

**Rafael Perrone Bezerra de Menezes**

**CONTROLE SERVO VISUAL DE VEÍCULOS AÉREOS  
MULTIRROTORES**

Salvador

2013

**Rafael Perrone Bezerra de Menezes**

**CONTROLE SERVO VISUAL DE VEÍCULOS AÉREOS  
MULTIRROTORES**

Dissertação apresentada ao Programa de Pós-Graduação em Mecatrônica da Universidade Federal da Bahia como parte dos requisitos para obtenção do título de Mestre em Mecatrônica.

Orientador: Prof. Dr. Leizer Schnitman

Salvador

2013

---

P453 Perrone, Rafael Bezerra de Menezes  
Controle servo visual de veículos aéreos multirrotores /  
Rafael Perrone Bezerra de Menezes. – Salvador, 2013.  
142 f. : il. color.

Orientador: Prof. Dr. Leizer Schnitman

Dissertação (mestrado) – Universidade Federal da Bahia.  
Escola Politécnica, 2013.

1. Robôs - Sistemas de controle. 2. Robótica. 3. Inteligência  
computacional. I. Schnitman, Leizer. II. Universidade Federal da  
Bahia. III. Título.

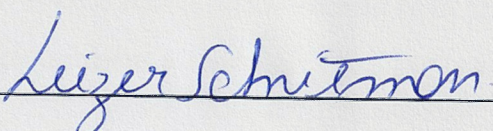
CDD: 629.89

---

# CONTROLE SERVO VISUAL DE VEÍCULOS AÉREOS MULTIRROTORES

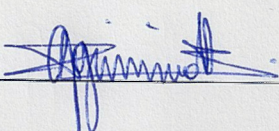
Rafael Perrone Bezerra de Menezes

Composição da Banca Examinadora:



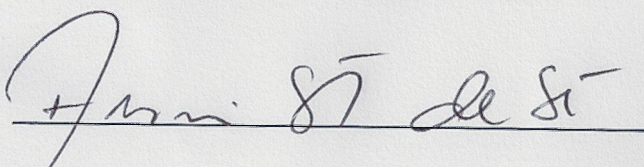
Prof. Dr. Leizer Schnitman, UFBA

- Orientador



Prof. Dr. Geraldo Figueiredo da Silveira Filho, CTI Renato Archer

- Membro Externo



Prof. Dr. Alirio Santos de Sá, UFBA

- Membro Interno

A minha mãe, por me ensinar  
a ter sede de conhecimento.

A meu pai, por me dar uma  
referência a seguir.

# Agradecimentos

Agradeço ao meu orientador, Prof. Leizer Schnitman, pela confiança e por clarear o caminho para a realização deste trabalho.

Ao Prof. Humberto Xavier de Araújo, por todo o tempo dedicado e o esclarecimento das tantas dúvidas que tive.

Agradeço o suporte e paciência de minha esposa, Mariana, e o enorme incentivo de meu pai, Francisco, minha mãe, Indiara, que também contribuiu com várias revisões do texto, e minha irmã, Juliana, que mesmo sem saber me motivou a escolher esse caminho. Este trabalho não teria começado nem terminado sem a ajuda de vocês. Muito obrigado. Agraço também a ajuda do Prof. Luciano Rebouças, Prof. Herman Lepikson e Lúcia Lago.

*“O sábio que se mostra orgulhoso e pedante  
revela que não sabe honrar a Ciência.”*  
— DR. ALFREDO GUIMARÃES CHAVES



# Resumo

O uso de câmeras aumenta a flexibilidade dos robôs. A visão permite que sistemas robóticos móveis operem em locais de onde se tem pouco ou nenhum conhecimento prévio. Sua aplicação também pode aumentar a confiabilidade e precisão das tarefas robóticas. A realimentação de informações visuais para controlar ou estabilizar a postura de robôs, chamado de controle servo visual ou servovisão, é uma das técnicas de controle baseada em visão. Classifica-se a servovisão em dois grupos: baseada em imagem, que define o erro no espaço da imagem; baseada em posição, que usa informações visuais para fazer uma reconstrução 3D, com o erro definido no espaço Euclidiano. Este trabalho apresenta uma avaliação do desempenho do controle servo visual baseado em imagem aplicado a veículos aéreos multirrotores. A abordagem baseada em imagem elimina a necessidade de reconstrução tridimensional com alto custo computacional presente na técnica baseada em posição. Outra vantagem é a reduzida sensibilidade a erros de calibração da câmera e ruídos na imagem quando comparada à baseada em posição. A servovisão clássica usa apenas um controlador proporcional constante. Tal abordagem clássica confere simplicidade ao sistema, porém tende a apresentar um elevado sinal de controle nos instantes iniciais do movimento, quando o erro é relativamente grande. O objetivo fundamental do trabalho aqui descrito é analisar abordagens que melhorem o desempenho da servovisão clássica mantendo sua complexidade relativamente baixa. Avaliaram-se duas alternativas: ganho variável e filtro *fuzzy*. Para comparar os três controladores (clássico, ganho variável e filtro *fuzzy*) realizaram-se experimentos simulados em dois cenários: primeiro, usando um sistema cinemático completamente atuado com seis graus de liberdade, para verificar a influência do controlador isoladamente; depois, usando um modelo dinâmico de um quadricóptero para verificar o comportamento em situações próximas da aplicação real. Tanto o ganho variável quanto o filtro *fuzzy* demonstraram ser capazes de solucionar, ou amenizar, parte dos problemas do controlador clássico. Por fim, validaram-se os algoritmos em um veículo aéreo não tripulado de quatro rotores — o quadricóptero. Verificou-se que as duas abordagens avaliadas apresentaram menor esforço de controle durante os instantes iniciais, enquanto o tempo de acomodação foi, também, menor. O desempenho em regime apresentou menor oscilação da orientação do robô com o ganho variável e ainda menor com o filtro *fuzzy* quando comparados à abordagem clássica, indicando que os dois controladores alternativos conseguem reduzir os efeitos das propriedades dinâmicas do veículo.

**Palavras-chave:** Controle. Robótica. Visão computacional.



# Abstract

*The use of cameras increase robots flexibility. Vision allows mobile robotic systems to operate in places where there is few a priori knowledge. Moreover, these sensors can increase the reliability and precision of robotic tasks. The feedback of visual information to control or stabilize the pose of a robot, known as visual servo control or visual servoing, is one of the vision-based control techniques. Visual servoing can be classified into two categories: image-based, that defines the error in image space; and position-based, which uses visual information for 3D reconstruction and defines the error in Euclidian space. This work presents a performance evaluation of image-based visual servoing applied to multirotor aerial vehicles. Image-based approach does not require costly 3D reconstructions and is known to be more robust to calibration errors and image noise when compared to position-based. The controller implicitly regulates the robot pose using the projection of a target object in the image plane as reference. Classical visual servoing approaches use only a constant proportional controller. Such classical approach is easy to implement, but it usually causes high control signals on initial instants, when error is relatively large. The fundamental objective of the work described here is to analyze approaches that can improve classical visual servoing performance keeping its low complexity. Two alternatives were evaluated: adaptive gain and fuzzy filter. To compare the three controllers (classical, adaptive gain and fuzzy filter) each one was simulated in two different scenarios: first, using a kinematic system completely actuated with six degrees of freedom, to evaluate the controller influence isolatedly; then, using a dynamical model of a quadrotor, to evaluate the behavior of the system in real situations. Both the adaptive gain and fuzzy filter approaches were able to solve or reduce part of the problems of the classical controller. Finally, all three algorithms were validated on a real quadrotor. The two aforementioned approaches presented lower control effort during initial iterations while settling time was also shorter. Steady state performance showed less oscillations in the vehicle orientation with adaptive filter and even lower with fuzzy filter when compared to the classical approach. These results indicate that both alternative controllers can reduce the effects of the dynamical properties of the vehicle.*

**Keywords:** Control. Robotics. Computer vision.

# Sumário

1	INTRODUÇÃO . . . . .	22
1.1	Um novo contexto para a robótica . . . . .	22
1.2	Histórico do controle de robôs baseado em visão . . . . .	25
1.3	Justificativa . . . . .	32
1.4	Contribuição . . . . .	34
1.5	Estrutura da dissertação . . . . .	35
2	CONTROLE SERVO VISUAL . . . . .	36
2.1	Projeção perspectiva . . . . .	36
2.2	Aquisição da imagem . . . . .	38
2.3	Processamento da imagem . . . . .	41
2.4	Definição e cálculo do erro . . . . .	43
2.4.1	<i>Controle servo visual baseado em posição</i> . . . . .	43
2.4.2	<i>Controle servo visual baseado em imagem</i> . . . . .	45
2.4.3	Arquiteturas híbridas . . . . .	48
2.5	Cálculo do sinal de controle . . . . .	50
2.6	Considerações . . . . .	51
3	FUNDAMENTAÇÃO TEÓRICA DO IBVS . . . . .	52
3.1	A matriz de interação . . . . .	52

---

3.1.1	Dedução da matriz de interação para pontos . . . . .	53
3.1.2	Cálculo da lei de controle . . . . .	57
3.1.3	Interpretação geométrica da matriz de interação . . . . .	58
3.1.4	Estabilidade . . . . .	60
3.1.5	Alternativas para obtenção da matriz de interação . . . . .	63
3.1.6	Aplicação do IBVS . . . . .	64
3.1.7	Problemas de estabilidade e convergência . . . . .	68
3.1.8	O ganho proporcional constante . . . . .	75
3.2	<b>Conclusão</b> . . . . .	76
4	<b>MODELAGEM E ALTERNATIVAS DE CONTROLE</b> . . . . .	78
4.1	<b>Nomenclatura de veículos aéreos com quatro rotores</b> . . . . .	78
4.2	<b>Modelagem matemática de quadrrrotores</b> . . . . .	79
4.3	<b>Particularidades da aplicação do IBVS em veículos quadrrrotores</b> . . . . .	83
4.3.1	Deslocamento das características visuais em decorrência da rolagem e arfagem . . . . .	84
4.4	<b>Melhorando o desempenho do IBVS clássico</b> . . . . .	86
4.4.1	Ganho variável . . . . .	87
4.4.2	Filtro <i>fuzzy</i> . . . . .	87
4.4.3	Considerações . . . . .	90
5	<b>RESULTADOS EXPERIMENTAIS</b> . . . . .	91
5.1	<b>Considerações sobre a aplicação</b> . . . . .	91
5.1.1	Movimentos do quadrrrotor em relação ao eixo ótico . . . . .	93
5.2	<b>Plataforma experimental</b> . . . . .	93
5.2.1	O quadrrrotor . . . . .	93
5.2.2	O computador remoto . . . . .	98

SUMÁRIO	xi
<b>5.3 Dados simulados</b>	99
5.3.1 Descrição da tarefa	99
5.3.2 Simulações com sistema cinemático completamente atuado com seis graus de liberdade	103
5.3.3 Simulações com o modelo dinâmico do quadricóptero	108
<b>5.4 Dados reais</b>	114
5.4.1 Aplicação no AR.Drone	114
<b>5.5 Considerações</b>	123
<b>6 CONCLUSÃO</b>	124
<b>6.1 Perspectivas futuras</b>	127
REFERÊNCIAS BIBLIOGRÁFICAS	129
APÊNDICE A – REPRESENTAÇÃO DE CORPOS NO ESPAÇO E SEU MOVIMENTO	135
<b>A.1 Representando posição</b>	136
<b>A.2 Representando orientação</b>	136
A.2.1 Matriz de rotação	137
A.2.2 Representação com três ângulos	137
A.2.3 Postura - combinando translação e rotação	138
<b>A.3 Velocidade de um ponto fixo em um sistema de coordenadas em movimento</b>	139
APÊNDICE B – CALIBRAÇÃO DE CÂMERAS	141
<b>B.1 Modelagem da câmera</b>	141

# Lista de ilustrações

FIGURA 1.1 – Alguns VANTs usados atualmente. (a) MQ-1 Predator, General Atomics; (b) Aeryon Scout, Aeryon Labs Inc.; (c) X8, DraganFly Innovations Inc.; (d) Tiriba, AGX Tecnologia Ltda. . . . .	25
FIGURA 1.2 – Alguns quadrrrotores usados em pesquisa. (a) ANU, Austrália; (b) ETH, Suíça; (c) CEA, França; (d) Stanford, EUA. . . . .	32
FIGURA 2.1 – Modelo de projeção perspectiva. . . . .	37
FIGURA 2.2 – Estrutura de um controlador servo visual baseado em posição. . .	44
FIGURA 2.3 – Estrutura de um controlador servo visual baseado na imagem. . .	45
FIGURA 3.1 – Vista tridimensional da postura inicial (vermelho) e desejada (azul) da câmera. (a) vista a 45°; (b) vista lateral paralela ao plano YZ. . . . .	65
FIGURA 3.2 – Resultados da simulação de IBVS para $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{z}}_d)$ . (e) Projeção inicial (vermelho) e desejada (azul) do objeto-alvo no plano da imagem com a trajetória de cada ponto (linha tracejada preta); (b) velocidades da câmera no espaço cartesiano; (c) posição e (d) orientação da câmera ao longo da tarefa. . . . .	67
FIGURA 3.3 – Movimentos esperados em tarefas de rotação pura para diferentes estimativas da matriz de interação. Pontos azuis representam a posição inicial e pontos vermelhos a posição desejada. Adaptada de Chaumette e Hutchinson (2006). . . . .	69

FIGURA 3.4 – Efeito de recuo da câmera devido a rotação pura de $60^\circ$ em relação ao eixo $z$ com $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A câmera se afasta do objeto-alvo na direção do eixo ótico. . . . .	70
FIGURA 3.5 – Resultado de uma tarefa de IBVS de rotação pura de $\pi$ radianos em relação ao eixo $z$ para $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A tarefa falha. . . . .	72
FIGURA 3.6 – Efeito de recuo da câmera devido a rotação pura de $60^\circ$ em relação ao eixo $z$ com $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d,o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A câmera se aproxima do objeto-alvo na direção do eixo ótico. . .	73
FIGURA 4.1 – Notação usada para o quadrirrotor neste texto. . . . .	79
FIGURA 4.2 – Configurações possíveis para os rotores. . . . .	80
FIGURA 4.3 – Simulação de efeito do ângulo de rolagem sobre a projeção das características visuais na imagem. . . . .	85
FIGURA 4.4 – Diferentes curvas da função de ganho variável. Influência da variação em um dos parâmetros, mantendo os outros dois constantes: linhas azuis, variando-se $\lambda(0)$ ; verdes, $\lambda(\infty)$ ; vermelhas, $\lambda'(0)$ . . .	88
FIGURA 4.5 – Função de pertinência triangular. Para cada valor do erro $e(k)$ no instante $k$ , o respectivo fator de ponderação $\gamma_f(k)$ é dado por uma função de pertinência triangular. . . . .	89
FIGURA 5.1 – Quadrirrotor AR.Drone, versão 1.0, desenvolvido pela Parrot. (a) com capa para ambientes internos; (b) com capa para ambientes externos. . . . .	94
FIGURA 5.2 – Diagrama da estrutura interna comunicação do AR.Drone. . . . .	95
FIGURA 5.3 – Imagem capturada pela câmera frontal do AR.Drone. . . . .	96

- FIGURA 5.4 – Posição inicial (vermelho) e desejada (azul) das características na imagem para as simulações. . . . . 102
- FIGURA 5.5 – Projeções inicial (vermelho) e final (azul) das características visuais na imagem para as simulações com o sistema cinemático completamente atuado com seis graus de liberdade. (a) ganho proporcional constante com  $\lambda = 0,4$ ; (b) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) filtro *fuzzy* com  $\lambda = 0,62$ . 105
- FIGURA 5.6 – (a) evolução do ganho variável e (b) erro filtrado para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. . . . . 105
- FIGURA 5.7 – Erro e velocidades cartesianas para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com  $\lambda = 0,4$ ; (b) e (e) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 0,62$ . . . . . 106
- FIGURA 5.8 – Posição e orientação para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com  $\lambda = 0,4$ ; (b) e (e) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 0,62$ . . . . . 107
- FIGURA 5.9 – Variação do tempo de acomodação para diferentes valores do ganho ((a)) e trajetória das características visuais na imagem para  $\lambda = 3,5$  ((b)). . . . . 109
- FIGURA 5.10 – Projeções inicial (vermelho) e desejada (azul) das características visuais na imagem para as simulações com o modelo dinâmico de um quadricóptero. (a) ganho proporcional constante com  $\lambda = 1,6$ ; (b) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) filtro *fuzzy* com  $\lambda = 2,8$ . . . . . 111



FIGURA 5.11 –(a) evolução do ganho variável e (b) erro filtrado para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. . . . .	111
FIGURA 5.12 –Erro e velocidades cartesianas para as simulações com o modelo dinâmico do quadrrrotor. (a) e (d) ganho proporcional constante com $\lambda = 1,6$ ; (b) e (e) ganho variável com $\lambda(0) = 2$ , $\lambda(\infty) = 1,2$ e $\lambda'(0) = 700$ ; (c) e (f) filtro <i>fuzzy</i> com $\lambda = 2,8$ . . . . .	112
FIGURA 5.13 –Posição e orientação para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com $\lambda = 1,6$ ; (b) e (e) ganho variável com $\lambda(0) = 2$ , $\lambda(\infty) = 1,2$ e $\lambda'(0) = 700$ ; (c) e (f) filtro <i>fuzzy</i> com $\lambda = 2,8$ . . . . .	113
FIGURA 5.14 –Posição inicial (vermelho) e desejada (azul) das características da imagem para a aplicação no quadrrrotor. . . . .	115
FIGURA 5.15 –O AR.Drone no ambiente de trabalho e a imagem capturada pela câmera frontal do veículo (canto inferior esquerdo). . . . .	116
FIGURA 5.16 –Aplicação do IBVS ao quadrrrotor com ganho proporcional constante $\lambda = 1,6$ . (a) projeções inicial e final das características visuais; (b) erro das características visuais; (c) sinal de controle de rolagem e arfagem; (d) sinal de controle de velocidade vertical e velocidade de guinada. . . . .	119
FIGURA 5.17 –Projeções inicial (vermelho) e desejada (azul) das características visuais na imagem para a aplicação no quadrrrotor. (a) ganho proporcional constante com $\lambda = 3$ ; (b) ganho variável com $\lambda(0) = 2$ , $\lambda(\infty) = 1,2$ e $\lambda'(0) = 700$ ; (c) filtro <i>fuzzy</i> com $\lambda = 3$ . . . . .	120
FIGURA 5.18 –Erro para a aplicação no quadrrrotor. (a) ganho proporcional constante com $\lambda = 3$ ; (b) ganho variável com $\lambda(0) = 2$ , $\lambda(\infty) = 1,2$ e $\lambda'(0) = 700$ ; (c) filtro <i>fuzzy</i> com $\lambda = 3$ . . . . .	121

FIGURA 5.19 – Sinal de controle para a aplicação no quadrrrotor. (a) e (d) ganho proporcional constante com  $\lambda = 3$ ; (b) e (b) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 3.122$

# Lista de tabelas

TABELA 5.1 – Parâmetros intrínsecos da câmera frontal do AR.Drone identificados por calibração. . . . .	98
TABELA 5.2 – Parâmetros intrínsecos da câmera usada nas simulações. . . . .	101
TABELA 5.3 – Coordenadas das posições inicial e desejada das características visuais e erro no momento inicial para as simulações. . . . .	102
TABELA 5.4 – Coordenadas da posição desejada das características visuais para a aplicação no quadricóptero. . . . .	115

# Lista de abreviaturas e siglas

IBVS *image-based visual servoing*

IMU *inertial measurment unit*

PBVS *position-based visual servoing*

UAV *unmanned aerial vehicle*

VANT veículo aéreo não tripulado

# Lista de símbolos

<b>P</b>	ponto no espaço tridimensional
$X, Y, Z$	coordenadas no espaço Euclidiano
$\phi, \theta, \psi$	orientação em relação ao sistema de coordenadas do veículo (rolagem-arfagem-guinada)
<b>r</b>	postura do robô no espaço Euclidiano
<b>p</b>	ponto no espaço da imagem
$x, y$	coordenadas normalizadas do ponto no plano da imagem (espaço 2D)
$\mathcal{F}_c$	sistema de coordenadas da câmera
$x_c, y_c, z_c$	coordenadas da câmera
<b>s</b>	parâmetro de característica visual
<b>m</b>	parâmetro de característica visual em píxeis
$u, v$	coordenadas do píxel no plano da imagem em píxeis
$u_0, v_0$	coordenadas do ponto principal
$\bar{u}, \bar{v}$	coordenadas normalizadas do píxel em relação ao ponto principal
$f$	distância focal
$\rho_l, \rho_a$	largura e altura do píxel em metros
$p_x, p_y$	relação entre distância focal e largura e altura do píxel, respectivamente
<b>K</b>	matriz de parâmetros intrínsecos da câmera
$k$	quantidade de parâmetros das características da imagem

---

$l$	graus de liberdade do robô
$L$	matriz de interação
$Z_{o,c}$	distância entre a câmera e o objeto-alvo
$\mathbf{v}$	vetor de velocidades lineares e angulares
$\mathbf{v}$	vetor de velocidades lineares
$v_x, v_y, v_z$	componentes do vetor de velocidades lineares
$\boldsymbol{\omega}$	vetor de velocidades angulares
$\omega_x, \omega_y, \omega_z$	componentes do vetor de velocidades angulares
$\lambda$	escalar positivo, ganho do controlador
$T$	empuxo no rotor
$\tau$	torque no rotor
$Q$	arrasto aerodinâmico
$\Omega$	velocidade angular do rotor
$b$	coeficiente de empuxo
$d$	distância do rotor ao centro de massa do veículo
$m$	massa
$g$	aceleração da gravidade
$\widehat{(\cdot)}$	valor estimado
$[a \ b]^T$	vetor coluna
$(a,b)$	vetor coluna, mesmo que $[a \ b]^T$
${}^j R_i$	matriz de rotação de $i$ para $j$
${}^j T_i$	matriz de transformação homogênea de $i$ para $j$
$\mathbf{A}^{-1}$	inversa da matriz $\mathbf{A}$
$\mathbf{A}^+$	pseudo-inversa da matriz $\mathbf{A}$

---

$\mathbf{A}^T$	transposta da matriz $A$
$\widehat{\mathbf{A}}$	estimação da matriz $A$
$\times$	produto vetorial
$\cdot$	produto escalar



# 1 Introdução

Este capítulo apresenta o contexto atual da robótica móvel aérea e seus desafios surgidos a partir da segunda metade do século XX. Algumas novas necessidades da civilização moderna em relação à robótica e das vantagens oferecidas pelo uso da visão em tais sistemas são discutidas. O controle servo visual é apresentado como alternativa para algumas dessas novas demandas. O capítulo também apresenta o histórico e o estado da arte desta técnica. Especificamente, mostram-se algumas direções que o controle baseado em imagem de veículos aéreos não tripulados vem tomando. Por fim, detalha-se a estrutura desta dissertação.

## 1.1 Um novo contexto para a robótica

O termo *robô* surgiu pela primeira vez em uma peça de teatro de 1921 do dramaturgo tcheco Karel Čapek chamada *R.U.R — Rossumovi univerzální roboti*, ou *Robôs Universais de Rossum*. Nessa obra de ficção científica, uma fábrica produz pessoas artificiais a partir de matéria orgânica sintética. Assim, *robô* deriva da palavra tcheca *robotá*, que significa literalmente corveia — o trabalho gratuito que os servos deviam prestar ao senhor feudal durante três ou mais dias da semana. De forma mais abrangente, simplesmente *trabalho*.

Construções egípcias que realizavam tarefas controladas por humanos datam de até 3000 a.C.; autômatos mecânicos — sistemas capazes de reproduzir alguns comportamentos de seres vivos de forma independente — datam desde a Grécia antiga e tiveram seu apogeu na Europa Renascentista dos séculos XVII e XVIII. O próprio Leonardo da Vinci desenvolveu o projeto de um cavaleiro mecânico que seria capaz de mover os braços, a cabeça, sentar e levantar. No entanto, com a evolução da tecnologia, o conceito de robô passou de algo puramente mecânico para sistemas que incluem noções de pensamento, razão, solução de problemas e até mesmo emoções e consciência (MATARIĆ, 2007).

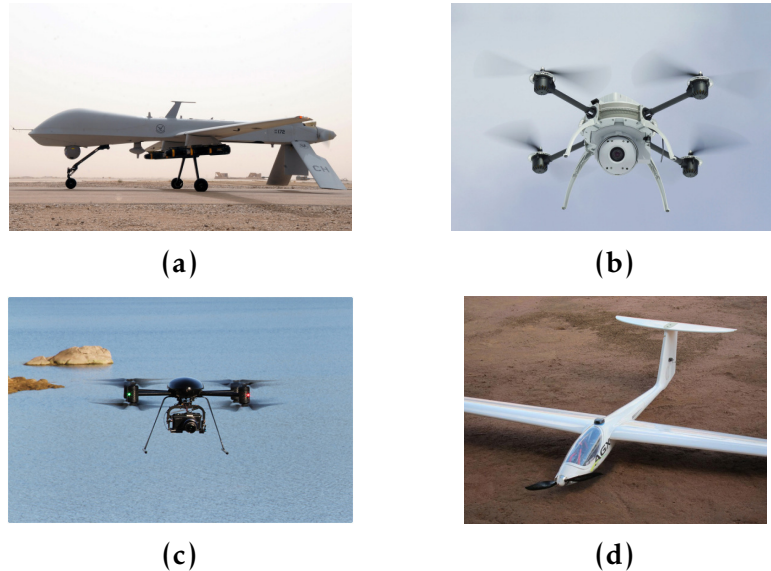
Tradicionalmente, aplicavam-se robôs quase que exclusivamente em fábricas. Entre suas missões estava retirar humanos de tarefas perigosas, repetitivas e enfadonhas, oferecendo ainda maior produtividade. Nesses casos, o cenário onde operavam precisava ser completamente controlado. Esta limitação se devia, em grande parte, à baixa capacidade dos sensores dos robôs industriais (HUTCHINSON; HAGER; CORKE, 1996). As necessidades dos humanos ditou a evolução da tecnologia. Não apenas a indústria trouxe novas exigências para os robôs, mas a própria sociedade demandou máquinas capazes de servir os humanos. Algumas dessas novas aplicações são robótica médica, de reabilitação, móvel (terrestre, aérea e subaquática), de campo, de construção e humanóide (GARCIA *et al.*, 2007).

