink* xPC.

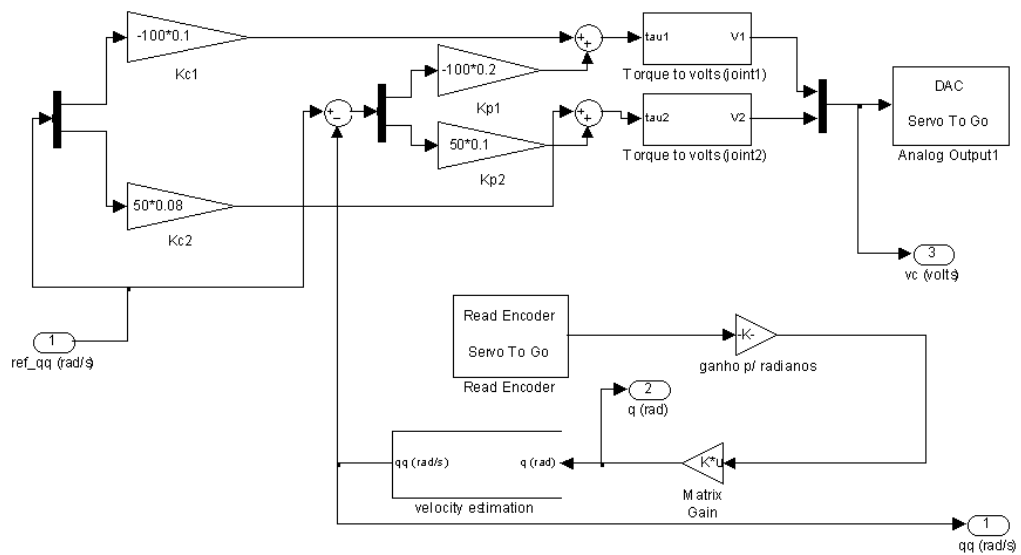


Figura B.9: Implementação da lei de controlo em velocidade em *MatLab Simulink xPC*