Para serem capazes de realizar tais tarefas, novas habilidades precisam ser incorporadas aos robôs. Entre elas estão a capacidade de operar em ambientes complexos, desorganizados e desconhecidos (chamados ambientes não estruturados) de forma segura mesmo na presença de humanos (CORKE, 2011). Um requisito importante para que atinjam esse objetivo é que tenham capacidade de perceber o local em que estão

inseridos. Isso exige sensores complexos, tais como sonares, *lasers* e câmeras. Esse último, particularmente, tem atraído interesse da comunidade científica. Prova disso é a grande quantidade de pesquisas que vêm sendo desenvolvidas associando robótica e visão computacional (BONIN-FONT; ORTIZ; OLIVER, 2008). Isso se deve, em parte, ao aumento do poder computacional dos computadores, que até o fim da década de 90 exigiam um *hardware* dedicado de processamento de imagens (CORKE, 2011).

A robótica móvel, por sua vez, recebeu atenção nas últimas duas décadas. Contudo, com tal desenvolvimento da computação aliado a novas técnicas de controle, localização e navegação, essa área teve um crescimento particularmente importante no último decênio. Surgiram algumas novas aplicações científicas, como os diversos exploradores enviados a Marte (BAJRACHARYA; MAIMONE; HELMICK, 2008) e automóveis modificados para operarem de forma autônoma, como o *Boss* (BAKER; DOLAN, 2009), criado para a competição de veículos autônomos *DARPA Grand Challenge*. Sistemas comerciais se tornaram sucesso de vendas, como o aspirador de pó doméstico autônomo *iRobot Roomba*, que vendeu 1,2 milhão de unidades em quatro anos (JONES, 2006). Além da terra, robôs móveis ocuparam a água e o ar. Veículos subaquáticos, como o DEPTHX (*Deep Phreatic Thermal Explorer*), com o objetivo de procurar vida extraterrestre na lua de Júpiter, Europa (KUMAGAI, 2007). Os veículos aéreos não tripulados — VANT ou UAV (do inglês *unmanned aerial vehicle*) — também ganharam interesse. Como se movem no espaço tridimensional, seu deslocamento não é limitado como o de veículos terrestres e superam obstáculos facilmente. Assim, esse tipo de veículo tem sido usado em diferentes aplicações: inspeção de linhas de transmissão (KATRAŠNIK; PERNUŠ; LIKAR, 2010), vigilância (GROCHOLSKY *et al.*, 2006), resgate, patrulhamento, mapeamento, detecção de incêndios, uso militar, monitoramento ambiental e, até mesmo,

filmagem para cinema (BONIN-FONT; ORTIZ; OLIVER, 2008). A Figura 1.1 mostra alguns dos VANTs usados no mundo.



**FIGURA 1.1** – Alguns VANTs usados atualmente. (a) MQ-1 Predator, General Atomics; (b) Aeryon Scout, Aeryon Labs Inc.; (c) X8, DraganFly Innovations Inc.; (d) Tiriba, AGX Tecnologia Ltda.

## 1.2 Histórico do controle de robôs baseado em visão

Após anos de evolução da robótica, é de se esperar que a tarefa de mover um robô até um determinado objeto seja banal e facilmente realizada. Entretanto, na prática, os sistemas de controle movem os robôs para uma posição e orientação determinadas onde, supostamente, o objeto de referência deveria estar. Garantir que esta posição e orientação são as necessárias para “mover o robô até o objeto” não é trivial. Por exemplo, na indústria automobilística, para que a soldagem de peças da carroceria do veículo aconteça com sucesso, primeiro deve-se saber a localização das chapas de aço. Segundo, o robô precisa atingir a postura esperada para soldar. Apesar de parecerem simples, essas tarefas são, na prática, complexas. Uma solução comum para

determinar a localização das peças é usar dispositivos mecânicos e pneumáticos de posicionamento — chamados de *welding fixtures* em inglês. Tais aparatos possuem custo bastante elevado e normalmente é necessário um item distinto para cada peça que o robô precisa soldar. Alguns fatores podem fazer com que as peças não fiquem na posição esperada: os dispositivos podem apresentar problemas no funcionamento, peças podem sofrer danos e empenar, ou o simples desgaste mecânico pode reduzir a precisão destes equipamentos.

Para certificar que um robô (manipulador, como os usados na indústria automobilística) atinja a postura esperada, calculam-se os ângulos necessários de cada uma de suas juntas. Então, movem-se os atuadores para a posição correspondente. Para que a postura final da extremidade do robô seja precisa, todas as peças que o compõem também devem ser precisas: comprimentos e rigidez dos *links*, alinhamento dos eixos, *encoders*, atuadores e controladores que exigem modelos dinâmicos bastante realistas. Tudo isso aumenta consideravelmente o custo, tamanho e complexidade dessas máquinas (CORKE, 2011).

Ao dotar os robôs com sensores de visão, é possível solucionar ou, ao menos minimizar, muitos dos problemas citados anteriormente. Se os sistemas robóticos são capazes de enxergar o objeto à sua frente e essa informação pode ser usada para controlar seus movimentos, será possível “mover o robô até o objeto” mesmo sem conhecimento prévio de sua posição ou da do objeto — e ainda que o objeto esteja se movendo! Por outro lado, a precisão dos elementos que o constituem pode ser relativamente menor, já que erros de sensores, atuadores e modelos serão compensados pelo sistema visual. De fato, este conceito começou recentemente a ser utilizado na indústria automotiva, com a eliminação da necessidade dos dispositivos de fixação das etapas iniciais de soldagem

(VENABLES, 2006).

Nos primórdios das aplicações de informações visuais à robótica, usava-se uma estratégia conhecida como *ver e mover* (*look and move approach*). Tal abordagem se caracterizava por uma estrutura semelhante a um controle em malha aberta devido aos grandes tempos de ciclo (CORKE, 1993). O controlador *ver e mover* requeria que o manipulador estivesse completamente parado para que uma nova imagem fosse adquirida. Esse fato demonstra uma dura realidade da área — uma grande dependência da capacidade de processamento dos computadores. A precisão final do sistema resultava diretamente da precisão da câmera, dos sensores e controladores do robô. Shirai e Inoue (1973), pioneiros no uso dessa técnica, descreveram como usar uma malha de realimentação visual para aumentar a precisão de uma tarefa de montagem de peças. Nesse experimento o tempo de ciclo chegava a 10 s em cada uma das etapas *ver e mover*. O termo controle servo visual foi cunhado posteriormente por Hill e Park (1979) (antes era conhecido como *realimentação visual*). Esta nova nomenclatura distinguia o trabalho deles, com processamento em tempo real, dos seus precursores, operados em blocos de *ver e mover*.

Um dos motivos para os altos tempos de ciclo das primeiras aplicações era a necessidade de calcular a postura do objeto e do robô no espaço tridimensional. Até que Weiss, Sanderson e Neuman (1985) apresentaram um controlador servo visual adaptativo que usava as características da imagem diretamente na lei de controle. Esta abordagem reduziu o tempo de ciclo, já que não requeria a reconstrução tridimensional da cena, mas ainda era limitada a manipuladores com poucos graus de liberdade. Weiss, Sanderson e Neuman (1987) consolidaram uma das mais importantes classificações: controle servo visual baseado em imagem (IBVS, do inglês *image-based visual servoing*) e

baseado em posição (PBVS, de *position-based visual servoing*), detalhados no Capítulo 2. Também estabeleceram a distinção entre controle servo visual direto e a estratégia *ver e mover dinâmica*. Enquanto a primeira calcula sinais para as juntas diretamente a partir de informações visuais, a segunda define apenas sinais de velocidade, já que usa também controladores de baixo nível para os atuadores do sistema. Entretanto, essa separação está caindo em desuso já que a maioria dos controladores servo visuais atuais usam tanto realimentação visual quanto de outros tipos de sensores e controladores de baixo nível (CORKE, 2011).

Feddema e Mitchell (1989) apresentaram uma nova abordagem que usava controle em malha fechada das juntas e um gerador de trajetórias. Essa estratégia solucionou problemas relacionados às baixas taxas de amostragem do sistema visual, que ainda eram consideravelmente menores que as dos controladores dos atuadores. Eles demonstraram um IBVS para quatro graus de liberdade, sendo três de translação e um de rotação. Hashimoto *et al.* (1991) mostrou as vantagens de usar uma quantidade de características visuais maior do que o número de graus de liberdade e fez uma das primeiras comparações de desempenho entre PBVS e IBVS. Chaumette, Rives e Espiau (1991) aplicaram a abordagem de função de tarefa (SAMSON; Le BORGNE; ESPIAU, 1991) a sistemas de controle visual usando quatro pontos como características visuais. Outros tipos de características da imagem também foram demonstrados: linhas (SILVEIRA *et al.*, 2002), elipses (ESPIAU; CHAUMETTE; RIVES, 1992), momentos de imagem<sup>1</sup> (TAHRI; CHAUMETTE, 2005). Espiau, Chaumette e Rives (1992) generalizaram o IBVS para primitivas geométricas genéricas. Benhimane e Malis (2007) usaram homografia<sup>2</sup> para

---

<sup>1</sup>Momentos de imagem são análogos dos momentos mecânicos, porém descrevem o arranjo de uma forma na imagem. Alguns exemplos são a área (momento de ordem zero), o centro de massa (primeira ordem) e a orientação (segunda ordem).

<sup>2</sup>Para objetos planares ou movimentos puramente rotacionais, a matriz de homografia relaciona um mesmo objeto visto a partir de posturas diferentes da câmera. Por exemplo, sejam  $\mathbf{p}_i$  e  $\mathbf{p}_d$  as coordenadas



rastreamento e controle visual sem necessidade do modelo 3D do objeto de referência.

Westmore e Wilson (1991) incorporaram algoritmos de filtro de Kalman estendido ao PBVS para estimar a postura do robô e controlar movimentos em duas dimensões. Posteriormente, aperfeiçoaram esse trabalho para permitir o controle tridimensional e lidar com oclusões (WILSON; WILLIAMS HULLS; BELL, 1996). Papanikolopoulos, Khosla e Kanade (1993) compararam o desempenho de três controladores (PI, alocação de polos e ótimo com filtro de Kalman) numa tarefa de rastreamento visual de um objeto tridimensional em movimentos planares com uma câmera acoplada a um braço robótico. Thuilot *et al.* (2002) examinaram um dos problemas inerentes do PBVS: como manter as características no campo de visão da câmera.

Desenvolvimentos mais recentes abordaram alguns dos problemas conhecidos do IBVS e PBVS. Malis, Chaumette e Boudet (1999) propuseram mesclar as características da imagem 2D com algumas informações tridimensionais obtidas sem a necessidade de conhecimento preciso do modelo do objeto. Corke e Hutchinson (2001) mostraram como desacoplar os movimentos de translação e rotação em relação aos eixos  $x$  e  $y$  daqueles em relação a  $z$  usando apenas características visuais. Gans, Hutchinson e Corke (2003) sugeriram um método que alterna rapidamente entre PBVS e IBVS. Coordenadas cartesianas são as mais comuns, mas Iwatsuki e Okiyama (2005) usaram cilíndricas e Corke (2010) mostrou as vantagens de usar coordenadas esféricas para câmeras não perspectivas, como a *olho de peixe* e a catadióptrica. Danès e Bellot (2006) unificaram o PBVS e o IBVS com uma formulação em espaço de estados usando LDI (*Linear Differential Inclusion*) com incertezas modeladas por LFT (*Linear Fractional Transformation*). Essa abordagem, batizada de *Multicriteria Visual Servoing*, permite

---

na imagem de um ponto visto por uma câmera nas posturas inicial e desejada, respectivamente. Então  $\mathbf{p}_i$  está relacionado a  $\mathbf{p}_d$  por  $\mathbf{p}_i \propto \mathbf{H}\mathbf{p}_d$ , onde  $\mathbf{H}$  é a matriz de homografia.

tratar vários requisitos de controle, como convergência, saturação de atuadores, limites de juntas e restrições espaciais e da imagem usando LMI (*Linear Matrix Inequalities*). Silveira e Malis (2012) propuseram um método de controle servo visual que não requer informações métricas nem conhecimentos acerca das dimensões do objeto-alvo. Além disso, esse método usa informações sobre todos os píxeis na imagem e não apenas de algumas características extraídas, como na maioria das abordagens de servovisão.

O interesse e longevidade desse tema instigaram a produção de algumas revisões como Corke (1993) e Silveira *et al.* (2002), por exemplo. Hutchinson, Hager e Corke (1996) elaboraram o primeiro tutorial e textos mais atuais discutem técnicas básicas (CHAUMETTE; HUTCHINSON, 2006) e avançadas (CHAUMETTE; HUTCHINSON, 2007). Alguns livros recentes sobre robótica trazem capítulos dedicados ao controle baseado em visão. É o caso de Spong, Hutchinson e Vidyasagar (2005), que traz abordagens básicas de controladores visuais cinemáticos. Siciliano *et al.* (2009) mostram controladores que incorporam a dinâmica dos robôs, bem como informações sobre processamento de imagens, extração de características, estimação de postura e visão estéreo. Em Corke (2011) a visão tem um destaque especial e grande parte do texto é dedicada à visão computacional e controle baseado em visão.

Apesar de ter sido desenvolvido originalmente para braços manipuladores, houve avanços na aplicação de controle servo visual a robôs móveis. Tsakiris, Rives e Samson (1998) aplicaram servovisão em sistemas mecânicos não holonômicos<sup>3</sup>. Eles mostraram que os controladores clássicos usando informações visuais não são capazes de regular assintoticamente a postura desses sistemas. Como solução, propuseram um controlador

---

<sup>3</sup>Sistemas holonômicos são aqueles cujas restrições são holonômicas, ou seja, podem ser descritos apenas em função das posições que descrevem tal sistema, e não de suas velocidades. Uma consequência prática em robótica é que robôs holonômicos possuem todos os seus graus de liberdade controláveis (podem ser atuados diretamente para moverem-se em qualquer direção).

com realimentação de estados variante no tempo em uma abordagem visual. Mariotini e Prattichizzo (2008) usaram geometria epipolar<sup>4</sup> para aplicar IBVS a um veículo autônomo não holonômico usando contornos como características visuais. Na última década surgiram aplicações em plataformas aéreas. Azinheira *et al.* (2002) regularam a postura de um dirigível aéreo em relação a um alvo fixo no chão. Bourquardez e Chaumette (2007) propuseram um controlador capaz de pousar um avião automaticamente usando as linhas geradas pelas extremidades da pista de pouso como referências. Yoshihata *et al.* (2007) usaram duas câmeras estáticas no chão para controlar a postura de um helicóptero. Usando uma estratégia de controle alternante entre as câmeras, essa abordagem é robusta contra oclusões temporárias e parciais. Plinval *et al.* (2011) simularam um controlador visual usando homografia para pouso e decolagem vertical (VTOL, do inglês *vertical take-off and landing*) de um quadricóptero. Ceren e Altuğ (2011) compararam o desempenho do IBVS e de um controlador servo visual híbrido (usando também algumas informações 3D estimadas) com alvos fixos e móveis e aplicaram a um quadricóptero. Alguns estudos recentes usaram abordagens diferentes das tradicionais: Guenard, Hamel e Mahony (2008) usaram características visuais expressas em coordenadas esféricas para desacoplar movimentos de translação e rotação. O controlador não linear empregado incorpora a dinâmica completa do quadricóptero e também usa informações de outros sensores além da câmera; Bourquardez *et al.* (2009) desenvolveram alguns controladores cinemáticos usando momentos de imagem como características visuais. Aproveitando-se do fato de que alguns momentos de imagem expressos em coordenadas esféricas são invariantes à rotação, resolveram problemas de falta de robustez com coordenadas cartesianas para situações em que o plano da

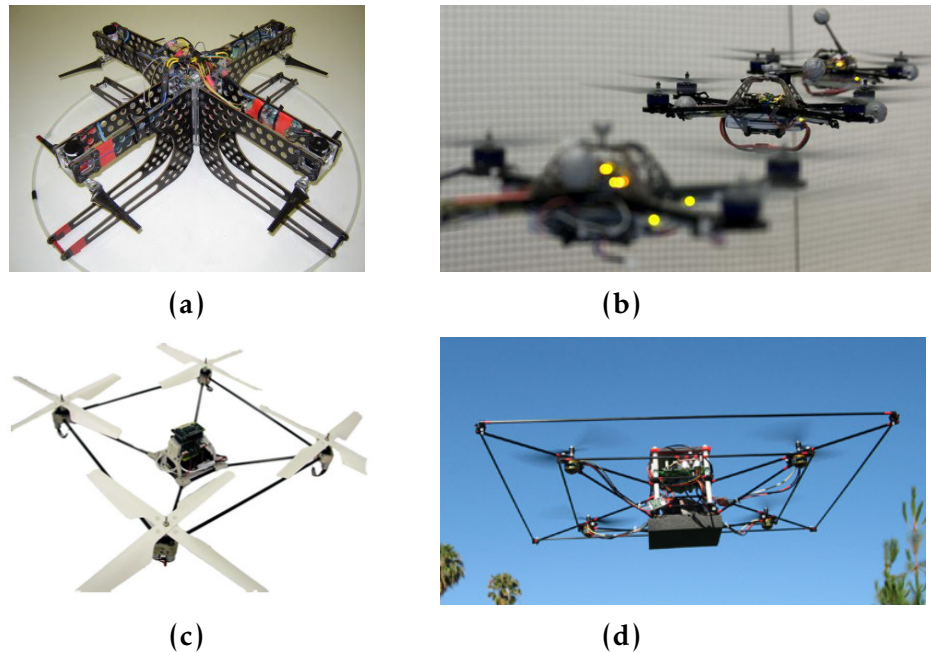
---

<sup>4</sup>Geometria epipolar é a geometria da visão estéreo. Ela estabelece as relações geométricas entre pontos no espaço tridimensional e suas projeções nos planos bidimensionais das imagens de duas câmeras em posições distintas.

imagem não está paralelo à referência.

### 1.3 Justificativa

Quadrirrotores (veículos aéreos de quatro rotores), foram considerados a plataforma padrão para pesquisa em robótica (MAHONY; KUMAR; CORKE, 2012). Suas principais vantagens são alta manobrabilidade, segurança e baixo custo. Tais propriedades os tornam em uma plataforma viável para experimentação com mapeamento, navegação e controle. O fato de moverem-se em três dimensões traz novos desafios de pesquisa quando comparados com robôs terrestres. A Figura 1.2 mostra alguns dos quadrirrotores usados em pesquisa por diferentes instituições do mundo.



**FIGURA 1.2** – Alguns quadrirrotores usados em pesquisa. (a) ANU, Austrália; (b) ETH, Suíça; (c) CEA, França; (d) Stanford, EUA.

O Laboratório de Controle do Programa de Pós-Graduação em Mecatrônica da Escola Politécnica da Universidade Federal da Bahia adquiriu recentemente novos equipamentos para fomentar a pesquisa em robótica industrial e áreas correlatas. Um

dos equipamentos incorporados foi um VANT com uma câmera embarcada fabricado pela empresa francesa Parrot, chamado de AR.Drone (o Capítulo 4 apresenta mais detalhes sobre este veículo).

Incorporar sensores visuais a robôs móveis é uma alternativa para permitir a navegação em ambientes não estruturados (BONIN-FONT; ORTIZ; OLIVER, 2008). Enquanto é possível usar controladores visuais que incorporam a dinâmica do sistema robótico, estratégias cinemáticas oferecem algumas vantagens na implementação em aparelhos comerciais: é possível trabalhar com o *firmware* do fabricante; a programação pode ser feita usando linguagem de programação de alto nível; o pesquisador pode concentrar seus esforços nos aspectos do controle visual sem considerar a — muitas vezes complexa — dinâmica do sistema.

Nesse contexto, o desenvolvimento de algoritmos de controle servo visual que podem ser testados rapidamente em sistemas comerciais de baixo custo, como o AR.Drone, são interessantes. Primeiro, por permitirem que qualquer instituição valide seus algoritmos no mesmo tipo de plataforma usada em outros lugares do mundo. Segundo, por serem independentes da dinâmica do sistema, são facilmente reutilizáveis, isto é, podem ser aplicados em outras plataformas posteriormente. Por fim, como são programados com linguagens difundidas no meio acadêmico (C/C++, Java e Python, por exemplo), pesquisadores não precisam dedicar tempo e esforço com detalhes de programação.

A revisão bibliográfica realizada neste trabalho mostrou que existem diversas abordagens atuais da técnica capazes de oferecer vantagens em relação à servovisão clássica. Porém, essas técnicas são mais complexas do que a clássica. No entanto, a plataforma experimental usada na pesquisa aqui descrita, os quadricópteros, também apresenta desafios adicionais por serem sistemas subatuado e que começou a ser usado como

plataforma de pesquisa para o controle baseado em imagem apenas recentemente. Por este ser o primeiro trabalho de aplicação do controle servo visual a quadrirrotores do Laboratório de Controle do Programa de Pós-Graduação em Mecatrônica, optou-se por usar as abordagens clássicas da servovisão.

## 1.4 Contribuição

A área de controle servo visual é multidisciplinar por natureza. Exige do seu usuário conhecimentos de teoria de controle, visão computacional e robótica. Com o desenvolvimento recente da técnica, pode ser necessário, também, conhecer algumas técnicas de inteligência artificial, identificação e modelagem de sistemas e até fotografia para superar dificuldades específicas. Após uma análise crítica da literatura própria do tema, a pesquisa aqui descrita concentrou-se em aspectos do controle, com os seguintes objetivos:

- avaliar as particularidades da aplicação de controle servo visual baseado em imagem a multirrotores;
- propor soluções para problemas particulares da aplicação do controle servo visual em multirrotores;
- avaliar o desempenho da abordagem clássica do controle servo visual baseado em imagem aplicado a multirrotores;
- analisar e aplicar controladores alternativos para melhorar o desempenho transitório e em regime da servovisão baseada em imagem aplicada em multirrotores;
- validar os controladores analisados em uma plataforma robótica experimental.

## 1.5 Estrutura da dissertação

Além desta introdução, esta dissertação está dividida em outros cinco capítulos. O Capítulo 2 apresenta o controle servo visual e suas principais classes: o PBVS e o IBVS. Discute as vantagens e desvantagens de cada uma delas e traz uma breve comparação entre as abordagens dinâmica e cinemática.

O controle servo visual baseado na imagem, IBVS, é o tema do Capítulo 3. Essa seção detalha a teoria clássica da técnica e explica sua escolha para este trabalho. Aborda questões de estabilidade e desempenho em malha fechada.

O Capítulo 4 traz detalhes específicos da aplicação do IBVS a veículos multirrotores. Analisa alternativas para melhorar o desempenho do IBVS clássico, que usa ganho proporcional constante, quando aplicado em multirrotores. Avalia duas abordagens: o ganho variável e o filtro *fuzzy*. Compara as três técnicas (clássica, ganho variável e filtro *fuzzy*) com simulações.

O Capítulo 5 valida os controladores aplicados em um quadricóptero. Primeiro feito por meio de simulações e, em seguida, através de resultados experimentais obtidos com uma plataforma disponível comercialmente. Descreve as ferramentas usadas, bem como a metodologia experimental, suas vantagens e problemas. Apresenta os resultados para os três controladores (clássico e as duas alternativas, ganho variável e filtro *fuzzy*) aplicados na plataforma experimental e traz uma análise crítica comparativa de seus desempenhos transitório e em regime. Este texto termina com uma conclusão do trabalho no Capítulo 6, que ainda propõe direções para trabalhos futuros.

## 2 Controle Servo Visual

Diversas escolhas precisam ser feitas ao se projetar um controlador baseado em visão: posição relativa entre câmera e robô, quantidade de câmeras, método de extração e escolha das características da imagem (quando for utilizada uma abordagem que requer o uso de características visuais), definição do erro e cálculo da lei de controle. A escolha desses itens pode alterar consideravelmente o desempenho final do sistema. Em um controlador servo visual, quatro etapas são claramente definidas: aquisição da imagem, processamento da imagem, definição do erro e cálculo do sinal de controle. Após apresentar os fundamentos da projeção perspectiva, este capítulo descreve as principais classificações estabelecidas na literatura para o controle servo visual, suas vantagens e desvantagens.

### 2.1 Projeção perspectiva

A formação da imagem requer a projeção do espaço tridimensional em um bidimensional — o plano da imagem. Portanto, uma dimensão é perdida e com ela a noção de profundidade. Esse processo recebe o nome de *projeção* e existem diferentes modelos para representá-lo. O mais comum para aplicações de visão computacional é a *projeção perspectiva* (SZELISKI, 2011), representado na Figura 2.1 e usado neste trabalho. Nesse



modelo, uma imagem não invertida é formada em um plano a uma distância  $f$  à frente da câmera. O sistema de coordenadas da câmera tem o eixo  $z$  perpendicular ao plano de projeção e o intersecta no ponto principal, enquanto  $x$  e  $y$  são, respectivamente, para a direita e para baixo.

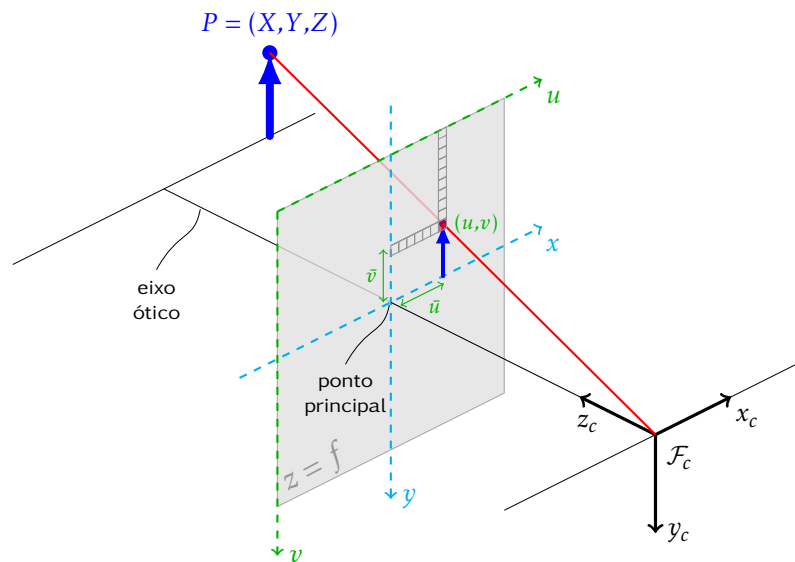


FIGURA 2.1 – Modelo de projeção perspectiva.

Na Figura 2.1, o sistema de coordenadas da câmera  $\mathcal{F}_c$ , representado com as letras minúsculas  $x_c, y_c, z_c$ , é posicionado com  $z_c$  paralelo ao eixo óptico,  $x_c$  paralelo ao eixo horizontal do plano da imagem e  $y_c$  paralelo ao eixo vertical da imagem. Ao plano da imagem, associam-se dois sistemas de coordenadas:  $xy$ , com origem no ponto principal<sup>1</sup>, fornece as posições normalizadas;  $uv$ , com origem no canto superior esquerdo, dá a localização em píxeis<sup>2</sup>. Um ponto no ambiente com coordenadas  $\mathbf{P} = (X, Y, Z)$  em relação a  $\mathcal{F}_c$  é projetado no plano da imagem com coordenadas  $\mathbf{p} = (x, y)$  dadas por

$$x = f \frac{X}{Z}, \quad y = f \frac{Y}{Z}, \quad (2.1)$$

<sup>1</sup>O ponto principal é a localização do ponto onde ocorre a interseção entre o eixo óptico e o plano da imagem.

<sup>2</sup>De acordo com o Grande Dicionário: Língua Portuguesa, Ed. Porto, 2010, a palavra que descreve a menor superfície homogênea constitutiva de uma imagem tem origem no termo da língua inglesa *pixel*. Este texto usa a versão aportuguesada descrita no referido dicionário, píxel (plural píxeis).

sendo  $f$  a distância focal. Essa relação, verificada por semelhança de triângulos, fornece as coordenadas normalizadas da imagem em relação à profundidade  $Z$  do ponto para a câmera  $P$ . A localização  $\mathbf{m} = (u, v)$  do ponto em píxeis é dada por

$$u = \frac{f}{\rho_l}x + u_0, \quad v = \frac{f}{\rho_a}y + v_0, \quad (2.2)$$

onde  $\rho_l$  e  $\rho_a$  representam a largura e altura do píxel, respectivamente e  $\mathbf{m}_0 = (u_0, v_0)$  é o ponto principal do sensor visual. Essas equações consideram uma câmera perspectiva perfeita. Para câmeras reais é necessário considerar distorções incorporando modelos não lineares. O Apêndice B aborda a modelagem de sensores visuais considerando tais distorções. Os parâmetros  $f, \rho_l, \rho_a, u_0, v_0$  são conhecidos como *parâmetros intrínsecos* e podem ser obtidos para uma câmera qualquer por um processo de calibração.

## 2.2 Aquisição da imagem

A aquisição das imagens pode ser feita usando apenas uma câmera (visão monocular), duas (visão binocular ou estéreo) ou mais câmeras. Intuitivamente, a visão monocular é mais simples e possui menor custo computacional, afinal apenas uma imagem precisa ser processada. Em contrapartida, ao se projetar o espaço tridimensional em um plano bidimensional, a informação de profundidade é perdida. Mesmo com uma única câmera, é possível estimar tal informação através da translação. Essas técnicas são conhecidas como estimaco de postura (*pose estimation* ou *Perspective-n-Point problem*), localizaco ou odometria visual. No entanto, isso implica em um processamento adicional, exige uma precisa calibrao da câmera e conhecimento prévio do modelo geométrico de um objeto de referênca capturado pela câmera.

Os seres vivos obtêm a noção de profundidade a partir de duas imagens da mesma cena fornecidas por dois olhos distintos — um processo conhecido como estereopsia. Assim, se por uma lado a visão binocular ou de múltiplas câmeras obriga o processamento de várias imagens, por outro ela oferece mais alternativas para recuperação da profundidade que o caso monocular. A reconstrução 3D (*3D reconstruction*), como essas técnicas são conhecidas, é bastante utilizada na visão robótica (CORKE, 2011). No entanto, todas elas requerem o conhecimento dos parâmetros intrínsecos da câmera com precisão.

As câmeras podem ser montadas de duas formas: diretamente no robô, recebendo o nome de *eye-in-hand* ou no ambiente de trabalho, sendo conhecida como *eye-to-hand*. No primeiro caso, a câmera enxerga o ambiente e, portanto, as características visuais usadas para o controle servo visual devem estar posicionadas de forma que permaneçam no campo de visão da câmera durante toda a tarefa. Por exemplo, pode estar presa à extremidade (*end-effector*) de um braço manipulador ou à base de um VANT. Assim, a postura da câmera em relação ao robô é fixa e geralmente conhecida. Uma das vantagens desta construção é a possibilidade de se obter uma vista com mais detalhes das características visuais. Por outro lado, para certos movimentos da câmera, tais características podem sair do campo de visão. Na montagem *eye-to-hand* a câmera está fixa no ambiente de trabalho do robô. Esta alternativa geralmente oferece uma vista global do ambiente, porém com menos detalhes. A oclusão de características visuais sempre é um potencial problema.

Montagens alternativas tentam resolver problemas dos casos descritos anteriormente. Yoshihata *et al.* (2007) usaram duas câmeras que conferem alguma robustez ao problema da oclusão da construção *eye-to-hand*. No entanto, à medida que mais câmeras são

adicionadas, mais difícil é criar grandes zonas de interseção e manter o alvo no campo de visão de todas elas. Malis, Chaumette e Boudet (2000) propuseram uma estratégia capaz de usar duas câmeras cujos campos de visão não se intersectam com a vantagem de um maior campo de visão sem a restrição de manter o objeto no campo de visão de todas elas. Com o objetivo de obter o melhor de cada alternativa, Lippiello, Siciliano e Villani (2005) sugeriram o uso concomitante de uma câmera *eye-in-hand* com várias *eye-to-hand*. Para evitar custos computacionais elevados, usaram um algoritmo que seleciona as características de imagem a serem usadas durante a execução da tarefa.

Existem diferentes tipos de lentes para câmeras comercializadas e muitas já foram usadas para controle de robôs. As mais comuns são aquelas cujas imagens podem ser modeladas pela projeção perspectiva. O campo de visão deste tipo de câmera é relativamente limitado, o que motivou o emprego de lentes grande angulares<sup>3</sup>, como a olho de peixe, e lentes catadióptricas, que combinam refração e reflexão, com o uso de espelhos.

É possível usar diferentes sistemas de coordenadas para identificar as características visuais. A mais comum são as coordenadas cartesianas, mas polares, cilíndricas e esféricas também foram usadas. Uma alternativa mais recente sugere que a imagem seja projetada em um esfera (FOMENA; CHAUMETTE, 2007) — e não em um plano, como é feito tradicionalmente. Esta abordagem elimina a ambiguidade existente entre algumas rotações e translações no IBVS (CORKE, 2010). Coordenadas não cartesianas eliminam um problema do IBVS para grandes rotações em torno do eixo ótico chamado de *camera retreat* — esse efeito conhecido do IBVS faz com que, para algumas tarefas em

---

<sup>3</sup>Grande angulares são lentes perspectivas com distância focal pequena e amplo campo de visão. Lentes olho de peixe são grande-angulares extremas. Essas lentes chegam a capturar 180°, mas suas imagens geralmente são distorcidas. Por isso não podem ser modeladas pela projeção perspectiva. Sistemas catadióptricos combinam lentes perspectivas com espelhos convexos para formar imagens de 360°, também bastante distorcidas.

que se deseja uma rotação pura, a câmera execute uma translação indesejada ao longo do eixo ótico.

## 2.3 Processamento da imagem

A pesquisa e aplicação do controle servo visual foi limitado nos primeiros anos do desenvolvimento da técnica pelo processamento da imagem. Por um lado, a principal vantagem do uso de sensores visuais é a grande quantidade de dados que fornecem (geralmente da ordem de 1–100 MB/s (CORKE, 2011)). Por outro, tal estratégia exige uma capacidade de processamento maior em comparação a outros tipos de sensores, como *encoders*, por exemplo. A maioria dos computadores pessoais de hoje é capaz de realizar a extração de características com taxas de amostragem relativamente altas.

Uma das alternativas para reduzir a quantidade de dados obtidos da imagem, extraem-se informações que podem ser usadas diretamente na lei de controle. Tais dados recebem o nome de *característica da imagem (image features)* — definidas como qualquer informação relevante que pode ser extraída da imagem. Além disso, para serem úteis ao controlador deve ser possível rastreá-las em cada quadro fornecido pela câmera e é preciso relacioná-las a algum valor numérico. O *parâmetro de uma característica da imagem* é qualquer valor real que pode ser calculado de uma ou mais características. Por exemplo, se um ponto na imagem é escolhido como característica visual, os parâmetros dessa característica podem ser as coordenadas  $(u, v)$  do centróide desse ponto. O trabalho de (SHI; TOMASI, 1994) descreve algumas características comumente usadas em visão computacional.

As características usadas geralmente são primitivas geométricas: pontos, retas e

segmentos de retas, planos, círculos e elipses, pois é relativamente simples rastreá-las usando algoritmos já disponíveis. Para tanto, é suficiente incorporar artificialmente um marcador fiducial (*fiducial marker*) à cena que será capturado pela câmera. Algumas dessas primitivas são encontradas na natureza ou em construções artificiais, como é o caso de linhas formadas pela margem de rios ou pelas extremidades de uma tubulação. Características alternativas foram usadas mais recentemente, como é o caso dos momentos de imagem (CHAUMETTE, 2004).

Primitivas geométricas e marcadores de referência são importantes para validar técnicas de controle sem se preocupar com aspectos de visão computacional. Entretanto, para criar aplicações de controle servo visual para a indústria ou que sirvam ao dia-a-dia dos humanos, é fundamental desenvolver algoritmos baseados em outros tipos de referências. Esses algoritmos devem ser robustos a ruído e oclusões, variações de iluminação e capazes de operar em ambientes desordenados. Quando o modelo geométrico do objeto a ser capturado pela câmera está disponível, é possível usar a técnica de rastreamento baseado em modelo, que é robusta contra oclusões. O estudo de Marchand e Chaumette (2004) apresenta algumas contribuições neste sentido usando estimação de movimento e faz uma revisão das técnicas de rastreamento existentes.

Cada parâmetro  $s(\mathbf{m}, \mathbf{a})$  das características da imagem depende de medidas  $\mathbf{m}$  de pontos de interesse na imagem e de um grupo de parâmetros  $\mathbf{a}$  que fornecem informações adicionais sobre o sistema. Alguns exemplos comuns para  $\mathbf{a}$  são parâmetros intrínsecos da câmera e o modelo 3D do objeto. Por sua vez,  $\mathbf{m}$  depende da postura relativa entre o objeto-alvo e a câmera. Consequentemente, para o caso *eye-in-hand*,  $\mathbf{m}$  está relacionado com a postura relativa entre o objeto-alvo e o próprio robô. Por exemplo,  $\mathbf{m}$  pode ser a posição em píxeis na imagem de um ponto de referência, enquanto  $s$

seria a localização desse ponto em coordenadas normalizadas. Nesse caso,  $\mathbf{s} = \mathbf{p} = (x, y)$ . Mais recentemente, foram desenvolvidas técnicas de controle servo visual que utilizam diretamente as intensidades dos píxeis (SILVEIRA; MALIS, 2012).

## 2.4 Definição e cálculo do erro

A definição do erro estabelece a principal distinção entre as arquiteturas de controle servo visual:

**PBVS** *Position-Based Visual Servoing* ou controle servo visual baseado em posição.

Usando informações visuais para estimar a posição e orientação da câmera, define-se o erro no espaço da tarefa (tridimensional).

**IBVS** *Image-Based Visual Servoing* ou controle servo visual baseado em imagem. Define-se o erro no espaço da imagem (bidimensional), usando parâmetros das características visuais diretamente no seu cálculo.

**Arquiteturas híbridas** Tentam aproveitar vantagens de cada umas das abordagens anteriores. Geralmente isso é feito desacoplando os movimentos de translação e rotação, usando IBVS para controlar alguns graus de liberdade e PBVS para outros. Vale destacar o controle servo visual 2 1/2 D e o particionado, que são detalhados posteriormente.

### 2.4.1 Controle servo visual baseado em posição

No controle servo visual baseado em posição, usam-se as informações obtidas da imagem — os parâmetros das características visuais, por exemplo — para estimar a

postura da câmera. Define-se o erro entre a postura atual e a desejada no espaço da tarefa  $\mathcal{T}$ . Por esse motivo, ele também é chamado de servovisão 3D. A Figura 2.2 mostra a estrutura básica de um controlador PBVS.

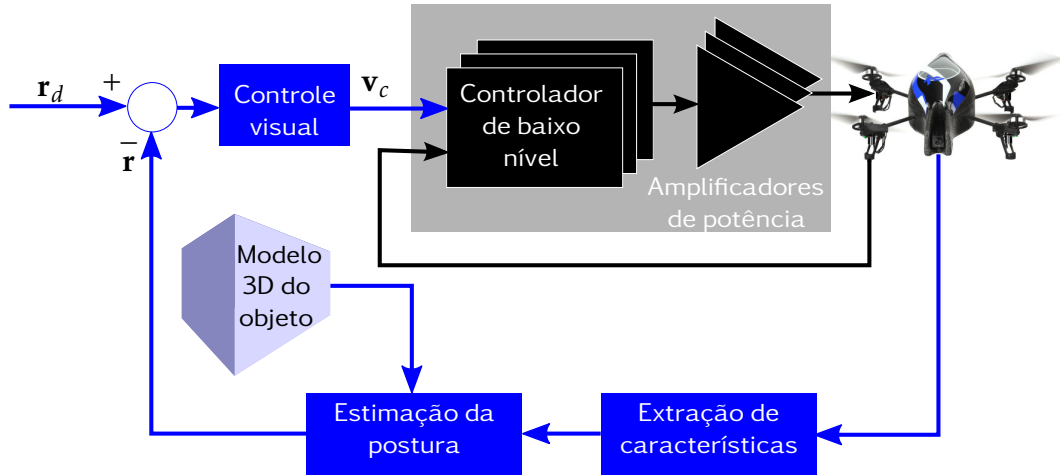


FIGURA 2.2 – Estrutura de um controlador servo visual baseado em posição.

Representa-se uma tarefa de posicionamento pela função  $\mathbf{E} : \mathcal{T} \mapsto \mathbb{R}^l$ . O objetivo é regular a postura da câmera, que pode ser representada, por exemplo, por  $\mathbf{r} = [X \ Y \ Z \ \phi \ \theta \ \psi]^T$  (onde  $X$ ,  $Y$  e  $Z$  representam a posição espacial da câmera em relação a um sistema de coordenadas fixo no ambiente e  $\phi$ ,  $\theta$  e  $\psi$  a orientação em ângulos de rolagem, arfagem e guinada) até uma postura desejada  $\mathbf{r}_d$ . Assim, o erro é definido como

$$\mathbf{E}(\mathbf{r}, t) = \mathbf{r}(t) - \mathbf{r}_d(t). \quad (2.3)$$

Existem abordagens que usam sinais de controle expressos em movimentos incrementais do robô (CORKE, 2011) e outras que fornecem o sinal de controle diretamente como um vetor de velocidades (CHAUMETTE; HUTCHINSON, 2006) (MALIS; CHAUMETTE; BOUDET, 1999). Independentemente do sinal de controle fornecido, o PBVS exige a estimação da postura a cada iteração, um problema conhecido como *localização 3D* (DEMENTHON; DAVIS, 1995). Tal estimação requer um conhecimento preciso do



modelo 3D do objeto-alvo e dos parâmetros intrínsecos da câmera. Está bem documentado na literatura que essa técnica é sensível a erros — tanto no modelo 3D do objeto, quanto nos parâmetros intrínsecos da câmera — e a ruídos na imagem (HUTCHINSON; HAGER; CORKE, 1996; CHAUMETTE; HUTCHINSON, 2006; SILVEIRA *et al.*, 2002). Apesar do constante incremento na capacidade de processamento dos computadores atuais, tal requisito ainda deve ser considerado ao se projetar um controlador visual, principalmente para sistemas embarcados. Uma vantagem dessa abordagem é o fato de ser possível controlar o movimento do robô no espaço tridimensional. Em contrapartida, não há controle sobre o movimento das características da imagem e elas podem sair de cena, fazendo com que a tarefa falhe.

### 2.4.2 Controle servo visual baseado em imagem

No controle servo visual baseado em imagem, representado na Figura 2.3, calcula-se o erro usando diretamente parâmetros das características visuais obtidas no plano da imagem. Por este motivo, também recebe o nome de servovisão 2D.

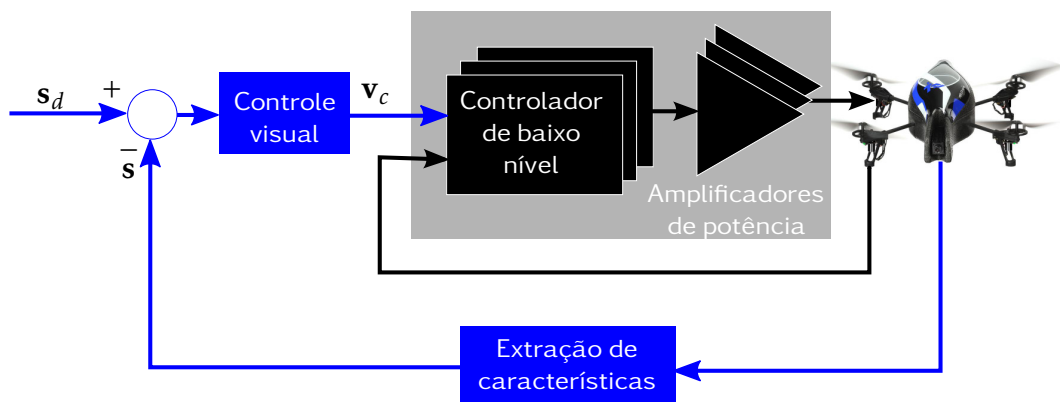


FIGURA 2.3 – Estrutura de um controlador servo visual baseado na imagem.

Para  $k$  parâmetros<sup>4</sup> de características visuais, associa-se o vetor  $\mathbf{s} = [s_1 \dots s_k]^T \in \mathcal{S} \subseteq$

<sup>4</sup>O número de *parâmetros* das características da imagem normalmente é maior do que o número de características das imagens. Como será visto mais adiante, para pontos, por exemplo, cada característica possui dois parâmetros correspondentes — um para cada eixo do plano da imagem.

$\mathbb{R}^k$  de parâmetros correspondentes, onde  $\mathcal{S}$  é o espaço de parâmetros de características da imagem. É possível definir, por exemplo, uma tarefa de posicionamento por uma função de erro  $\mathbf{e} : \mathcal{S} \mapsto \mathbb{R}^l$ , onde  $l \leq k$  é o número de graus de liberdade do robô. A função  $\mathbf{e}$  é conhecida como *função de tarefa cinemática*. O objetivo nesse caso é regular  $\mathbf{s}$  até  $\mathbf{s}_d$  para controlar implicitamente o movimento da câmera. A tarefa é considerada completa quando

$$\mathbf{e}(\mathbf{s}) = \mathbf{s} - \mathbf{s}_d = 0, \quad (2.4)$$

onde  $\mathbf{s}_d$  é o vetor de valores desejados para as características visuais. Geralmente especificam-se tais valores analiticamente usando um modelo de projeção como o da Equação 2.1 ou com o método *teach by showing*, em que se move a câmera previamente para a postura necessária e usa-se a imagem correspondente para encontrar  $\mathbf{s}_d$ . No primeiro caso é necessário conhecer os parâmetros intrínsecos da câmera.

Define-se o sinal de controle no espaço de atuadores (juntas para manipuladores, rodas ou rotores para robôs móveis, por exemplo) ou no espaço da tarefa (velocidades lineares e angulares). É necessário, então, relacionar as taxas de mudança nos parâmetros das características da imagem,  $\dot{\mathbf{s}}$ , à velocidade da câmera. Se os parâmetros visuais são as coordenadas de pontos,  $\dot{\mathbf{s}}$  é a velocidade desses pontos no plano da imagem. Escreve-se essa relação como

$$\dot{\mathbf{s}} = \mathbf{L}_s(\mathbf{r})\mathbf{v}_c, \quad (2.5)$$

onde  $\mathbf{L}_s \in \mathbb{R}^{k \times l}$  é a *matriz de interação* relacionada ao vetor de parâmetros de características visuais  $\mathbf{s}$  e  $\mathbf{v}_c = (\mathbf{v}, \boldsymbol{\omega})$  é o vetor de velocidades da câmera, composto de velocidades lineares,  $\mathbf{v} = (v_x, v_y, v_z)$ , e angulares,  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ .

A Seção 3.1.1 mostra que, para características visuais pontuais, a Equação 2.5 pode ser escrita como

$$\dot{\mathbf{s}} = \mathbf{L}_s(\mathbf{s}, Z_{o,c}) \mathbf{v}_c, \quad (2.6)$$

onde  $Z_{o,c}$  é a distância entre a câmera e o objeto-alvo.

É possível identificar algumas das principais características do IBVS. Primeiro, não há necessidade de realizar a estimação da postura nem conhecer precisamente o modelo geométrico do objeto-alvo. Segundo, a calibração da câmera não é necessária já que, como será visto mais adiante, ele é bastante robusto a erros nos parâmetros intrínsecos. Por essas particularidades, o IBVS geralmente possui um custo computacional menor do que outras arquiteturas servovisuais. Este fato contribuiu para que, historicamente, a maioria dos experimentos com controle servo visual fosse realizado com esse método — o que se verifica pela literatura reduzida do PBVS, quando comparado ao IBVS (CORKE, 2011). Mesmo com a capacidade de processamento dos computadores atuais este ainda é um fator relevante, principalmente quando se considera o desenvolvimento de controladores embarcados. Por outro lado, nos métodos clássicos há a necessidade de se obter a inversa da matriz de interação. Como o Capítulo 4 mostra, esse cálculo é relativamente simples e não implica, necessariamente, um aumento considerável do custo computacional. Contudo, é necessário conhecer ou estimar a distância entre a câmera e o objeto-alvo,  $Z_{o,c}$ . Outra característica importante do controle baseado em imagem é uma maior chance de permanência das características visuais no campo de visão da câmera. Já no espaço da tarefa não há controle algum e a trajetória do robô pode ser imprevisível. O Capítulo 4 aborda o IBVS e a matriz de interação com maiores detalhes.

### 2.4.3 Arquiteturas híbridas

Abordagens híbridas buscam tirar proveito das vantagens do IBVS e do PBVS. Muito do interesse nessas técnicas surgiu a partir da publicação de Chaumette (1998), que evidenciou problemas de estabilidade e convergência da servovisão. Esse trabalho mostra um exemplo de IBVS usando primitivas geométricas pontuais e uma tarefa que exige a rotação de  $180^\circ$  em relação ao eixo ótico. O texto mostra que o controlador tenta levar o robô ao infinito e a tarefa falha. As alternativas híbridas, de forma geral, buscam desacoplar o controle de alguns graus de liberdade para evitar esse problema. Gans, Hutchinson e Corke (2003) trazem uma comparação de desempenho de algumas estratégias híbridas.

#### 2.4.3.1 Controle servo visual 2 1/2 D

A arquitetura proposta por Malis, Chaumette e Boudet (1999) foi a primeira a propor o uso de leis de controle distintas para os movimentos de translação e rotação. Ela usa um vetor de parâmetros de características visuais semelhante ao da servovisão 2D. Porém, incorpora algumas características 3D — por isso é chamado de servovisão 2 1/2 D (*dois e meio D*). Extraem-se essas informações tridimensionais de uma matriz de homografia entre a imagem obtida na posição atual e outra na desejada. Esse método não requer precisão do modelo 3D do objeto e aumenta o domínio de estabilidade. Assim, o controlador 2 1/2 D consegue resolver alguns dos problemas do IBVS e do PBVS desacoplando os movimentos de translação e rotação. Entretanto, ainda depende da decomposição da matriz de homografia e isso pode apresentar problemas de sensibilidade a ruídos na imagem (GANS; HUTCHINSON; CORKE, 2003).

### 2.4.3.2 Controle servo visual particionado

O surgimento da servovisão 2 1/2 D motivou a busca por controladores que dependessem apenas de características da imagem. Corke e Hutchinson (2001) propuseram a separação entre o controle de translação e rotação em relação aos eixos paralelos ao plano da imagem,  $x$  e  $y$ , daqueles relacionados a  $z$ . Independentemente da escolha das características da imagem, é possível escrever a Equação 2.5

$$\dot{\mathbf{s}} = \mathbf{L}_{xy}\mathbf{v}_{xy} + \mathbf{L}_z\mathbf{v}_z, \quad (2.7)$$

onde  $\mathbf{L}_{xy}$  corresponde às colunas de  $\mathbf{L}_s$  relacionadas à translação e rotação em relação aos eixos  $x$  e  $y$  e  $\mathbf{L}_z$  às colunas relacionadas aos movimentos em relação a  $z$ . Consequentemente,  $\mathbf{v}_{xy} = (v_x, v_y, \omega_x, \omega_y)$  e  $\mathbf{v}_z = (v_z, \omega_z)$ .

Enquanto  $\mathbf{v}_{xy}$  é calculado de maneira semelhante ao IBVS, usam-se duas novas características da imagem para obter  $\mathbf{v}_z$ :  $\alpha$ , relacionada com a rotação da câmera em relação ao eixo  $z$ , é o ângulo formado entre o eixo horizontal da imagem e o segmento de reta que une duas características pontuais;  $\sigma^2$ , tem relação com a translação da câmera em  $z$ , é a área definida pelo polígono criado pelas linhas que unem as características pontuais.

A Equação 2.7 mostra que esse método desacopla explicitamente todos os movimentos em relação ao eixo  $z$  daqueles em relação aos eixos  $x$  e  $y$ . O objetivo é tratar especificamente um problema conhecido do IBVS para tarefas que exigem rotação em relação ao eixo  $z$  que são acompanhadas de uma translação até o infinito. Tal problema é discutido com detalhes no Capítulo 3.

## 2.5 Cálculo do sinal de controle

Weiss, Sanderson e Neuman (1987) classificaram os controladores baseados em visão em três categorias: ver e mover estático (*static look and move*), ver e mover dinâmico (*dynamic look and move*) e controle servo visual.

**Ver e mover estático** As etapas de estimação da postura e a de movimento do robô nunca ocorrem simultaneamente. Essa característica era decorrente de uma limitação dos computadores e robôs da época. Não se usa para designar controladores servo visuais.

**Ver e mover dinâmico** Há execução simultânea de comandos processamento de imagem e de movimento do robô. Nesse caso, os comandos enviados pelo controlador visual ao robô estão na forma de movimentos incrementais representados por pequenas diferenças de postura ou por velocidades no espaço tridimensional. Um controlador de baixo nível usa as informações do controlador visual para comandar os atuadores e executar o movimento do robô. Tal controlador de baixo nível é realimentado por sensores não visuais — *encoders*, por exemplo. Também é conhecido como controle servo visual cinemático.

**Controle servo visual** Comandos de visão e movimento são simultâneos, porém define-se o sinal de controle visual diretamente no nível dos atuadores — juntas para manipuladores e rotores para VANTs multirrotores, por exemplo. Portanto, usa apenas realimentação visual na estrutura do controlador. Essa abordagem também é chamada de controle servo visual direto ou dinâmico.

Alguns autores consideram que essa classificação vem perdendo importância, uma vez que grande parte dos controladores atuais usam tanto realimentações visuais quanto de outros tipos de sensores (CORKE, 2011). Contudo, ainda se encontram trabalhos que distinguem as abordagens cinemáticas (BENHIMANE; MALIS, 2007; CEREN; ALTUĞ, 2011) e dinâmicas (PLINVAL *et al.*, 2011; BOURQUARDEZ *et al.*, 2009).

## 2.6 Considerações

O controle servo visual é uma técnica relativamente antiga, mas permanece em desenvolvimento ativo. Este capítulo mostrou as principais etapas de um controlador que usa esta técnica. Descreveu suas duas principais classes: o IBVS, que usa apenas características na imagem; e o PBVS, que usa essas características para estimar a postura espacial do robô. Essa estimação de postura faz com que o PBVS tenha um custo computacional maior e seja mais sensível a erros de modelagem. Tais características tornam a servovisão 2D interessante para aplicação em sistemas com instrumentos de relativamente baixa precisão. Como mostra o Capítulo 5, esse é o caso da plataforma experimental usada neste trabalho e, por esses motivos, o IBVS foi a técnica usada. A seguir, o Capítulo 3 aprofunda a teoria do IBVS.

## 3 Fundamentação teórica do IBVS

Este capítulo detalha o controle servo visual baseado na imagem. Define-se a matriz de interação e mostra sua dedução para o caso de primitivas geométricas pontuais. O cálculo de  $\mathbf{L}_s$  requer o conhecimento da profundidade  $Z_{o,c}$ , que nem sempre é conhecido. Algumas alternativas para tratar a falta dessa informação são discutidas. O capítulo faz a análise da estabilidade do controlador e ilustra um caso prático de aplicação do IBVS em ambiente simulado, evidenciando alguns problemas conhecidos da técnica.

### 3.1 A matriz de interação

A Equação 2.5 mostra que a matriz de interação relaciona a velocidade das características da imagem,  $\dot{\mathbf{s}}$ , com a velocidade da câmera no espaço da tarefa,  $\dot{\mathbf{r}} = \mathbf{v}_c = (\mathbf{v}, \boldsymbol{\omega})$ .

Assim,

$$\dot{\mathbf{s}} = \mathbf{L}_s(\mathbf{r})\dot{\mathbf{r}} \quad (3.1)$$

e define-se  $\mathbf{L}_s$  como

$$\mathbf{L}_s(\mathbf{r}) = \left[ \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \right] = \begin{bmatrix} \frac{\partial s_1(\mathbf{r})}{\partial r_1} & \dots & \frac{\partial s_1(\mathbf{r})}{\partial r_l} \\ \vdots & \ddots & \vdots \\ \frac{\partial s_k(\mathbf{r})}{\partial r_1} & \dots & \frac{\partial s_k(\mathbf{r})}{\partial r_l} \end{bmatrix}, \quad (3.2)$$



onde  $k$  é a quantidade de parâmetros das características da imagem e  $l$  é o número de graus de liberdade da tarefa.

O primeiro a usar a matriz de interação foi Weiss, Sanderson e Neuman (1987), que a chamou de *feature sensitivity matrix*. Pela semelhança com a matriz jacobiana de um manipulador robótico — que relaciona velocidades de juntas com velocidade da extremidade de um braço manipulador — ela também recebe a denominação de jacobiano da imagem (HUTCHINSON; HAGER; CORKE, 1996).

### 3.1.1 Dedução da matriz de interação para pontos

Uma câmera se move com velocidade  $\mathbf{v}_c = (\mathbf{v}, \boldsymbol{\omega})$ , onde  $\mathbf{v} = (v_x, v_y, v_z)$  são as velocidades de translação e  $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$  as de rotação em relação a um sistema de coordenadas fixo no ambiente. O ponto  $\mathbf{P} = (X, Y, Z)$ , onde  $X$ ,  $Y$  e  $Z$  representam a posição desse ponto no espaço da tarefa em relação ao sistema de coordenadas da câmera (como descrito na Seção 2.1), tem projeção perspectiva  $\mathbf{p} = (x, y)$  no plano da imagem.  $\mathbf{P}$  é estático em relação a um referencial fixo no ambiente e tem velocidade aparente em relação à câmera dada por (o Apêndice A mostra a dedução dessa equação)

$$\dot{\mathbf{P}} = -\boldsymbol{\omega} \times \mathbf{P} - \mathbf{v}, \quad (3.3)$$

que equivale a

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = - \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

e pode ser escrita como o sistema de equações

$$\dot{X} = Y\omega_z - Z\omega_y - v_x, \quad (3.4)$$

$$\dot{Y} = Z\omega_x - X\omega_z - v_y, \quad (3.5)$$

$$\dot{Z} = X\omega_y - Y\omega_x - v_z. \quad (3.6)$$

A derivada temporal de (2.1) é, usando a regra do quociente,

$$\dot{x} = f \frac{\dot{X}Z - X\dot{Z}}{Z^2}, \quad \dot{y} = f \frac{\dot{Y}Z - Y\dot{Z}}{Z^2}. \quad (3.7)$$

Reescrevendo (2.1) como

$$X = \frac{xZ}{f}, \quad Y = \frac{yZ}{f},$$

lembrando que a coordenada  $Z$  está expressa em relação ao sistema de coordenadas da câmera, ou seja, representa a distância entre o objeto-alvo e a câmera,  $Z_{o,c}$ , e substituindo em (3.7) junto com (3.4)–(3.6), obtém-se

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \frac{f}{Z_{o,c}} & 0 & \frac{x}{Z_{o,c}} & \frac{xy}{f} & -\frac{f^2 + x^2}{f} & y \\ 0 & -\frac{f}{Z_{o,c}} & \frac{y}{Z_{o,c}} & \frac{f^2 + y^2}{f} & -\frac{xy}{f} & -x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.8)$$

que relaciona a velocidade da câmera com a das características da imagem em termos das coordenadas normalizadas da imagem. Esta relação pode ser escrita na forma compacta

$$\dot{\mathbf{p}} = \dot{\mathbf{s}} = \mathbf{L}_{s_1} \mathbf{v}_c. \quad (3.9)$$

Porém, sensores visuais fornecem informações em píxeis. Então, rearranjando (2.2),

$$x = \frac{\rho_l}{f} \bar{u}, \quad y = \frac{\rho_a}{f} \bar{v},$$

onde  $\bar{u} = u - u_0$  e  $\bar{v} = v - v_0$  e com derivada temporal dada por

$$\dot{x} = \frac{\rho_l}{f} \dot{\bar{u}}, \quad \dot{y} = \frac{\rho_a}{f} \dot{\bar{v}}. \quad (3.10)$$

Substituindo (3.10) em (3.8),

$$\begin{bmatrix} \dot{\bar{u}} \\ \dot{\bar{v}} \end{bmatrix} = \begin{bmatrix} -\frac{f}{\rho_l Z_{o,c}} & 0 & \frac{\bar{u}}{Z_{o,c}} & \frac{\rho_a \bar{u} \bar{v}}{f} & -\frac{f^2 + \rho_l^2 \bar{u}^2}{\rho_l f} & \bar{v} \\ 0 & -\frac{f}{\rho_a Z_{o,c}} & \frac{\bar{v}}{Z_{o,c}} & \frac{f^2 + \rho_a^2 \bar{v}^2}{\rho_a f} & -\frac{\rho_l \bar{u} \bar{v}}{f} & -\bar{u} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.11)$$

que relaciona a velocidade da câmera com a das características da imagem *expressas em coordenadas de píxel* em relação ao ponto principal. De forma concisa,

$$\dot{\mathbf{m}} = \mathbf{L}_{m_1} \mathbf{v}_c, \quad (3.12)$$

onde  $\dot{\mathbf{m}}$  representa a variação temporal de  $\mathbf{m}$  e  $\mathbf{L}_{m_1} \in \mathbb{R}^{2 \times 6}$  é a matriz de interação para um único ponto. O posto<sup>1</sup> dessa matriz é dois e a nulidade<sup>2</sup> é, portanto, igual a quatro. O espaço nulo engloba vetores de velocidades espaciais do robô que não causam movimentos na imagem. Um exemplo simples para visualizar isso é o de um ponto único no centro da imagem. Movimentos de translação e rotação no eixo  $z$  não causam

<sup>1</sup>O posto de uma matriz é o número de linhas ou colunas linearmente independentes dessa matriz.

<sup>2</sup>O espaço nulo ou *kernel* de uma matriz  $\mathbf{A}$ , representado por  $\mathcal{N}(\mathbf{A})$ , é um conjunto de vetores  $\mathbf{x}$  tal que  $\mathbf{A}\mathbf{x} = \mathbf{0}$ . A nulidade de  $\mathbf{A}$  é a dimensão do seu espaço nulo.

mudanças na posição desse ponto na imagem.

Para obter uma matriz cujo espaço nulo seja trivial<sup>3</sup> e controlar todos os graus de liberdade  $l = 6$  do robô é necessário garantir que  $k = 6$ , ou seja, usar três pontos (lembrar que cada ponto fornece dois parâmetros,  $u$  e  $v$ ) não coincidentes, não colineares<sup>4</sup> e que não pertençam a um cilindro contendo o eixo ótico. A matriz de interação pode ser obtida concatenando verticalmente as matrizes para cada ponto,

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \dot{u}_2 \\ \dot{v}_2 \\ \dot{u}_3 \\ \dot{v}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{m_1} \\ \mathbf{L}_{m_2} \\ \mathbf{L}_{m_3} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.13)$$

sendo  $\mathbf{L}_{m_i}$ , com  $i = 1, 2, 3$ , matrizes de interação para cada um dos três pontos. Genericamente para  $\frac{k}{2}$  pontos (como cada ponto possui dois parâmetros,  $\frac{k}{2}$  pontos possuem  $k$  parâmetros),

$$\begin{bmatrix} \dot{u}_1 \\ \dot{v}_1 \\ \vdots \\ \dot{u}_{k/2} \\ \dot{v}_{k/2} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{m_1} \\ \vdots \\ \mathbf{L}_{m_{k/2}} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.14)$$

ou

$$\dot{\mathbf{m}} = \mathbf{L}_m \mathbf{v}_c, \quad (3.15)$$

com  $\mathbf{L}_m \in \mathbb{R}^{k \times l}$ . Analogamente, é possível associar as diferentes matrizes  $\mathbf{L}_{s_1} \cdots \mathbf{L}_{s_{k/2}}$  para

<sup>3</sup>O espaço nulo de uma matriz  $\mathbf{A}$  é dito trivial quando a única solução possível para  $\mathbf{Ax} = \mathbf{0}$  é o próprio vetor nulo  $\mathbf{x} = \mathbf{0}$ .

<sup>4</sup>Pontos são ditos colineares quando estão posicionados sobre uma mesma reta

obter

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \vdots \\ \dot{x}_{k/2} \\ \dot{y}_{k/2} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_{s_1} \\ \vdots \\ \mathbf{L}_{s_{k/2}} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \quad (3.16)$$

ou, de forma compacta,

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v}_c. \quad (3.17)$$

### 3.1.2 Cálculo da lei de controle

Quando se deseja controlar a postura de um robô, espera-se obter as velocidades no espaço da tarefa que ele deve realizar para atingir um determinado objetivo. Portanto, o problema a ser solucionado é inverso ao expresso em (3.17), ou seja,

$$\mathbf{v}_c = \mathbf{L}_s^{-1} \dot{\mathbf{s}}. \quad (3.18)$$

Para o caso genérico de  $\frac{k}{2} \geq 3$  e, conseqüentemente,  $k \geq l = 6$ , representado em (3.16),

$$\mathbf{v}_c = \mathbf{L}_s^+ \dot{\mathbf{s}}, \quad (3.19)$$

onde  $\mathbf{L}_s^+$  é a pseudo-inversa de  $\mathbf{L}_s$  dada por  $\mathbf{L}_s^+ = (\mathbf{L}_s^T \mathbf{L}_s)^{-1} \mathbf{L}_s^T$ .

Em sistemas reais é difícil saber o valor exato tanto de  $\mathbf{L}_s$  quando  $\mathbf{L}_s^+$  devido a erros de calibração da câmera, ruído na imagem e à dependência do valor  $Z_{o,c}$ . Por esse motivo, usa-se um modelo, estimação ou aproximação de  $\mathbf{L}_s^+$ , representados por  $\widehat{\mathbf{L}}_s^+$ :

$$\mathbf{v}_c = \widehat{\mathbf{L}}_s^+ \dot{\mathbf{s}}. \quad (3.20)$$

Para alvos estáticos, a derivada de (2.4) em relação ao tempo é

$$\dot{\mathbf{e}} = \dot{\mathbf{s}}, \quad (3.21)$$

que, substituída em (3.20), fornece

$$\mathbf{v}_c = \widehat{\mathbf{L}}_s^+ \dot{\mathbf{e}} \quad (3.22)$$

e pode ser descrita genericamente por

$$\mathbf{v}_c = h(\widehat{\mathbf{L}}_s^+ \mathbf{e}), \quad (3.23)$$

em que  $h$  é uma função usada para regular  $\mathbf{s}$  a  $\mathbf{s}_d$ . Tal função pode ser um simples ganho proporcional ou outras mais complexas, tais como, por exemplo, de controle ótimo, adaptativo ou não linear. Com o objetivo de se obter um decaimento exponencial desacoplado do erro, faz-se  $\dot{\mathbf{e}} = -\lambda \mathbf{e}$  (onde  $\lambda$  é um escalar positivo) e a lei de controle é obtida por (CHAUMETTE; HUTCHINSON, 2006)

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e}. \quad (3.24)$$

### 3.1.3 Interpretação geométrica da matriz de interação

A análise da matriz de interação para pontos (Equação 3.11) mostra que o movimento associado da característica visual pontual depende das coordenadas  $(\bar{u}, \bar{v})$  do ponto, da profundidade  $Z_{o,c}$  do objeto-alvo até o plano da imagem e dos parâmetros intrínsecos  $(\rho_l, \rho_a, u_0, v_0, f)$  da câmera. Cada coluna de  $\mathbf{L}_s$  relaciona a velocidade da característica

pontual ao componente respectivo de  $\mathbf{v}_c$ . Por exemplo, observando a primeira coluna de (3.11) nota-se que, para um valor unitário de  $v_x$ , a velocidade linear de  $\mathbf{s}$  em relação ao eixo  $u$  será de  $-\frac{f}{\rho_l Z_{o,c}}$ . As coordenadas  $X$  e  $Y$  não interferem na velocidade das características e as três primeiras colunas, relacionadas à translação, são inversamente proporcionais a  $Z_{o,c}$ . Tais fatos permitem fazer as seguintes conclusões:

**Translações nos eixos  $x$  e  $y$**  Translações da câmera no eixo  $x$  e  $y$  causam um movimento da característica na mesma direção, porém em sentido contrário. Isso é verificado pelo sinal negativo nas posições (1,1) e (2,2) de (3.11) (onde o primeiro termo dentro de cada parênteses representa a linha da matriz e o segundo a coluna). Além disso, o módulo dessa translação independe da posição da projeção,  $(\bar{u}, \bar{v})$ .

**Translação pura no eixo  $x$  ou  $y$**  Mover a câmera para a direita ou para a esquerda não altera a posição vertical do ponto na imagem ( $\bar{v}$ ). Analogamente, movimentos verticais não alteram a posição horizontal. Verifica-se isso através do valor nulo nas posições (1,2) e (2,1) de (3.11).

**Translação no eixo  $z$**  Faz com que as projeções se aproximem ou afastem das extremidades da imagem. Há uma exceção para projeções sobre o ponto principal. Nesse caso, não há alteração na imagem.

**Rotação em  $z$**  Gera movimentos de sinais opostos para  $u$  e  $v$ , ou seja, causa rotação em relação ao ponto principal. Novamente há uma exceção para pontos sobre o ponto principal. Nesse condição,  $\bar{u} = \bar{v} = 0$ . Esta característica é crucial para a compreensão de alguns problemas de desempenho mostrados a seguir.

**Rotações em  $x$  ou  $y$**  Em algumas condições, rotações em relação a  $x$  causam movimentos semelhantes a translações em  $y$ . Para distâncias focais grandes, o termo (1,4) se

aproxima de zero e o valor da quarta coluna se aproxima da segunda. Esse efeito é mais proeminente para valores pequenos de  $\bar{u}$  e  $\bar{v}$ , ou seja, para pontos que estão próximos do ponto principal. Movimentos de rotação em relação a  $y$  sofrem um efeito análogo em relação a translações em  $x$ .

### 3.1.4 Estabilidade

Chaumette e Hutchinson (2006) apresentaram uma análise da estabilidade usando o segundo método de Lyapunov. Considerando a função candidata de Lyapunov definida pelo norma do erro quadrático

$$\mathcal{L} = \frac{1}{2} \|\mathbf{e}(t)\|^2, \quad (3.25)$$

cuja derivada temporal é

$$\dot{\mathcal{L}} = \mathbf{e}^T \dot{\mathbf{e}}. \quad (3.26)$$

Lembrando que para alvos estáticos  $\dot{\mathbf{e}} = \dot{\mathbf{s}}$  e substituindo (3.17) e (3.24) em (3.26),

$$\dot{\mathcal{L}} = -\lambda \mathbf{e}^T \mathbf{L}_s \widehat{\mathbf{L}}_s^+ \mathbf{e}. \quad (3.27)$$

Para garantir a estabilidade assintótica *global* do sistema é necessário que  $\dot{\mathcal{L}} < 0$ , considerando que a restrição de visibilidade dos objetos-alvo estejam garantidas (ou seja, os alvos não saem do campo de visão da câmera). Uma condição suficiente para isso é

$$\mathbf{L}_s \widehat{\mathbf{L}}_s^+ > 0. \quad (3.28)$$

Para os casos em que o número de características visuais é igual ao de graus de



liberdade ( $k = l = 6$ ) e as características visuais e o controlador são projetados de forma que  $\mathbf{L}_s$  e  $\widehat{\mathbf{L}}_s^+$  possuem posto cheio e igual a seis, a condição (3.28) é garantida se a aproximação  $\widehat{\mathbf{L}}_s^+$  for precisa, ou seja,  $\mathbf{L}_s \widehat{\mathbf{L}}_s^+ = \mathbf{I} > 0^5$ .

Para três primitivas geométricas pontuais, a matriz de interação pode se tornar singular caso os pontos sejam colineares ou pertençam a um cilindro contendo o eixo ótico (MICHEL; RIVES, 1993). Ademais, é possível provar que com apenas três pontos existem quatro posturas distintas da câmera que geram os mesmos valores de  $\mathbf{s}$  (e para os quais, conseqüentemente,  $\mathbf{e} = 0$ ). Ou seja, existem quatro mínimos globais e é impossível diferenciá-los (CHAUMETTE; HUTCHINSON, 2006). A matriz também pode se tornar quase singular caso usem-se apenas três pontos (HASHIMOTO; EBINE; KIMURA, 1996). Pelos motivos citados, é comum usar quatro ou mais pontos.

Em contrapartida, com  $k > 6$  a condição (3.28) nunca será satisfeita, uma vez que o posto de  $\mathbf{L}_s \widehat{\mathbf{L}}_s^+ \in \mathbb{R}^{k \times k}$  é, no máximo, igual a  $l = 6^6$  e seu espaço nulo é não trivial. Isso implica que as posturas em que  $\mathbf{e} \in \mathcal{N}(\widehat{\mathbf{L}}_s^+)$  correspondem a mínimos locais. Uma alternativa nesse caso é avaliar a estabilidade assintótica *local*. Definindo uma nova função de erro  $\mathbf{e}' = \widehat{\mathbf{L}}_s^+ \mathbf{e}$ , sua derivada temporal é

$$\dot{\mathbf{e}}' = \widehat{\mathbf{L}}_s^+ \dot{\mathbf{e}} + \dot{\widehat{\mathbf{L}}_s^+} \mathbf{e}. \quad (3.29)$$

Substituindo (3.17) em (3.29), obtém-se

$$\dot{\mathbf{e}}' = (\widehat{\mathbf{L}}_s^+ \mathbf{L}_s + \mathbf{O}) \mathbf{v}_c, \quad (3.30)$$

<sup>5</sup>Qualquer que seja a matriz  $\mathbf{A}$ ,  $\mathbf{A}^T \mathbf{A}$  é semidefinida positiva e o  $\text{posto}(\mathbf{A}) = \text{posto}(\mathbf{A}^T \mathbf{A})$ .

<sup>6</sup>Sejam as matrizes  $\mathbf{A} \in \mathbb{R}^{a \times b}$  e  $\mathbf{B} \in \mathbb{R}^{b \times c}$ , o  $\text{posto}(\mathbf{AB}) \leq \min(\text{posto}(\mathbf{A}), \text{posto}(\mathbf{B}))$ .

onde  $\mathbf{O} \in \mathbb{R}^{6 \times 6}$  é igual a zero quando  $\mathbf{e} = 0$ , qualquer que seja  $\widehat{\mathbf{L}}_s^+$ . Usando a lei de controle (3.24) e lembrando da definição de  $\mathbf{e}'$ ,

$$\dot{\mathbf{e}}' = -\lambda(\widehat{\mathbf{L}}_s^+ \mathbf{L}_s + \mathbf{O})\mathbf{e}', \quad (3.31)$$

que é assintoticamente estável numa vizinhança em torno de um ponto.

$$\widehat{\mathbf{L}}_s^+ \mathbf{L}_s > 0, \quad (3.32)$$

com  $\widehat{\mathbf{L}}_s^+ \mathbf{L}_s \in \mathbb{R}^{6 \times 6}$ .

Mais uma vez, caso as características visuais e o esquema de controle sejam projetados de forma que  $\mathbf{L}_s$  e  $\widehat{\mathbf{L}}_s^+$  possuem posto cheio e igual a seis, então a condição (3.32) é garantida se a aproximação  $\widehat{\mathbf{L}}_s^+$  for precisa.

Apesar da estabilidade assintótica *local* ser garantida quando  $k \geq 6$ , não é possível garantir a estabilidade assintótica *global* do IBVS. Por outro lado, determinar o tamanho da vizinhança onde a estabilidade e a convergência são garantidas ainda é um problema em aberto. No entanto, em relatos de aplicações práticas verifica-se que essa vizinhança não é pequena, fato que tem favorecido os resultados obtidos. Para mais detalhes sobre a análise da estabilidade, o leitor deve verificar Chaumette e Hutchinson (2006). Espiau, Chaumette e Rives (1992) fazem uma análise da estabilidade usando a função de tarefa e Samson, Le Borgne e Espiau (1991) apresentam uma análise genérica da estabilidade de tarefas robóticas.

### 3.1.5 Alternativas para obtenção da matriz de interação

Já que o valor real de  $\mathbf{L}_s$  é desconhecido, é necessário estimar ou aproximar seu valor.

A lista abaixo relaciona algumas das alternativas descritas na literatura:

$\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{Z}_{o,c})$  Deve-se atualizar a matriz de interação a cada iteração usando (3.11).

É necessário obter os parâmetros das características da imagem e estimar a profundidade  $Z_{o,c}$  com métodos de estimação de postura continuamente durante toda a tarefa. Em relação à estabilidade, esta escolha parece ser vantajosa, pois, idealmente,  $\widehat{\mathbf{L}}_s^+ \mathbf{L}_s = \mathbf{I}^7$ . Isso faz com que características visuais tenham trajetórias mínimas entre a posição inicial e final, sendo retas quando as propriedades dinâmicas do robô permitir. No entanto, pode causar movimentos inadequados da câmera para garantir tais trajetórias mínimas das características visuais. Como será visto adiante, existe também a possibilidade de atingir posturas que levam a mínimos locais e singularidades da tarefa.

$\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{Z}_{d,o,c})$  A matriz de interação é constante ao longo de toda a tarefa e determinada em uma etapa anterior à execução da servovisão. Calcula-se usando (3.11) com valores dos parâmetros das características visuais e da profundidade na postura final. Para condições longe da desejada e que exijam grandes deslocamentos da câmera, as trajetórias no espaço da imagem são imprevisíveis (ao contrário da opção anterior, cujas trajetórias se aproximam a retas). No entanto, a trajetória no espaço da tarefa tende a ser melhor, já que as restrições na imagem não são tão rígidas.

$\widehat{\mathbf{L}}_s^+ = \frac{1}{2}(\mathbf{L}_s(\mathbf{s}, \widehat{Z}_{o,c}) + \mathbf{L}_s(\mathbf{s}_d, \widehat{Z}_{d,o,c}))^+$  Faz-se uma média aritmética dos dois casos anteriores.

<sup>7</sup> $\mathbf{I}$  é uma matriz identidade de dimensão apropriada.

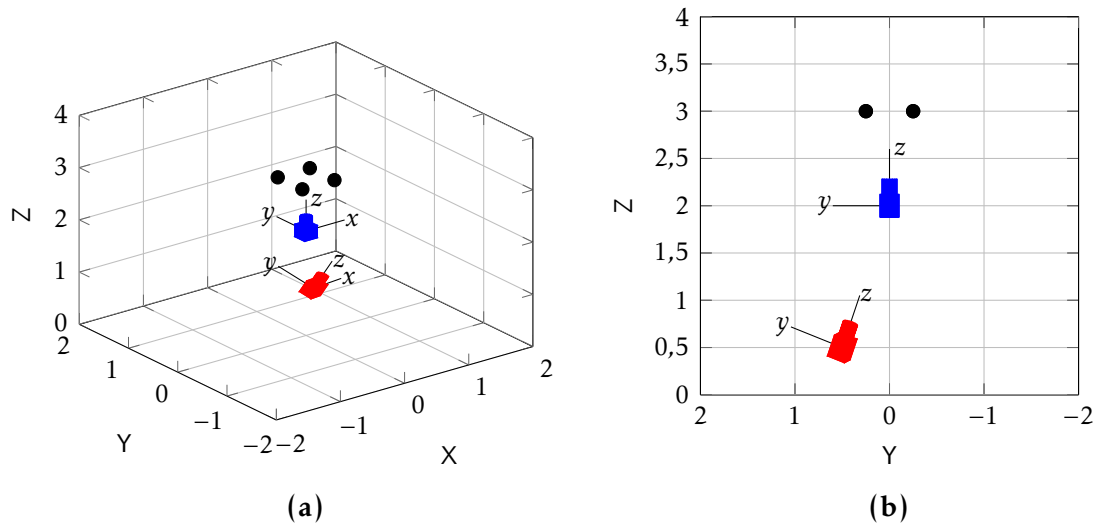
Os componentes da velocidade da câmera não sofrem grandes oscilações e fornecem trajetórias adequadas tanto na imagem quanto no espaço 3D. Relatos de experimentos práticos demonstram um bom desempenho dessa abordagem.

Outras estratégias estimam a matriz de interação numericamente, sem levar em consideração sua forma analítica dada pela Equação 3.11 e, portanto, não requerem conhecimento prévio dos parâmetros da câmera: Hosoda e Asada (1994) usaram algoritmos de mínimos quadrados no tempo discreto, Miura *et al.* (2005) usaram técnicas de otimização, Gonçalves *et al.* (2008) usaram lógica fuzzy para obter diretamente uma estimação da inversa da matriz de interação.

### 3.1.6 Aplicação do IBVS

O exemplo a seguir demonstra uma aplicação simulada da servovisão 2D em uma tarefa de posicionamento de um braço manipulador clássico. O objetivo é mover a câmera localizada inicialmente em  $X = Y = Z = 0,5\text{m}$  em relação a um sistema de coordenadas fixo no ambiente e rotacionada de  $15^\circ$  em relação ao eixo  $y$  e  $18^\circ$  em relação aos eixos  $x$  e  $z$  ( $x$ ,  $y$  e  $z$  são eixos do sistema de coordenadas fixo na câmera). A postura desejada é em  $X = Y = 0$ ,  $Z = 2\text{m}$  e sem qualquer rotação. O objeto-alvo é fixo no ambiente e composto de quatro círculos cujos centros são posicionados nos vértices de um quadrado com  $50\text{cm}$  de lado, paralelo ao plano  $XY$  e em  $Z = 3\text{m}$ . A Figura 3.1 mostra a postura inicial e desejada da câmera.

A câmera tem lente perspectiva com distância focal de  $8\text{mm}$ ,  $1024 \times 1024$  píxeis com  $10\mu\text{m}$  de lado e ponto principal localizado em  $(u, v) = (512, 512)$ . As coordenadas iniciais dos quatro pontos de referência são:  $(64, 648)$ ,  $(86, 823)$ ,  $(272, 768)$ ,  $(239, 605)$ . Na



**FIGURA 3.1** – Vista tridimensional da postura inicial (vermelho) e desejada (azul) da câmera. (a) vista a 45°; (b) vista lateral paralela ao plano YZ.

posição desejada, suas projeções são: (216,54), (91,54), (91,179), (216,179).

Define-se a matriz de interação para pontos como  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d_o,c})$ . Assim, para  $\widehat{\mathbf{Z}}_{d_o,c} = 1$  m, ela possui valor constante ao longo de toda a tarefa. Usando Equação 3.11 para quatro pontos, obtém-se:

$$\widehat{\mathbf{L}}_s^+ = \begin{bmatrix} -800 & 0 & -800 & 0 & -800 & 0 & -800 & 0 \\ 0 & -800 & 0 & -800 & 0 & -800 & 0 & -800 \\ -200 & -200 & -200 & 200 & 200 & 200 & 200 & -200 \\ 50 & 850 & -50 & 850 & 50 & 850 & -50 & 850 \\ -850 & -50 & -850 & 50 & -850 & -50 & -850 & 50 \\ -200 & 200 & 200 & 200 & 200 & -200 & -200 & -200 \end{bmatrix}$$

O ganho  $\lambda$  é escolhido de forma que o tempo de acomodação seja inferior de 8 s, definido como atingido quando o erro é menor do que 5% do erro inicial. Como trata-se de IBVS, o erro é definido no espaço da imagem e, portanto, usou-se valores em píxeis para calcular esse indicador. Utiliza-se como valor de referência o menor erro inicial em cada um dos eixos da imagem. Como há quatro características visuais e cada uma delas possui uma coordenada  $u$  e outra  $v$ , só se considera que o erro do sistema está abaixo do

---

valor determinado quando todos os valores  $u$  e  $v$  de todos os pontos são menores que esta meta. Nesse caso,  $\lambda = 0,02$ . A Figura 3.2 apresenta os resultados dessa simulação.

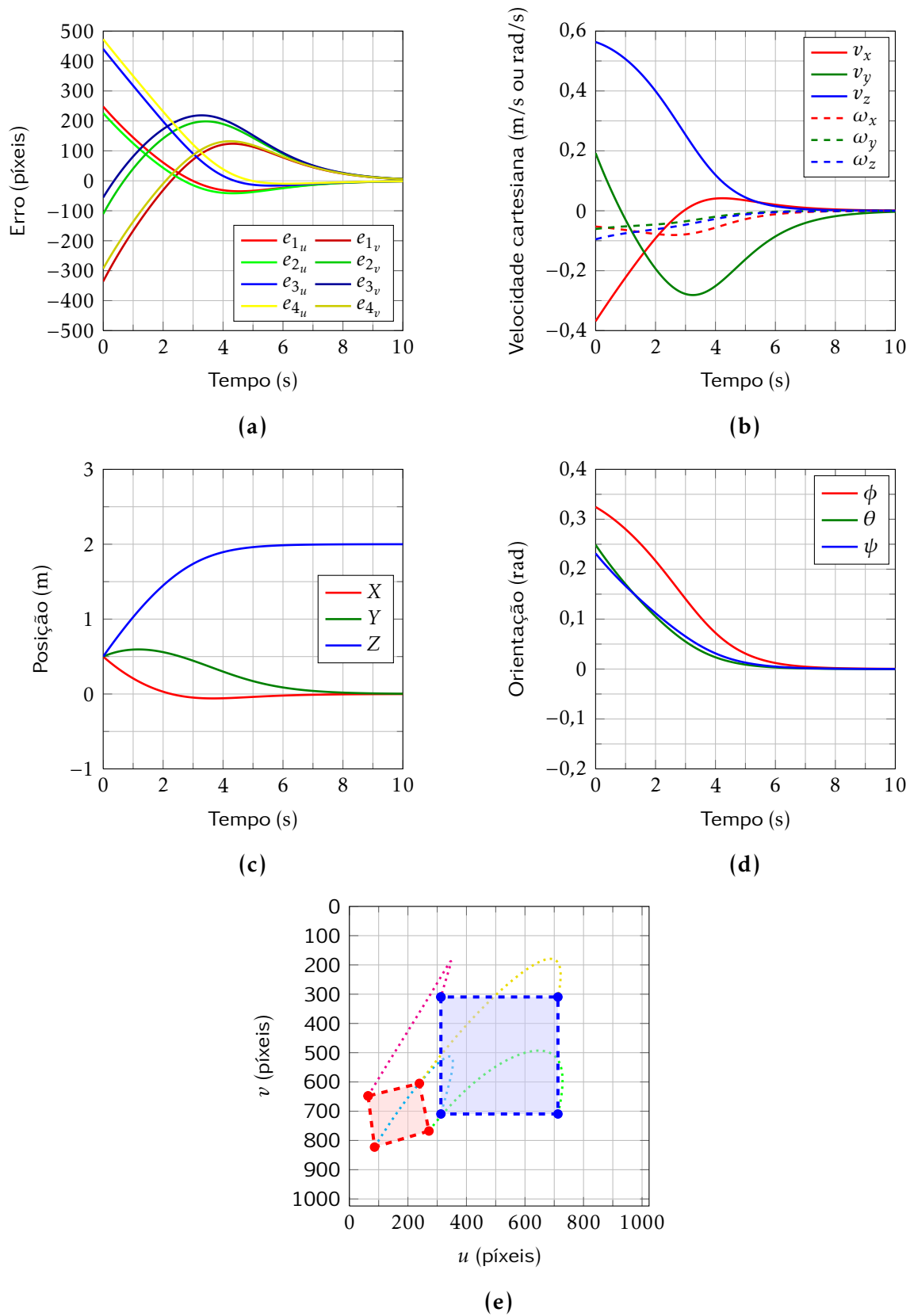


FIGURA 3.2 – Resultados da simulação de IBVS para  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{z}}_d)$ . (e) Projeção inicial (vermelho) e desejada (azul) do objeto-alvo no plano da imagem com a trajetória de cada ponto (linha tracejada preta); (b) velocidades da câmera no espaço cartesiano; (c) posição e (d) orientação da câmera ao longo da tarefa.

### 3.1.7 Problemas de estabilidade e convergência

Fazer com que  $k > 6$  geralmente é suficiente para garantir a estabilidade do controlador (mesmo que em uma vizinhança de  $\mathbf{s}_d$ ) e convergência do erro em uma tarefa de controle servo visual baseado na imagem. No entanto, Chaumette (1998) mostrou que, independentemente do número de características visuais e da configuração destas, a matriz de interação pode se tornar singular durante uma tarefa de servovisão. O referido trabalho de Chaumette traz uma análise detalhada dos problemas de estabilidade e convergência do controle servo visual.

Um dos problemas observados por Chaumette é a possibilidade de se atingir posturas que representam mínimos locais. Nesses casos, o robô permanece parado mesmo em posturas em que o erro não seja nulo. Uma alternativa para evitar tais ocorrências é escolher  $\mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d,c})$ , o que permitiria que o sistema evite os mínimos locais. Por esse motivo, como será visto no Capítulo 5, essa foi a estimação usada nos experimentos realizados.

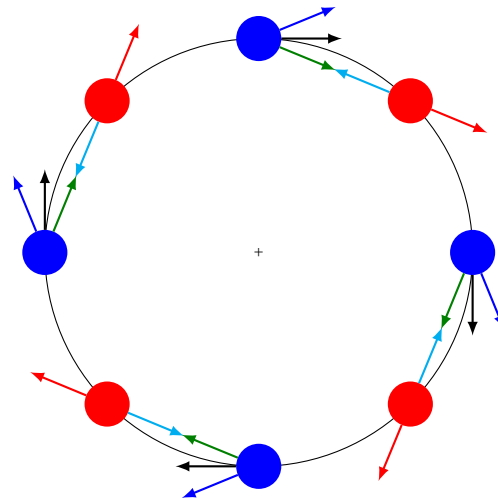
Outro problema apontado por Chaumette foi a possibilidade de movimentos indesejados no espaço da tarefa, denominados movimentos cartesianos subótimos. A análise realizada em Chaumette e Hutchinson (2006) é detalhada a seguir.

#### 3.1.7.1 Movimentos Cartesianos subótimos

Para analisar a ocorrência de movimentos cartesianos subótimos, considera-se uma tarefa como a exemplificada na Figura 3.3, onde a posição inicial está representada por pontos azuis e as desejadas por vermelhos.

Como descrito anteriormente, o uso de  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$  procura obter um decaimento

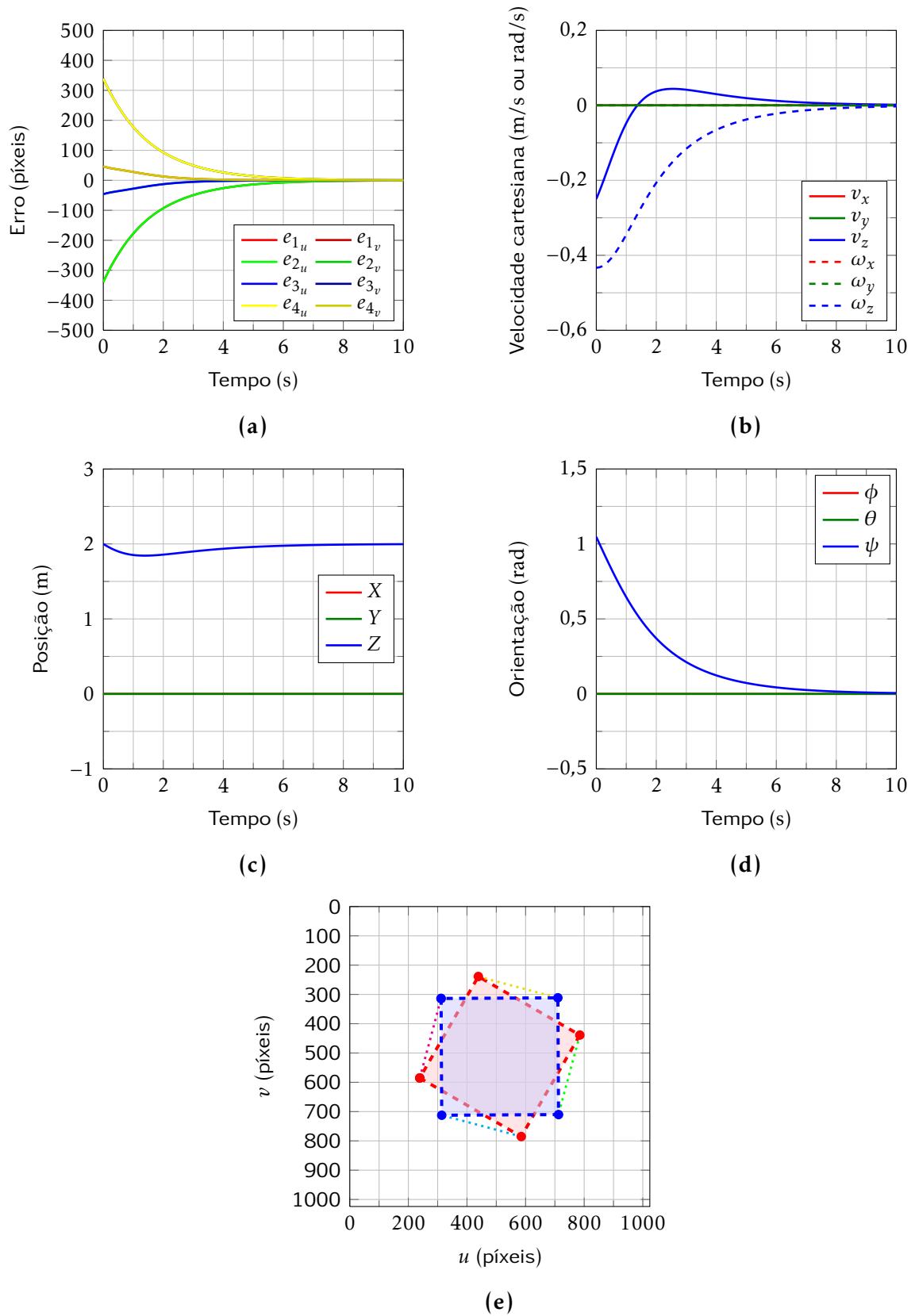




**FIGURA 3.3** – Movimentos esperados em tarefas de rotação pura para diferentes estimativas da matriz de interação. Pontos azuis representam a posição inicial e pontos vermelhos a posição desejada. Adaptada de Chaumette e Hutchinson (2006).

exponencial do erro  $\mathbf{e}$ . Nesse caso, as trajetórias no plano da imagem são aproximadamente retas. Portanto, para movimentos de rotação pura em relação ao eixo ótico, o movimento das características é representado pelas setas verdes — retas da posição inicial para a final. O movimento da câmera necessário para garantir tal deslocamento na imagem é uma rotação em relação ao eixo ótico acompanhada de um recuo translacional também ao longo do eixo ótico, seguido de uma nova translação de aproximação para concluir a tarefa. Este efeito, evidenciado na Figura 3.4, recebe o nome de recuo da câmera (*camera retreat*) (CORKE; HUTCHINSON, 2001).

No caso extremo em que a rotação necessária é de  $180^\circ$ , cada característica visual com coordenadas iniciais  $(\bar{u}, \bar{v})$  terá como coordenadas desejadas  $(-\bar{u}, -\bar{v})$ . Para que as trajetórias sejam retas, as características visuais deveriam passar pelo ponto principal da imagem (uma singularidade da matriz de interação). Nessa condição, a câmera executaria uma translação pura, afastando-se do objeto até o infinito (e sem qualquer movimento de rotação). De fato, analisando a Equação 3.11 com as características localizadas no ponto principal e a câmera no infinito do eixo  $Z$ , a matriz de interação



**FIGURA 3.4** – Efeito de recuo da câmera devido a rotação pura de  $60^\circ$  em relação ao eixo  $z$  com  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A câmera se afasta do objeto-alvo na direção do eixo óptico.

para cada ponto  $i$  é

$$\mathbf{L}_{s_i} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \quad (3.33)$$

Nessa condição, tanto  $\mathbf{L}_s$  quanto  $\widehat{\mathbf{L}}_s^+$  têm posto dois, o que corresponde a uma singularidade. A condição (3.32) não é assegurada e a tarefa falha, como mostra a Figura 3.5. Corke e Hutchinson (2001) apresentam um modelo geométrico desse efeito que suporta tais conclusões.

Caso se use  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{Z}_{d_o,c})$ , o movimento das características visuais será o representado pelas setas azuis na Figura 3.3. Verifica-se esse fato considerando, inicialmente, uma tarefa oposta à anterior, com as características saindo de  $\mathbf{s}_d$  até  $\mathbf{s}$ . A lei de controle será

$$\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+(\mathbf{s}_d - \mathbf{s}), \quad (3.34)$$

que, mais uma vez, causará inicialmente trajetórias retas dos pontos vermelhos para os pontos azuis, representadas pelas setas de cor ciano na Figura 3.3. De volta ao problema original, a velocidade da câmera será exatamente a oposta, já que  $(\mathbf{s} - \mathbf{s}_d) = -(\mathbf{s}_d - \mathbf{s})$ . Ou seja, o movimento no plano da imagem será aquele indicado pelas setas vermelhas nos pontos vermelhos que, transformados para os pontos azuis, são as setas também azuis. Isso significa que o movimento da câmera será composto de uma rotação em relação ao eixo ótico combinado com uma translação de aproximação até o objeto-alvo, como mostra a Figura 3.6. Mais uma vez, no caso de rotações puras de  $180^\circ$  em relação ao eixo ótico, o movimento será de translação pura de aproximação, sem qualquer rotação. Nessa condição,  $Z_{o,c} = 0$  e a matriz de interação para cada ponto que não esteja sobre o ponto principal será

$$\mathbf{L}_{s_i} = \begin{bmatrix} \infty & 0 & \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty & \infty & \infty \end{bmatrix}. \quad (3.35)$$

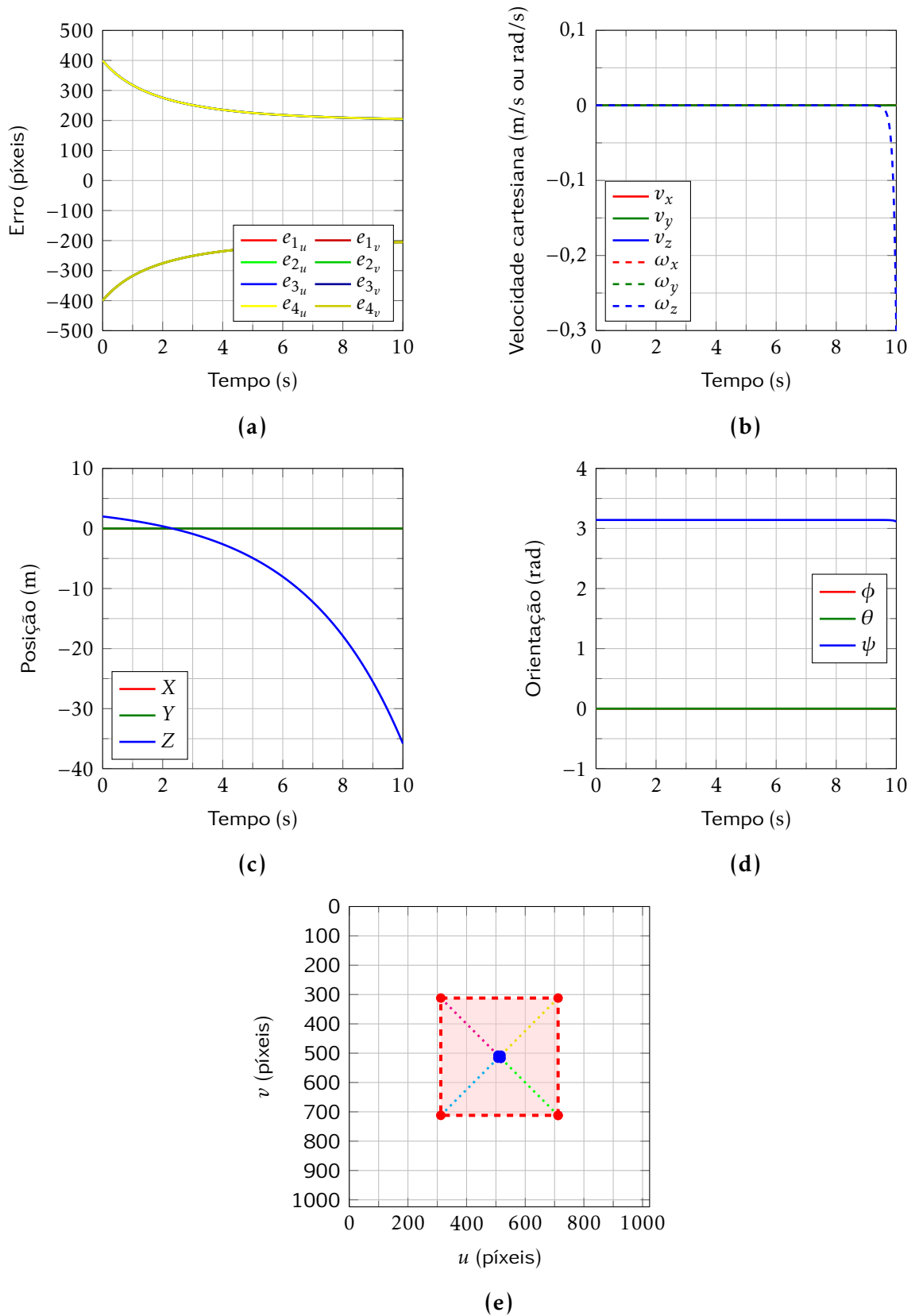
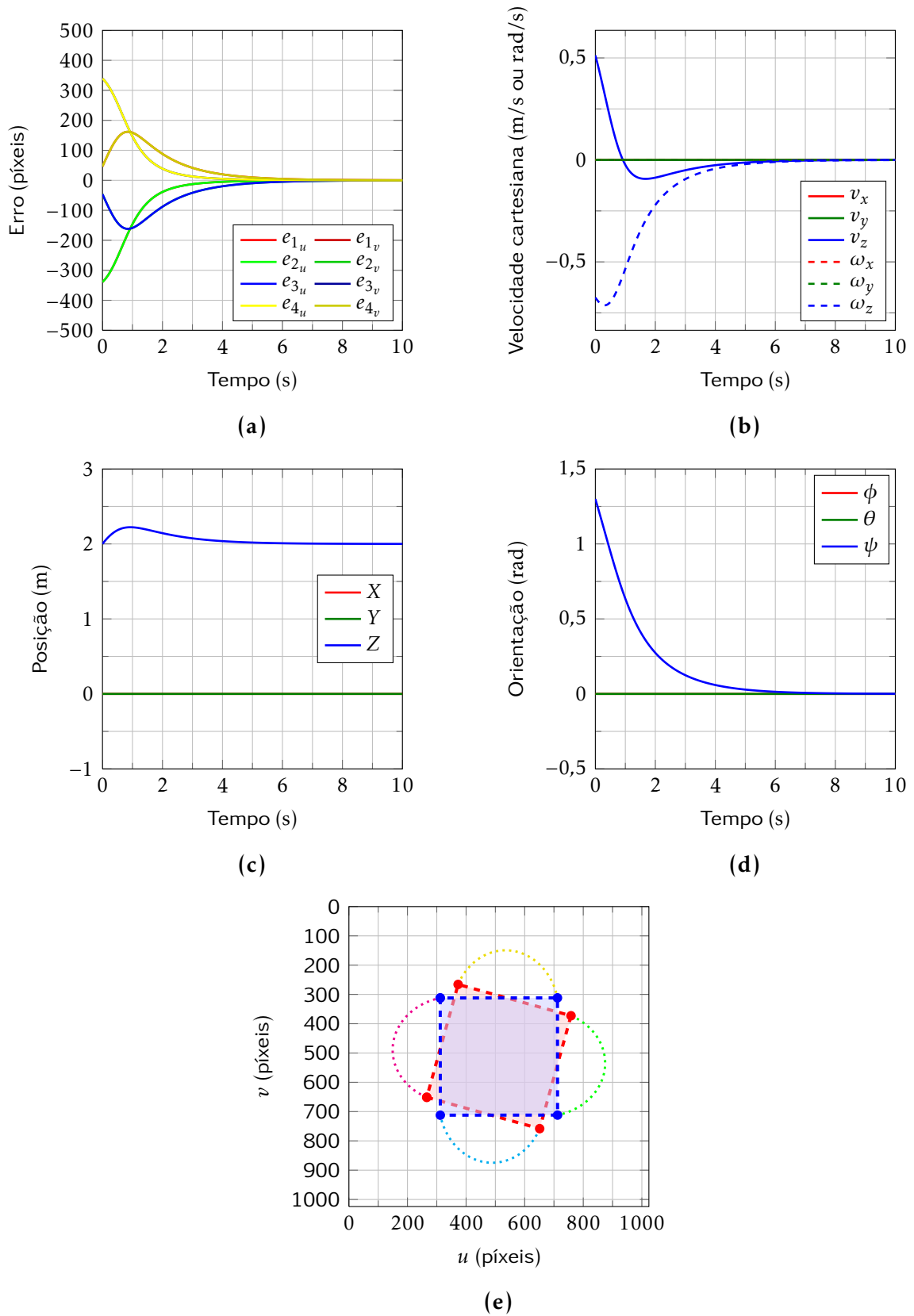


FIGURA 3.5 – Resultado de uma tarefa de IBVS de rotação pura de  $\pi$  radianos em relação ao eixo  $z$  para  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A tarefa falha.



**FIGURA 3.6** – Efeito de recuo da câmera devido a rotação pura de  $60^\circ$  em relação ao eixo  $z$  com  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d_o,c})$ . (e) projeção da cena na imagem; (b) velocidades cartesianas; (c) posição; (d) orientação da câmera. A câmera se aproxima do objeto-alvo na direção do eixo óptico.

Nesse caso,  $\mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d,o,c})$  não é singular (já que é constante durante toda a tarefa). Contudo, o problema ocorre devido à singularidade de  $\mathbf{L}_s^+(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c})$ , que está envolvida na condição de estabilidade (ver Equação 3.32).

Vale ressaltar que nos dois casos anteriormente descritos, apesar de produzirem movimentos subótimos, a tarefa convergirá — exceto para os casos extremos de rotações de  $\pi$  radianos. O movimento indesejado de translação é tão menos perceptível quanto menor for o erro  $\mathbf{e}$ . Uma possível alternativa é usar outro tipo de característica visual no cálculo da matriz de interação, como retas, por exemplo (CHAUMETTE, 1998).

Escolhendo  $\widehat{\mathbf{L}}_s^+ = \frac{1}{2}(\mathbf{L}_s(\mathbf{s}, \widehat{\mathbf{Z}}_{o,c}) + \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d,o,c}))^+$  é intuitivo notar que a média entre as duas estratégias anteriores produzirá o movimento representado pela setas pretas na Figura 3.3, mesmo para erros grandes. O movimento será sempre de rotações puras, exceto para o caso extremo. Malis (2004) traz mais detalhes sobre essa abordagem.

### 3.1.7.2 Atingindo mínimos locais

Uma postura corresponde a uma mínimo local quando  $\mathbf{v}_c = \mathbf{0}$  e  $\mathbf{s} \neq \mathbf{s}_d$  (ou seja, o erro  $\mathbf{e} = \mathbf{s} - \mathbf{s}_d$  não é nulo). Lembrando das Equações 3.20 ( $\mathbf{v}_c = \widehat{\mathbf{L}}_s^+ \dot{\mathbf{s}}$ ) e 3.24 ( $\mathbf{v}_c = -\lambda \widehat{\mathbf{L}}_s^+ \mathbf{e}$ ), se  $\mathbf{v}_c = \mathbf{0}$  e  $\mathbf{s} - \mathbf{s}_d \neq \mathbf{0}$ , isso equivale a

$$(\mathbf{s} - \mathbf{s}_d) \in \mathcal{N}(\widehat{\mathbf{L}}_s^+) \quad \text{ou} \quad \dot{\mathbf{s}} \in \mathcal{N}(\widehat{\mathbf{L}}_s^+). \quad (3.36)$$

Para o caso de três pontos,  $\mathbf{L}_s$  tem posto cheio igual a seis e, portanto,  $\mathcal{N}(\widehat{\mathbf{L}}_s^+) = \mathbf{0}$ , que significa que não há mínimos locais. Porém, como citado anteriormente, existem quatro posturas distintas que produzem os mesmos valores de  $\mathbf{s}$  (e, conseqüentemente,  $\mathbf{s} = \mathbf{s}_d$ ), ou seja, existem quatro mínimos globais. Teoricamente, para garantir um único mínimo

global, são necessários quatro pontos (CHAUMETTE, 1998). Nesse caso,  $\mathbf{L}_s \in \mathbb{R}^{8 \times 6}$  o que implica que  $\mathcal{N}(\widehat{\mathbf{L}}_s^+) = 2$ . Contudo, o fato de  $\mathcal{N}(\widehat{\mathbf{L}}_s^+) = 2$  não é suficiente para garantir a existência de mínimos locais, pois ainda é necessário que as posturas em que  $\mathbf{s} - \mathbf{s}_d$  (ou  $\mathbf{s}$ )  $\in \mathcal{N}(\widehat{\mathbf{L}}_s^+)$  sejam coerentes fisicamente, ou seja, que tal postura exista. A ocorrência de mínimos locais se deve à existência de movimentos da imagem que não são realizáveis.

### 3.1.8 O ganho proporcional constante

O ganho proporcional constante é a opção clássica apresentada na literatura do controle servo visual. Verifica-se tal fato pela quantidade de trabalhos teóricos e aplicados que usam essa abordagem. Como exemplo, é possível apontar os trabalhos citados neste texto — a maioria deles usa controladores proporcionais. A simplicidade desse controlador permite concentrar a pesquisa em aspectos próprios do controle baseado em imagem. Por outro lado, algumas pesquisas exploram controladores alternativos, como por exemplo: Silveira *et al.* (2002) com LQR (*linear quadratic regulator*), Danès e Bellot (2006) usaram controladores ótimos e Allibert, Courtial e Chaumette (2010) aplicaram controladores preditivos.

A estrutura clássica costuma apresentar problemas em condições que requerem grandes translações e rotações (CHAUMETTE, 1998). Allibert, Courtial e Touré (2008) mostraram esse fato na prática, comparando o desempenho da abordagem clássica com um controlador preditivo visual. Duas causas contribuem para esse comportamento: primeiro, há a possibilidade de atingir um mínimo local. Porém, como apontado anteriormente, experimentos práticos indicam que a região de estabilidade local do IBVS não é pequena; segundo, o ganho proporcional constante implica que grandes erros

causarão elevados sinais de controle. Tais sinais podem causar movimentos indesejados no espaço da tarefa e fazer com que as características visuais saiam do campo de visão da câmera, fazendo com que a tarefa falhe.

Para evitar os problemas descritos anteriormente, esse trabalho avalia duas alternativas: ganho variável e filtro *fuzzy*. Espera-se manter a simplicidade dos controladores servo visuais clássicos melhorando o desempenho do sistema em malha fechada. O Capítulo 4 trata deste tema com detalhes.

As expressões *grandes deslocamentos*, *grandes translações e rotações* e *grandes erros e sinais de controle* são relativas, isto é, grandes deslocamentos são, em geral, relativos à profundidade da cena. No entanto, não se encontrou nenhum trabalho ou método capaz de quantificar tais valores. Portanto, a verificação dos problemas causados pelo ganho proporcional constante no estudo de caso apresentado neste texto foi empírica.

## 3.2 Conclusão

Este capítulo detalhou a fundamentação teórica do controle servo visual baseado na imagem. Definiu-se matematicamente a matriz de interação e a deduziu analiticamente para o caso de primitivas geométricas pontuais. Dada a dificuldade de se obter valores reais da matriz para pontos devido à dependência do valor de  $Z_{o,c}$ , mostrou algumas alternativas para estimação de  $L_s^+$  descritas na literatura. A literatura mostra que a estabilidade global assintótica do IBVS para pontos não é garantida, mas que, no entanto, é possível garantir a estabilidade em uma vizinhança da postura desejada. Evidenciou-se a existência de potenciais problemas descritos na literatura relacionados à singularidades da tarefa e posturas que levam a mínimos locais, assim como alternativa



para solucioná-los. Evidenciou-se que o IBVS usando  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{\mathbf{Z}}_{d_o,c})$  é problemático para movimentos de rotação em relação ao eixo ótico.

O Capítulo 4 descreve os conceitos básicos da modelagem de veículos multirrotores e particularidades da aplicação da servovisão 2D nesse tipo de sistema. Detalha as alternativas sugeridas para melhorar o desempenho da abordagem clássica: o ganho variável e o filtro *fuzzy* e apresenta justificativas para as escolhas feitas nesse trabalho.

## 4 Modelagem e alternativas de controle

Após uma descrição sucinta do modelo matemático de um veículo quadrirrotor, este capítulo descreve particularidades da aplicação do IBVS a este tipo de VANT. Para evitar alguns dos problemas do IBVS clássico citados anteriormente, são apresentadas duas técnicas existentes na literatura com o objetivo de melhorar o desempenho do IBVS clássico.

### 4.1 Nomenclatura de veículos aéreos com quatro rotores

Parece não haver um consenso em relação à denominação adequada para veículos aéreos de quatro rotores. É comum encontrar na literatura, tanto em português quanto em inglês, os termos quadrirrotor (*quadrotor*) e quadricóptero (*quadcopter*). A palavra *helicóptero* vem do grego *hélix* (espiral) e *pterón* (asa) — como *pterodáctilo*, do grego *dáktylos* (dedo), "dedo alado". Portanto, o termo *quadrirrotor*, do latim *quadri* (quatro), e *rotor* (contração do inglês *rotator*, algo que gira sobre o próprio eixo), parece mais correto, já que este tipo de veículo não possui asas. A expressão *quadricóptero* também é indevida por unir um prefixo latino com um sufixo grego. Já os termos tetrarotor, do grego *tetra* (quatro), e tetracóptero (*tetracopter*), não são comuns (MAHONY; KUMAR; CORKE, 2012).

## 4.2 Modelagem matemática de quadrrrotores

Quadrrrotores são veículos aéreos compostos de quatro rotores independentes fixos a uma estrutura rígida formada por duas hastes em forma de cruz. Como são sistemas subatuados, os graus de liberdade não atuados precisam ser controlados através da dinâmica do sistema. A Figura 4.1 apresenta a notação usada neste texto, o par de rotores ímpares (azuis) giram em sentido anti-horário, enquanto os pares (vermelhos) giram no sentido oposto; em ciano estão indicados o sistema de coordenadas fixo e os respectivos ângulos de rolagem ( $\phi$ ), arfagem ( $\theta$ ) e guinada ( $\psi$ ); o asterisco verde no rotor 1 indica a frente do veículo e o ciano no rotor 2, o lado direito. Todos os ângulos do quadrrrotor são expressos em relação ao sistema de coordenadas do veículo, como descrito no Apêndice A.

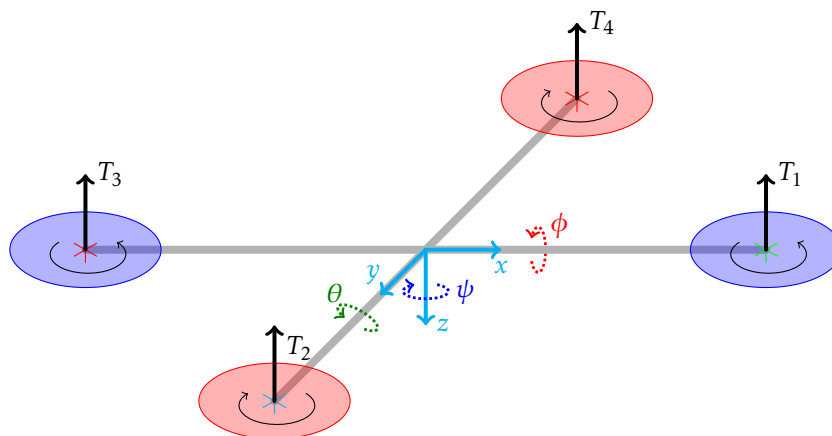


FIGURA 4.1 – Notação usada para o quadrrrotor neste texto.

Rotores ímpares giram em sentido oposto aos pares. No exemplo da Figura 4.2, os rotores 1 e 3 giram no sentido anti-horário, enquanto 2 e 4 no sentido horário; setas médias representam o torque suficiente para que cada rotor gere o empuxo necessário para igualar a força da gravidade (manter o sistema parado no ar), setas grossas representam torques maiores que os médios e setas finas, torques menores. O

sistema é controlado gerando-se diferentes empuxos, representados por  $T_i, i = 1, \dots, 4$ , em seus rotores:

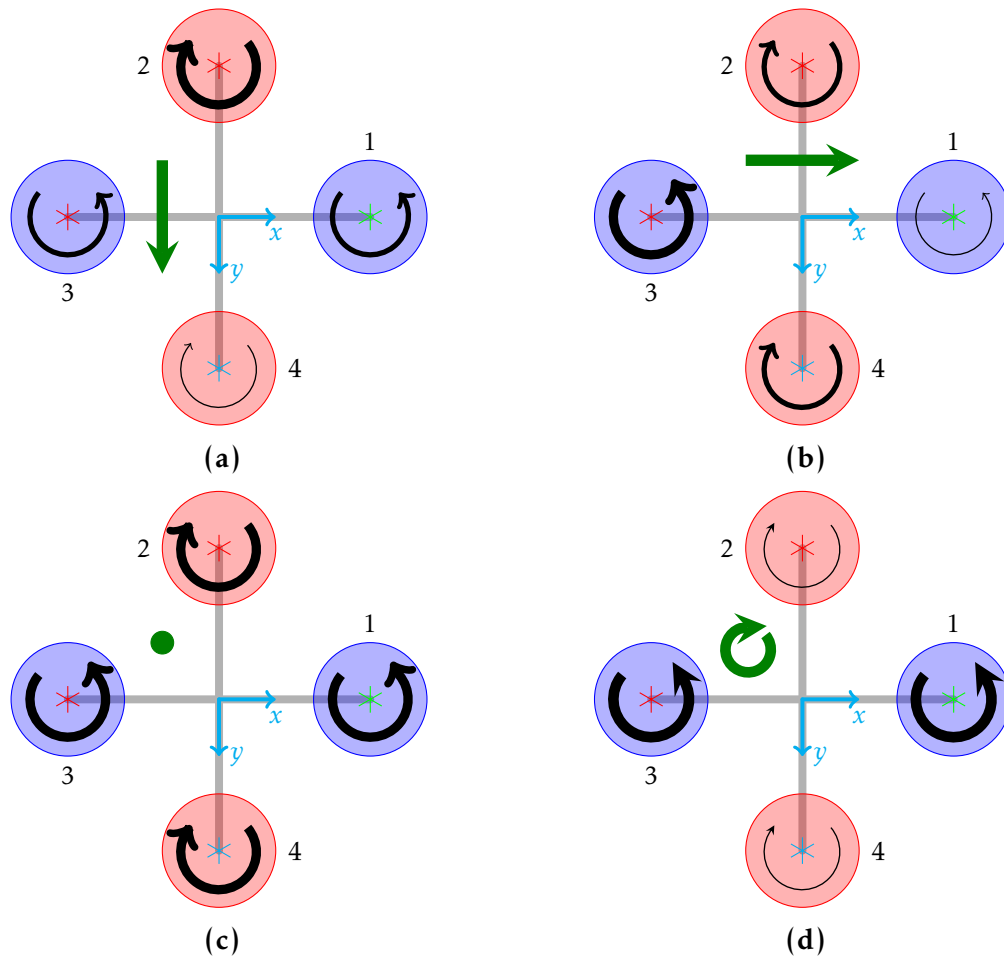


FIGURA 4.2 – Configurações possíveis para os rotores.

**Translação transversal** Translação no eixo  $y$  é realizada controlando o ângulo de rolagem (*roll*), representado por  $\phi$ . Para transladar no sentido positivo, por exemplo, faz-se  $T_4 > T_2$  e  $T_1 = T_3$ , como mostra a Figura 4.2a;

**Translação longitudinal** É obtida alterando o ângulo de arfagem (*pitch*),  $\theta$ . Para o sentido positivo, faz-se  $T_3 > T_1$  e  $T_2 = T_4$  (Figura 4.2b);

**Translação vertical** Realizada através da mudança do empuxo total gerado por todos os rotores simultaneamente. Caso o empuxo resultante seja maior do que a força

da gravidade, o veículo sobe, caso contrário, desce (Figura 4.2c);

**Rotação em relação ao eixo vertical** O giro de cada rotor causa um torque no sentido contrário na estrutura do quadricóptero. Como cada par de rotores gira em sentidos opostos, esse torque gerado na estrutura também tem sentidos contrários. Assim, para controlar o ângulo de guinada (*yaw*),  $\psi$ , basta fazer com que o torque gerado por um par seja maior do que o torque gerado pelo outro par, mantendo, no entanto, a resultante total igual à força da gravidade. Por exemplo, fazendo  $T_2 = T_4 > T_1 = T_3$ , como exposto na Figura 4.2d.

A velocidade angular do rotor  $i$  é  $\Omega_i$  e o empuxo é um vetor perpendicular no sentido negativo de  $z$ , dado por

$$T_i = b\Omega_i^2, \quad i = 1, 2, 3, 4 \quad (4.1)$$

em que  $b > 0$  é o coeficiente de empuxo e depende da densidade do ar, das dimensões, formato e quantidade de hélices. Aplicando a segunda lei de Newton,

$$m\dot{v} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} - {}^aR_q \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}, \quad (4.2)$$

onde  $v$  é a velocidade do veículo em relação a um sistema de coordenadas fixo no ambiente,  $g$  é a aceleração da gravidade,  $m$  é a massa total do veículo,  $T = \sum T_i$  é o empuxo resultante e  ${}^aR_q$  é uma matriz de rotação do sistema de coordenadas do quadricóptero para um sistema de coordenadas fixo no ambiente.

O desbalanceamento dos empuxos gerados pelos pares de rotores causam torques resultantes não nulos na estrutura do VANT, fazendo com que ele gire. O torque de

rolagem, em torno do eixo  $x$ , é dado por

$$\tau_x = L(T_4 - T_2), \quad (4.3)$$

onde  $L$  é a distância do rotor ao centro de massa do quadricóptero, localizado na origem do sistema de coordenadas do veículo. Substituindo (4.1),

$$\tau_x = bL(\Omega_4^2 - \Omega_2^2). \quad (4.4)$$

Analogamente, o torque de arfagem (em torno do eixo  $y$ ) é

$$\tau_y = bL(\Omega_4^2 - \Omega_2^2). \quad (4.5)$$

Os torques aplicados a cada hélice pelos rotores são opostos pelo arrasto aerodinâmico

$$Q_i = d\Omega_i^2, \quad i = 1, 2, 3, 4, \quad (4.6)$$

sendo  $d$  o coeficiente de arrasto dependente dos mesmos parâmetros de  $b$ . O arrasto aerodinâmico gera um torque de reação resultante em relação ao eixo  $z$  (torque de guinada) dado por

$$\tau_z = Q_1 - Q_2 + Q_3 - Q_4 \quad (4.7)$$

$$= d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2). \quad (4.8)$$

Escrevendo na forma matricial,

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b \\ 0 & -bL & 0 & bL \\ bL & 0 & -bL & 0 \\ d & -d & d & -d \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \Gamma \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (4.9)$$

Se  $b, L, d > 0$  a matriz  $\Gamma$  tem posto cheio e pode ser invertida. Nesse caso, faz-se

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (4.10)$$

para encontrar o empuxo  $T$  e torques  $\tau_x, \tau_y, \tau_z$  desejados.

Mais detalhes sobre a modelagem matemática e técnicas de controle de quadrrrotores podem ser encontradas em Mahony, Kumar e Corke (2012). Corke (2011) fornece alguns exemplos didáticos para simulação deste tipo de veículo. Santana e Borges (2009) utilizam o formalismo de Euler-Lagrange para determinar as equações diferenciais para os ângulos de rotação.

### 4.3 Particularidades da aplicação do IBVS em veículos quadrrrotores

Os controladores servo visuais implementados neste trabalho fornecem sinais de referência para um outro controlador do nível dos atuadores. De acordo com a classificação criada por Weiss, Sanderson e Neuman (1987) esta seria a abordagem *ver e mover dinâmico*. Para aplicar esse tipo de controle em um quadrrrotor é preciso considerar

algumas das propriedades dinâmicas do veículo: a necessidade de se alterar os ângulos de rolagem e arfagem para poderem transladar e a orientação do movimento em relação ao eixo  $z$ , que altera o ângulo de guinada.

### 4.3.1 Deslocamento das características visuais em decorrência da rolagem e arfagem

Quadrirrotores são sistemas subatuados e precisam rotacionar para transladar. Porém, sempre que o veículo gira em um determinado sentido para que possa transladar, as projeções das características visuais em uma câmera que esteja fixa na estrutura do veículo giram no sentido contrário. A Figura 4.3 mostra esse efeito: 4.3a e 4.3b mostram uma vista 3D do quadrirrotor e projeção das características visuais na imagem para um ângulo de rolagem nulo, respectivamente; 4.3c e 4.3d mostram a vista 3D do quadrirrotor e a projeção das características para um ângulo de rolagem de  $60^\circ$ . Observa-se que o sentido da rotação do plano da imagem é oposto ao do quadrirrotor.

Algumas estratégias foram criadas para tratar tal problema. Ceren e Altuğ (2011) usaram uma câmera fixa no ambiente (*eye-to-hand*) e limitaram seus experimentos a translações e rotações no eixo  $z$ . Guenard, Hamel e Mahony (2008) incorporaram o modelo dinâmico do veículo no controlador visual. Bourquardez *et al.* (2009) limitaram o movimento do sistema para translações puras, além de usar o modelo de projeção esférica e primitivas geométricas não pontuais. Mahony, Kumar e Corke (2012) sugerem corrigir a projeção na imagem usando a matriz de interação para calcular o deslocamento causado pelo rolamento e arfagem. Esse valor é, então, subtraído do valor real da projeção. Primeiro, particiona-se (3.14) separando os termos referentes às rotações em



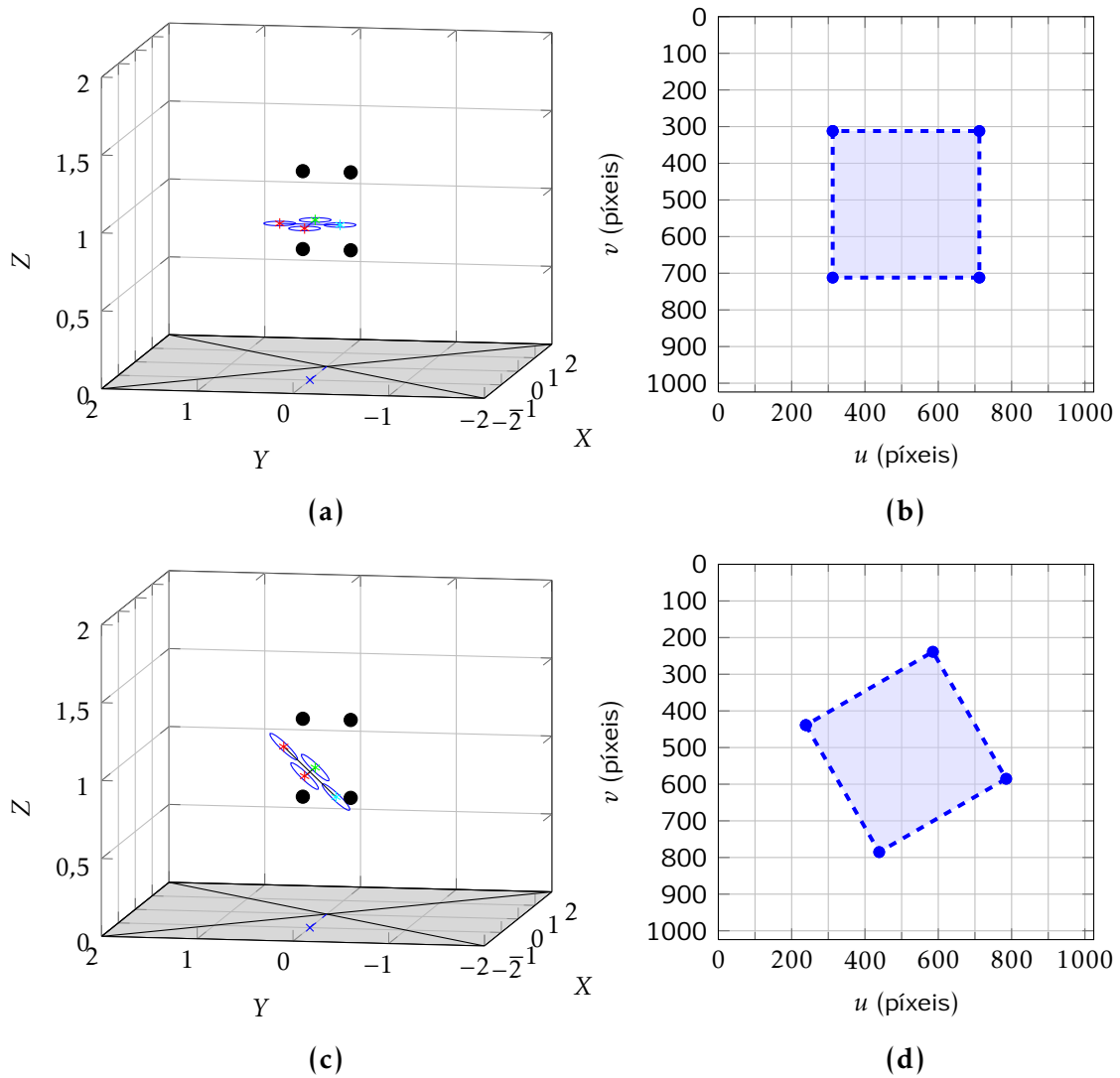


FIGURA 4.3 – Simulação de efeito do ângulo de rolagem sobre a projeção das características visuais na imagem.

relação a  $x$  e  $y$ ,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \mathbf{L}_1(\bar{u}, \bar{v}) \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_z \end{bmatrix} + \mathbf{L}_2(\bar{u}, \bar{v}) \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}, \quad (4.11)$$

onde  $\mathbf{L}_1$  são as colunas da matriz de interação correspondentes a translação em  $x, y, z$  e rotação em  $z$ , enquanto  $\mathbf{L}_2$  àquelas correspondentes a rotação em  $x$  e  $y$ . Rearranjando,

$$\begin{bmatrix} \dot{u}' \\ \dot{v}' \end{bmatrix} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} - \mathbf{L}_2(\bar{u}, \bar{v}) \begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix}, \quad (4.12)$$

onde  $\dot{u}'$  e  $\dot{v}'$  são os valores da projeção após a remoção da influência dos movimentos de rolagem e arfagem.

Esse método requer que valores de  $\omega_x$  e  $\omega_y$  estejam disponíveis e sejam precisos. Corke (2011) propõe um cálculo aproximado da influência dos ângulos de rolagem e arfagem na projeção das características visuais para sistemas que usam o modelo de projeção esférica.

## 4.4 Melhorando o desempenho do IBVS clássico

O Capítulo 3 mostrou que o controle servo visual clássico é susceptível a problemas quando a postura inicial é distante da desejada. Os grandes erros na imagem decorrentes dessa distância implicam sinais de controle iniciais proporcionalmente elevados.

O trabalho aqui descrito baseia-se em duas alternativas para minimizar as condições citadas: o uso de ganhos proporcionais variáveis e filtros *fuzzy* que ajustam os sinais de erro. A ideia central das duas abordagens é reduzir os grandes sinais de controle (geralmente no início da tarefa) e, proporcionalmente, aumentar sinais pequenos quando se está próximo da condição desejada. O filtro *fuzzy* também reduz a variação entre as amplitudes dos sinais de controle entre cada iteração (SEQUEIRA GONÇALVES *et al.*, 2004).

### 4.4.1 Ganho variável

Esse método de cálculo de ganho variável foi proposto inicialmente por (BENSALAH, 1996) para sistemas de visão ativa<sup>1</sup>. Usa-se uma função exponencial que decresce à medida que o erro aumenta. Obtém-se, portanto, um ganho em função do erro — erros maiores têm ganhos menores. O valor desse novo ganho, denominado  $\lambda_e$ , é dado por

$$\lambda_e(\mathbf{e}) = a e^{(-b|\mathbf{e}|)} + c, \quad (4.13)$$

sendo  $a, b, c > 0$  parâmetros que precisam ser determinados. Seus valores são dados por

$$a = \lambda(0) - \lambda(\infty),$$

$$b = \lambda'(0)/a,$$

$$c = \lambda(\infty),$$

onde  $\lambda(0)$ ,  $\lambda(\infty)$  e  $\lambda'(0)$  representam, respectivamente, o ganho quando o erro é nulo, quando o erro tende ao infinito e a inclinação da curva quando o erro é nulo. A Figura 4.4 mostra curvas dessa função para diferentes valores de  $\lambda(0)$ ,  $\lambda(\infty)$  e  $\lambda'(0)$ .

Substituindo (4.13) em (3.24), obtém-se a lei de controle com ganho proporcional variável

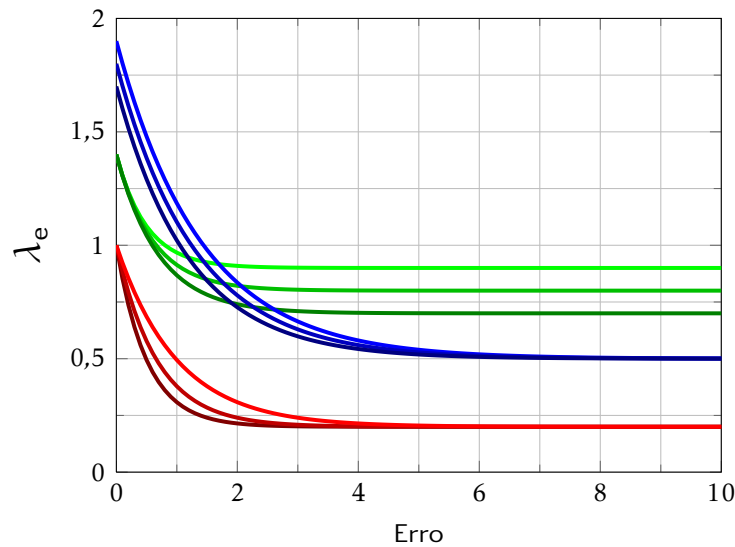
$$\mathbf{v}_{c_{variável}} = -\lambda_e \widehat{\mathbf{L}}_s^+ \mathbf{e}. \quad (4.14)$$

### 4.4.2 Filtro *fuzzy*

Filtros *fuzzy* reduzem o próprio erro, mantendo o ganho do controlador constante.

Essa abordagem foi proposta inicialmente para resolver problemas de discretização de

<sup>1</sup>Sistemas de visão ativa manipulam os parâmetros da câmera para melhor investigar o ambiente e obter informações dele.



**FIGURA 4.4** – Diferentes curvas da função de ganho variável. Influência da variação em um dos parâmetros, mantendo os outros dois constantes: linhas azuis, variando-se  $\lambda(0)$ ; verdes,  $\lambda(\infty)$ ; vermelhas,  $\lambda'(0)$ .

sinais de controle em controladores preditivos (DE COSTA SOUSA; SETNES, 1999). Sequeira Gonçalves *et al.* (2004) utilizaram essa técnica em servovisão, contudo, aplicaram apenas em braços manipuladores com dois graus de liberdade. Não se encontrou nenhum trabalho que aplicasse tais filtros em robôs móveis ou outros sistemas com dinâmica mais rápida, como é o caso dos quadrrrotores.

Define-se um novo erro, chamado erro filtrado e representado por  $\mathbf{e}_f$ , dado por

$$\mathbf{e}_f = \gamma_f \mathbf{e}, \quad (4.15)$$

onde  $\gamma_f \in [0, 1]$  é um fator de ponderação *fuzzy* obtido a partir de uma função de pertinência como, por exemplo, a triangular (Figura 4.5).

Define-se o suporte da função de pertinência a partir dos erros mínimo e máximo. Sequeira Gonçalves *et al.* (2004) sugerem que se use o valor do erro no instante inicial,  $\mathbf{e}(0)$ , acrescido de um valor pequeno  $\epsilon > 0$  para garantir  $\gamma_f(0) \neq 0$ . Portanto, o valor de

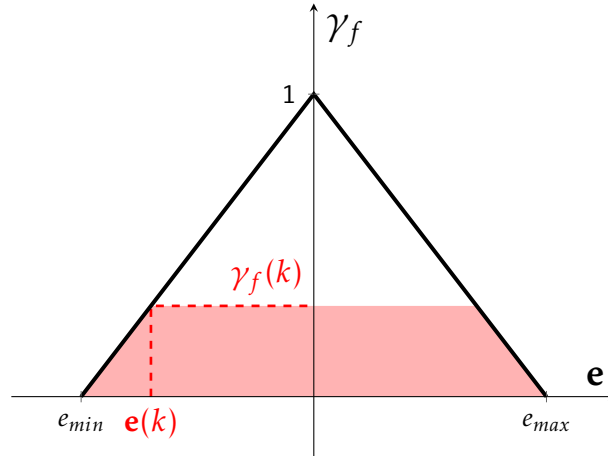


FIGURA 4.5 – Função de pertinência triangular. Para cada valor do erro  $e(k)$  no instante  $k$ , o respectivo fator de ponderação  $\gamma_f(k)$  é dado por uma função de pertinência triangular.

$e_{min}$  é simétrico a  $e_{max}$

$$\begin{cases} e_{min} = -|e(0)| - \epsilon, \\ e_{max} = |e(0)| + \epsilon. \end{cases} \quad (4.16)$$

Por ser uma função de pertinência triangular, define-se  $\gamma_f$  como

$$\gamma_f(e) = -\frac{1}{|e_{max}|}|e| + 1 \quad (4.17)$$

e encontra-se a nova lei de controle substituindo o fator de ponderação *fuzzy* definido na Equação 4.15 em 3.24,

$$\mathbf{v}_{c_{fuzzy}} = -\lambda \gamma_f \widehat{\mathbf{L}}_s^+ \mathbf{e}. \quad (4.18)$$

Uma das vantagens do filtro *fuzzy* em relação à adaptativa é a simplicidade: possui apenas um parâmetro para ser sintonizado, o  $e_{max}$ . Esse único parâmetro ainda pode ser obtido automaticamente no início da tarefa. Por outro lado, Sequeira Gonçalves *et al.* (2004) mostram que com essa abordagem o decaimento do erro não é puramente exponencial. As trajetórias das características visuais já não são mais retas para o caso

ideal e agora dependem da função de pertinência usada.

A Seção 5.3.2 mostra exemplos práticos simulados para os três tipos de controladores usados. Nesses exemplos, como não se considera a dinâmica do sistema, é mais fácil perceber a influência de cada controlador no comportamento em malha fechada.

### 4.4.3 Considerações

Este capítulo apresentou a nomenclatura e convenções de quadrrrotores usadas neste texto e introduziu a teoria de modelagem deste veículo. A seguir, abordou particularidades da aplicação do IBVS em veículos quadrrrotores e avaliou duas abordagens alternativas para reduzir as magnitudes dos sinais de controle da servovisão 2D clássica: o ganho variável e o filtro *fuzzy*.

O Capítulo 5 faz uma comparação dos três controladores com simulações aplicadas em três sistemas: um sistema cinemático completamente atuado com seis graus de liberdade, um modelo simulado que considera as propriedades dinâmicas de quadrrrotores e em uma plataforma experimental real.

# 5 Resultados experimentais

Este capítulo mostra uma comparação de desempenho entre o IBVS com ganho proporcional, ganho variável e filtro *fuzzy* com experimentos simulados em um sistema cinemático completamente atuado com seis graus de liberdade. Em seguida, os três controladores são validados em quadricópteros. Tal validação é feita em duas etapas: simulação com um modelo dinâmico de um quadricóptero e aplicação em uma plataforma experimental — um VANT real disponível comercialmente.

## 5.1 Considerações sobre a aplicação

O sistema de controle servo visual usado na validação deve manter as seguintes características:

1. montagem da câmera *eye-in-hand*;
2. modelo de projeção perspectiva;
3. características visuais pontuais;
4. controlador proporcional;
5. controlar todos os graus de liberdade do veículo;

6. não usar informações de outros sensores na malha de controle visual que não a câmera;
7. formulação cinemática do controlador servo visual;
8. objeto-alvo rígido e imóvel em relação a um sistema de coordenadas fixo no ambiente.

Tais especificações preservam a simplicidade do controlador. A abordagem cinemática oferece algumas vantagens. Primeiro, apesar da câmera estar embarcada no veículo, o processamento da imagem e cálculo dos sinais de controle são feitos em um computador remoto. Para isso é necessário que a imagem seja enviada do quadricóptero para um computador remoto através de uma conexão sem fio. Tal computador executa o processamento da imagem e cálculo do sinal de controle e os envia de volta para o VANT. Esse tráfego de dados implica em taxas de operação do sistema visual relativamente baixas. Para manter o quadricóptero estável enquanto voa, o veículo precisa de uma malha de controle de baixo nível que opere com altos ganhos a altas taxas. Essa malha de baixo nível usa informações de sensores inerciais, como giroscópios e acelerômetros recebendo sinais de referências da malha externa (alto nível, controlador servo visual). Segundo, controladores servo visuais cinemáticos são reutilizáveis, já que não dependem das propriedades dinâmicas do sistemas que estão operando. Abordagens semelhantes foram utilizadas em outros trabalhos (SARIPALLI; MONTGOMERY; SUKHATME, 2003; BOURQUARDEZ *et al.*, 2009; CEREN; ALTUĞ, 2011).

A opção por não usar informações de outros sensores além da câmera é justificada por dois motivos: manter a estrutura do controlador exclusivamente baseada em informações visuais, como na formulação clássica do IBVS e evitar que informações de



instrumentos de baixa precisão ou ruidosas interfiram no desempenho do controlador.

### 5.1.1 Movimentos do quadricóptero em relação ao eixo ótico

Os trabalhos com servovisão de quadricópteros citados nesta dissertação utilizam características visuais posicionadas no chão e câmeras apontadas verticalmente para baixo. Nessa configuração, o movimento de guinada é em torno do eixo ótico. Como mostrado na Seção 3.1.7.1, isso pode acarretar movimentos indesejados de afastamento e aproximação da câmera em relação ao objeto-alvo (translação no eixo  $z$ ). Este trabalho usou uma estratégia diferente usando uma câmera frontal com eixo ótico paralelo ao eixo  $x$  e o objeto-alvo posicionado verticalmente (o plano em que as características visuais estão localizadas é ortogonal ao plano do solo).

## 5.2 Plataforma experimental

Os aparelhos usados como plataforma experimental podem ser divididos em dois grupos: o veículo e o computador remoto. Cada um deles são descritos a seguir.

### 5.2.1 O quadricóptero

O VANT usado para validação dos controladores propostos foi o AR.Drone (Figura 5.1), disponível comercialmente e fabricado pela empresa francesa Parrot. Esse veículo tem sido usado em algumas universidades para pesquisa em robótica móvel, controle e navegação devido ao baixo custo e quantidade de instrumentos disponíveis (KRAJNÍK *et al.*, 2011). Segundo os próprios engenheiros que desenvolveram o projeto (BRISTEAU

*et al.*, 2011): “[...] o projeto é um exemplo de uso sofisticado de sensores de baixo custo (MEMS e câmeras) para mercados de massa em que o preço de venda é da maior importância”.

O AR.Drone (versão 1.0) pesa 420g e é equipado com quatro motores sem escovas presos a hastes de fibra de carbono e um corpo de plástico. Dispõe de duas câmeras, uma voltada para a frente e outra apontada verticalmente para baixo. Possui dois tipos de coberturas plásticas distintas, uma para ser usada em ambientes externos (apenas uma capa para o compartimento da bateria) e outra para ambientes internos, que inclui, ainda, proteções de poliestireno expandido para as hélices. É alimentado por uma bateria capaz de fornecer energia para o sistema em voo por aproximadamente doze minutos.



**FIGURA 5.1** – Quadrirotor AR.Drone, versão 1.0, desenvolvido pela Parrot. (a) com capa para ambientes internos; (b) com capa para ambientes externos.

Esse veículo possui duas placas, uma de controle e outra de sensores. A placa de controle é composta por um processador ARM9 operando a 486 MHz com 128 MB de memória DDR RAM a 200 MHz. Esse processador roda um sistema operacional baseado em Linux e é responsável por todos os cálculos, comunicação e controle. A placa de controle adquire dados dos sensores e envia comandos para os rotores a 200 Hz. O usuário (ou o computador remoto) pode especificar a rolagem, arfagem, guinada e

velocidade vertical. No entanto, esses comandos são recebidos pela placa de controle a uma taxa de 30Hz. A Figura 5.2 mostra um diagrama de blocos com estrutura interna do quadricóptero e setas que indicam o sentido do fluxo de informações. O AR.Drone possui um *chip* para redes sem fio responsável por estabelecer a conexão com o computador remoto

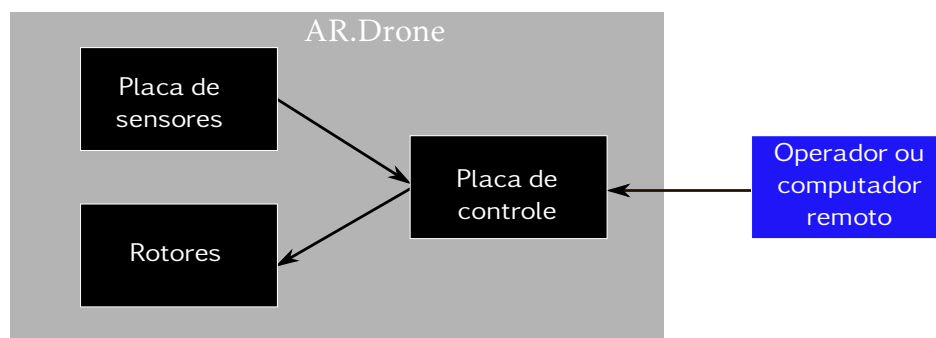


FIGURA 5.2 – Diagrama da estrutura interna comunicação do AR.Drone.

A placa de sensores usa um microcontrolador PIC operando a 40 MHz como interface para os sensores: uma acelerômetro de três eixos, um giroscópio de dois eixos, um giroscópio vertical de um eixo e dois sensores ultrassônicos. Os sensores ultrassônicos servem para estimar a altitude do aparelho. O acelerômetro e os giroscópios constituem uma unidade de medidas inerciais (IMU) de baixo custo (aproximadamente 10 USD em 2010) (BRISTEAU *et al.*, 2011).

A câmera frontal tem lente grande angular com campo de visão em torno de 75° por 60° e resolução de 320 por 240 píxeis. A vertical, que também é usada para estimar a velocidade em relação ao solo, tem lente com campo de visão de aproximadamente 45° por 35° e resolução de 176 por 144 píxeis. Ambas transmitem a uma taxa de 15 quadros por segundo (PISKORSKI *et al.*, 2012). As duas câmeras fornecem imagens ruidosas e com distorções visíveis. A Figura 5.3 mostra uma imagem capturada pela

câmera frontal do veículo.



FIGURA 5.3 – Imagem capturada pela câmera frontal do AR.Drone.

O fornecedor do equipamento disponibiliza uma interface de programação de aplicativos (API) usando a linguagem de programação C++. É possível desenvolver programas que usam informações dos sensores, incluindo câmeras, e enviam sinais de controle. Contudo, para a realização destes experimentos, utilizou-se uma versão compacta dessa API (KRAJNÍK *et al.*, 2011), também em C++. Essa versão compacta fornece os seguintes dados sobre o AR.Drone: ângulos  $\phi$ ,  $\theta$  e  $\psi$ , velocidades  $v_x$  e  $v_y$  e altitude — a velocidade  $v_z$  não está disponível. A velocidade nominal máxima de translação é de 5 m/s.

Todos os experimentos foram feitos usando a cobertura para ambientes internos, mostrada na Figura 5.1a. Mais detalhes sobre o *hardware* do AR.Drone podem ser encontrado em Bristeau *et al.* (2011) e Piskorski *et al.* (2012).

### 5.2.1.1 Alguns problemas do AR.Drone e da API

Os sinais de controle do IBVS são velocidades lineares e angulares. Porém, o AR.Drone é controlado em função dos ângulos de rolagem, arfagem e guinada. Valores muito baixos dos sinais de controle não alteram a postura do veículo, fazendo com que surja uma faixa morta de comandos. O veículo apresenta uma translação indesejada paralela ao solo (*drift*) quando os sinais de controle de rolagem e arfagem são pequenos ou nulos. O fabricante relata no manual de desenvolvimento (PISKORSKI *et al.*, 2012) que "com os valores de rolagem e arfagem configurados para zero, o veículo permanecerá na posição horizontal porém continuará deslizando-se no ar devido a sua inércia. Apenas a

*resistência do ar o fará parar."*

Verifica-se visualmente que o AR.Drone é sensível a perturbações do ambiente. O vento causado pelo reflexo no chão, teto e paredes do ar movimentado por suas hélices é capaz de deslocá-lo.

A API não fornece nenhuma informação de posição espacial. Tal limitação impede garantir repetibilidade de condições iniciais e finais e análise do erro na posição espacial.

### **5.2.1.2 Calibração da câmera frontal**

O fornecedor do AR.Drone não disponibiliza os parâmetros intrínsecos das câmeras embarcadas. Portanto, realizou-se o procedimento de calibração para determinar tais parâmetros usando a ferramenta *calibrate2dGrid* da biblioteca ViSP (MARCHAND; SPINDLER; CHAUMETTE, 2005). Como o IBVS é robusto a erros de calibração, o critério usado não foi rigoroso. O valores encontrados estão descritos na Tabela 5.1.

**TABELA 5.1** – Parâmetros intrínsecos da câmera frontal do AR.Drone identificados por calibração.

Parâmetro	Valor	Descrição
$u_0$	154,331 30	Coordenada horizontal em píxeis do ponto principal
$v_0$	117,106 55	Coordenada vertical em píxeis do ponto principal
$p_x$	208,019 27	Relação entre distância focal e dimensão horizontal do píxel
$p_y$	207,533 71	Relação entre distância focal e dimensão vertical do píxel
$k_{ud}$	0,020 23	Parâmetro de distorção radial de conversão $xy-uv$
$k_{du}$	-0,020 33	Parâmetro de distorção radial de conversão $uv-xy$

## 5.2.2 O computador remoto

O computador responsável por receber e processar a imagem enviada pelo AR.Drone, calcular os sinais de controle e enviá-lo de volta para o VANT é um Apple Macbook Pro (15-inch, early 2011). Possui processador Intel Core i7 de quatro núcleos operando a 2,2 GHz e 8 GB de memória DDR3 RAM. O sistema operacional é Ubuntu 12.04 LTS (Precise Pangolin) 64 bit.

A detecção e rastreamento das características visuais, assim como o cálculo de suas posições na imagem foi feito usando a biblioteca ViSP, também desenvolvida em C++. Todos os algoritmos de controle foram implementados usando C++.

As simulações foram feitas usando o MathWorks MATLAB R2010b (7.11.0.584) 64 bit. Para simular câmeras, projeção perspectiva, transformações homogêneas e o

modelo dinâmico do quadricóptero, usaram-se funções disponíveis nos *toolboxes Robotics Toolbox* e *Machine Vision Toolbox* (CORKE, 2007).

## 5.3 Dados simulados

### 5.3.1 Descrição da tarefa

O objetivo da tarefa é regular a postura do quadricóptero de maneira que seja necessário usar todos os quatro sinais de velocidade disponíveis para esse tipo de equipamento: translação nos três eixos e rotação em relação ao eixo  $z$ . Uma alternativa para forçar o cumprimento desse requisito é transladar e rotacionar a postura inicial do veículo em relação à postura desejada. Também se deseja que o sistema realize movimentos grandes o suficiente para que alguns dos problemas conhecidos do IBVS sejam evidenciados. Na prática, escolheram-se posturas que exigissem o deslocamento das características visuais na maior parte do plano da imagem. Todas as posturas estão expressas em relação a um sistema de coordenadas fixo no ambiente. As rotações seguem o padrão de rolagem-arfagem-guinada (como descrito no Apêndice B) em relação ao sistema de coordenadas do veículo mostrado na Figura 4.1.

Em todas as simulações, a matriz de interação usada foi  $\widehat{\mathbf{L}}_s^+ = \mathbf{L}_s^+(\mathbf{s}_d, \widehat{Z}_{d_{o,c}})$ . Dois fatos motivaram essa escolha. Primeiro, a imagem fornecida pelas câmeras do AR.Drone é ruidosa e com algumas distorções visíveis, o que implicaria em erros na estimação da profundidade  $Z_{o,c}$ , necessário para as demais alternativas para estimação da matriz de interação mencionados na Seção 3.1.5. Segundo, o uso de  $\mathbf{L}_s^+(\mathbf{s}_d, \widehat{Z}_{d_{o,c}})$  evita que se atinjam mínimos locais. Tal escolha, em contrapartida, implica em trajetórias não míni-

mas das características visuais na imagem, especialmente para grandes deslocamentos (como é o caso dessa aplicação). Tomou-se especial cuidado ao sintonizar os controladores e na escolha das condições inicial e desejada para evitar que, por esse motivo, as características saíssem da imagem. A matriz de interação é, portanto, constante durante toda a tarefa e igual a

$$\widehat{\mathbf{L}}_s^+ = \begin{bmatrix} -400 & 0 & 125 & -19.5312 & -219.5312 & -62.5 \\ 0 & -400 & -125 & 219.5312 & 19.5312 & -62.5 \\ -400 & 0 & -125 & 19.5312 & -219.5312 & -62.5 \\ 0 & -400 & -125 & 219.5312 & -19.5312 & 62.5 \\ -400 & 0 & -125 & -19.5313 & -219.5312 & 62.5 \\ 0 & -400 & 125 & 219.5312 & 19.5313 & 62.5 \\ -400 & 0 & 125 & 19.5313 & -219.5312 & 62.5 \\ 0 & -400 & 125 & 219.5312 & -19.5313 & -62.5 \end{bmatrix}.$$

Configurou-se a câmera simulada semelhante à encontrada na plataforma experimental usada. O único parâmetro não considerado foi a distorção geométrica — a câmera simulada gera imagens sem distorções. A Tabela 5.2 apresenta as características da câmera simulada.



TABELA 5.2 – Parâmetros intrínsecos da câmera usada nas simulações.

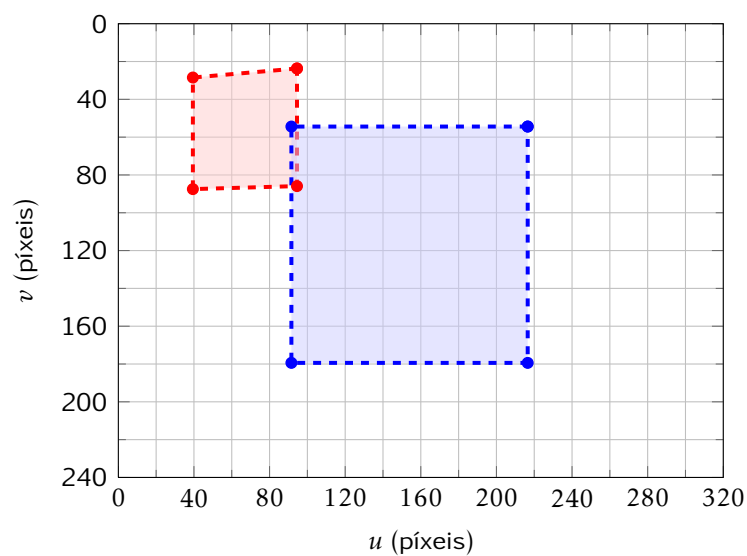
Parâmetro	Valor	Descrição
$u_0$	154	Coordenada horizontal em píxeis do ponto principal
$v_0$	117	Coordenada vertical em píxeis do ponto principal
$f$	$2 \times 10^{-3}$	Distância focal da lente em m
$\rho_u$	$1 \times 10^{-5}$	Dimensão horizontal do píxel em m
$\rho_v$	$1 \times 10^{-5}$	Dimensão vertical do píxel em m
$p_x$	200	Relação entre distância focal e dimensão horizontal do píxel
$p_y$	200	Relação entre distância focal e dimensão vertical do píxel

O objeto-alvo é composto por quatro círculos pretos cujos centros estão localizados nos vértices de um quadrado com 50 cm de lado. A Tabela 5.3 mostra as coordenadas das projeções do objeto-alvo na imagem com o quadrirrotor nas posições inicial e desejada e o erro no momento inicial. O menor erro inicial em  $u$  é 53 píxeis e 27 em  $v$ . Os controladores foram sintonizadas com o objetivo de fornecer um tempo de acomodação de 8 s usando o mesmo critério adotado na Seção 3.1.6. Para alcançar o tempo de acomodação definido, todos os erros em  $u$  devem ser menores do que 2 píxeis e 1 píxel em  $v$ .

**TABELA 5.3** – Coordenadas das posições inicial e desejada das características visuais e erro no momento inicial para as simulações.

Coordenada	Valor inicial	Valor desejado	Erro inicial
$u_1$	95	217	-122
$v_1$	24	55	-31
$u_2$	39	92	-53
$v_2$	28	55	-27
$u_3$	39	92	-53
$v_3$	87	180	-93
$u_4$	95	217	-122
$v_4$	86	180	-94

A Figura 5.4 mostra as posições iniciais (em vermelho) e desejadas (em azul) das características na imagem.



**FIGURA 5.4** – Posição inicial (vermelho) e desejada (azul) das características na imagem para as simulações.

No espaço Euclidiano, o centro desse quadrado está localizado a uma altura  $Z = 1$  m e na posição  $X = 1,5$  m e  $Y = 0$ . A postura desejada para o VANT é à frente do objeto-alvo, com o plano da imagem paralelo ao quadrado de referência. O vetor da postura desejada é  $\mathbf{r}_d = [X \ Y \ Z \ \phi \ \theta \ \psi]^T = [0,7 \ 0 \ 1 \ 0 \ 0 \ 0]^T$  e o vetor da posição inicial é  $\mathbf{r}_i = [0 \ -1 \ 0,5 \ 0 \ 0 \ -\pi/18]^T$ .

O módulo dos ângulos de rolagem e arfagem são limitados a 0,1 rad (aproximadamente  $5,7^\circ$ ). O objetivo desta limitação é evitar que, como dito anteriormente, o deslocamento das características visuais em função dos ângulos do quadricóptero seja elevado. O maior erro causado por tais rotações é de  $\pm 22$  pixels para o veículo na posição desejada.

### 5.3.2 Simulações com sistema cinemático completamente atuado com seis graus de liberdade

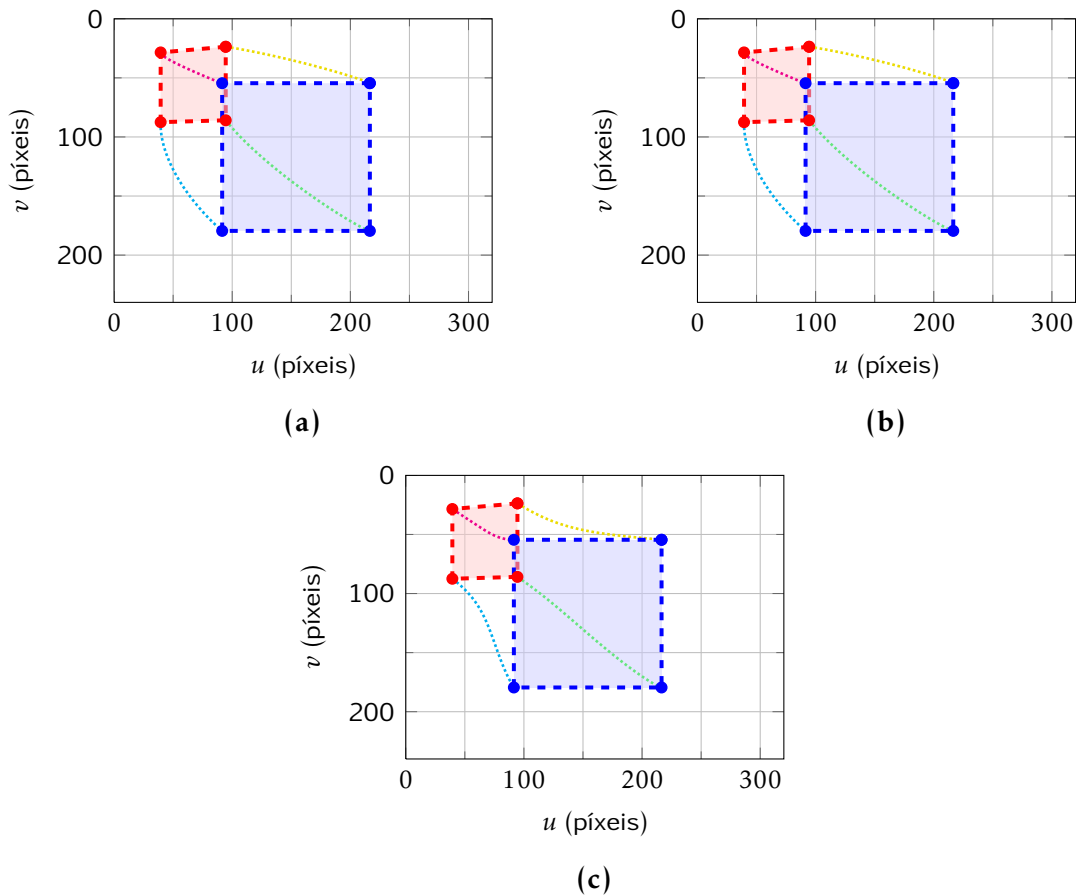
Estas simulações comparam o desempenho das três abordagens de controle — a clássica e as duas sugeridas — sem a interferência de fatores externos tais como ruídos de sinais, perturbações físicas, erros de modelagem e características dinâmicas do sistema controlado. Usou-se um sistema genérico completamente atuado com seis graus de liberdade puramente cinemático. Todas as posturas estão expressas em relação a um sistema de coordenadas fixo no ambiente. As rotações seguem o padrão de rolagem-arfagem-guinada (como descrito no Apêndice B) em relação ao sistema de coordenadas da câmera, cuja origem coincide com o sistema de coordenadas do veículo mostrado na Figura 4.1.

Os controladores foram sintonizados para que o tempo de acomodação da tarefa

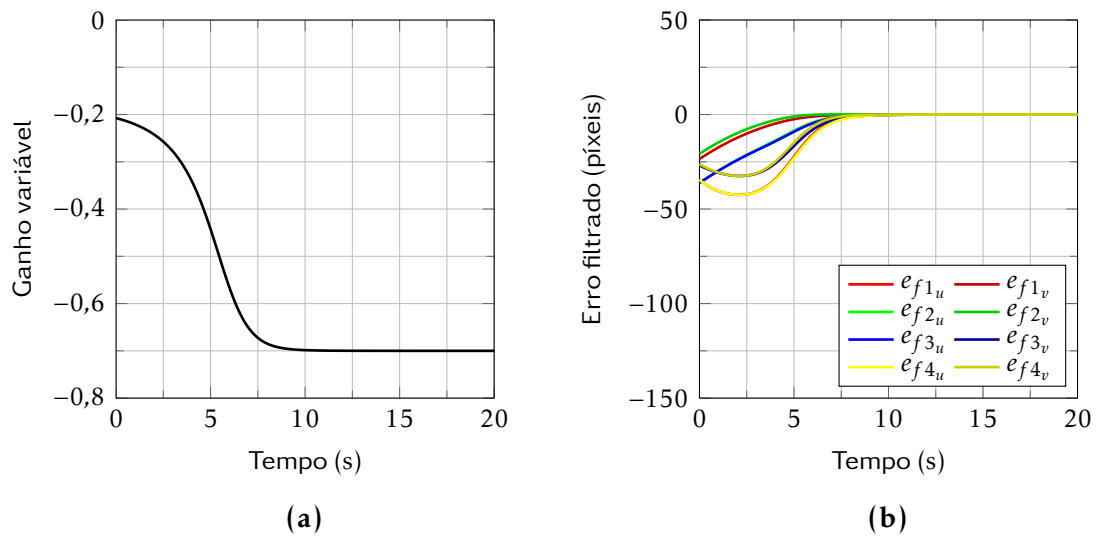
fosse de 8 s, com o mesmo critério usado para o experimento da Seção 3.1.6.

Em decorrência do decaimento não exponencial do erro para o filtro *fuzzy*, as trajetórias das características visuais na imagem para o controlador com erro filtrado (5.5c) não são tão retilíneas quanto as dos outros dois controladores (5.5a e 5.5b). A Figura 5.6a mostra evolução do ganho variável e a Figura 5.6b mostra o efeito do filtro *fuzzy* sobre o erro — os valores absolutos e as diferenças entre os erros de cada referência são reduzidos, para os controladores com ganho variável e filtro *fuzzy*, respectivamente.

O decaimento do erro (5.7a e 5.7b), variação da postura (5.8a e 5.8b) e da orientação (5.8d e 5.8e) apresentam resultados semelhantes para o ganho proporcional constante e o variável. Com o filtro *fuzzy* (5.7c, 5.8c e 5.8f) estes resultados são diferentes em decorrência do decaimento não exponencial do erro para esta técnica. As amplitudes máximas do sinal de controle para o ganho variável (5.7e) e o filtro *fuzzy* (5.7f) são menores que para o controlador clássico (5.7d). Enquanto o controlador com ganho proporcional constante gera sinais de amplitudes próximas de 0,3 m/s, nos outros dois casos esses valores são da ordem de 0,2 m/s, mantendo o tempo de acomodação.



**FIGURA 5.5** – Projeções inicial (vermelho) e final (azul) das características visuais na imagem para as simulações com o sistema cinemático completamente atuado com seis graus de liberdade. (a) ganho proporcional constante com  $\lambda = 0,4$ ; (b) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) filtro *fuzzy* com  $\lambda = 0,62$ .



**FIGURA 5.6** – (a) evolução do ganho variável e (b) erro filtrado para as simulações com sistema cinemático completamente atuado com seis graus de liberdade.

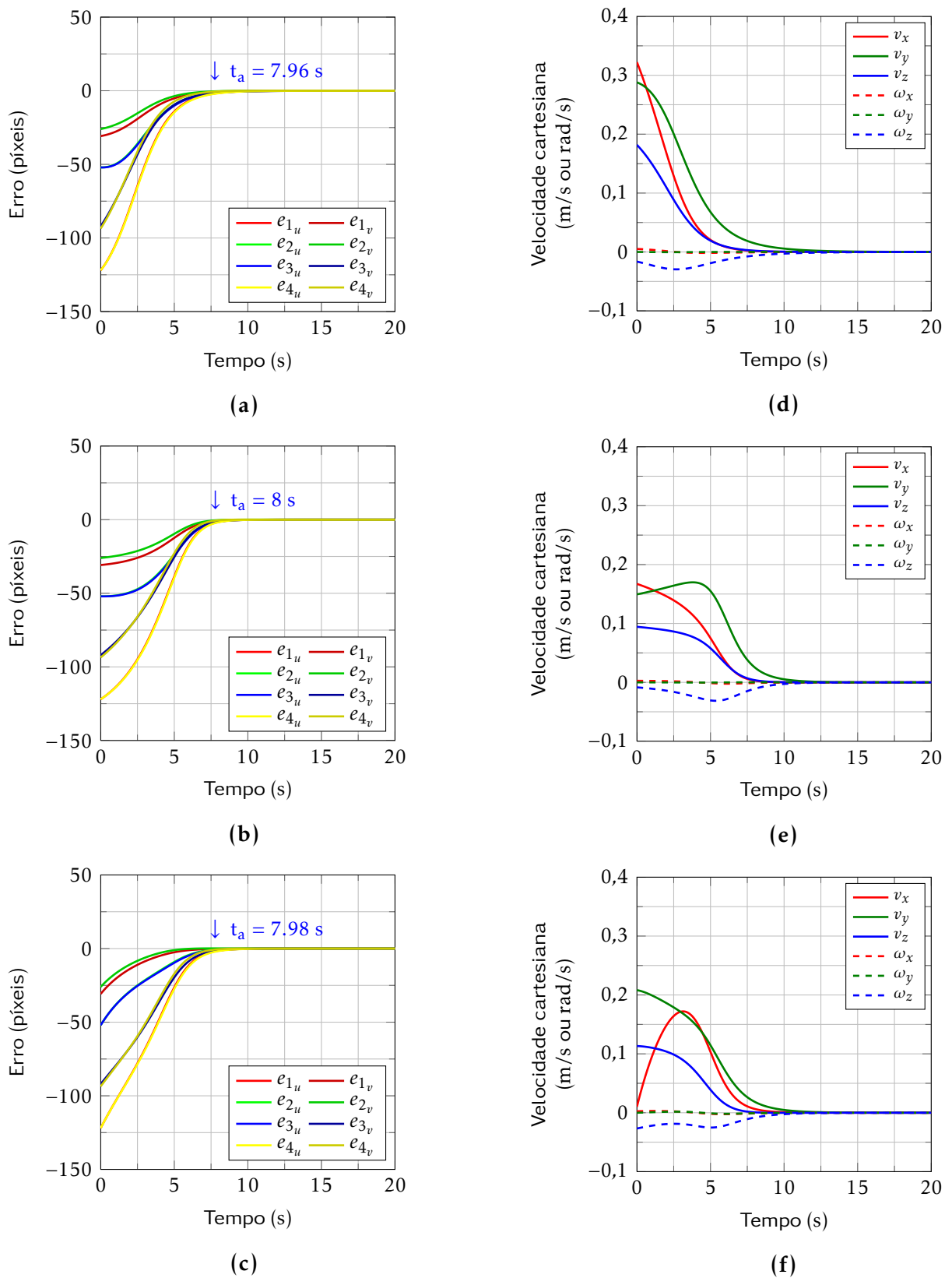
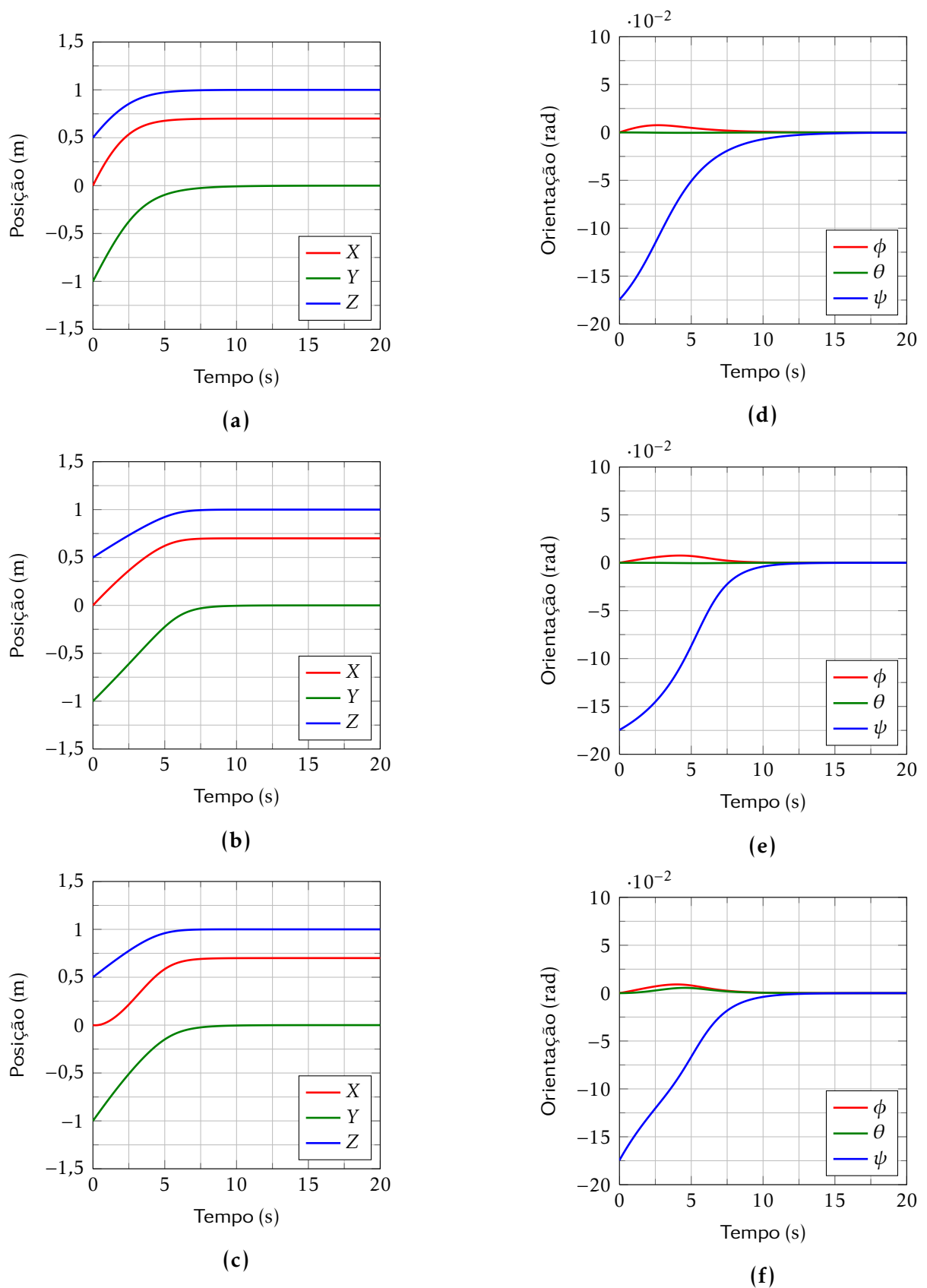


FIGURA 5.7 – Erro e velocidades cartesianas para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com  $\lambda = 0,4$ ; (b) e (e) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 0,62$ .



**FIGURA 5.8** – Posição e orientação para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com  $\lambda = 0,4$ ; (b) e (e) ganho variável com  $\lambda(0) = 0,7$ ,  $\lambda(\infty) = 0,1$  e  $\lambda'(0) = 450$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 0,62$ .

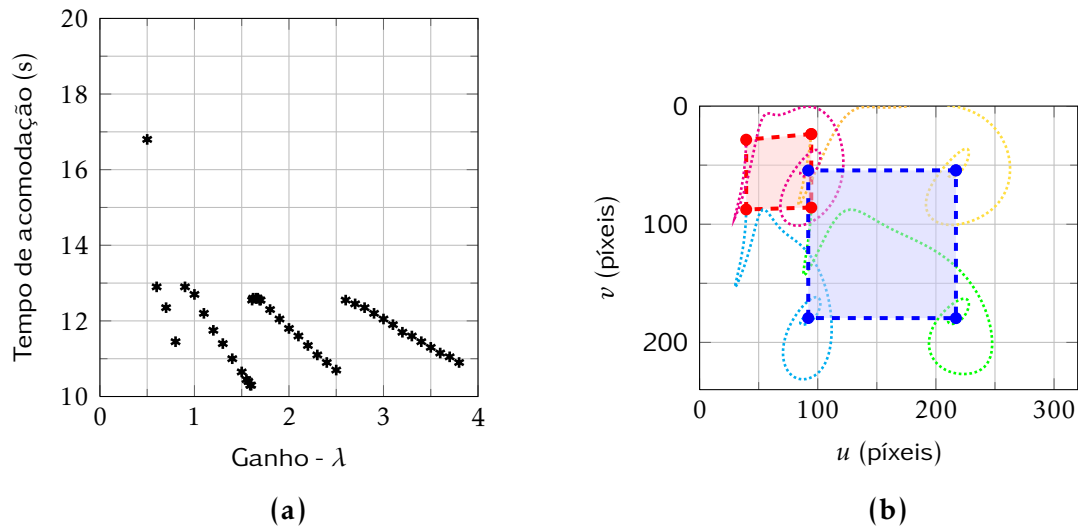
### 5.3.3 Simulações com o modelo dinâmico do quadricóptero

Para as simulações com o modelo dinâmico do quadricóptero usou-se um modelo deste veículo disponível no *Robotics Toolbox*. Todas as demais configurações da simulação são idênticas às da Seção 5.3.2. Como os quadricópteros são sistemas subatuados, os sinais de controle de velocidade são apenas quatro: três de velocidade de translação e um de velocidade de rotação.

Ao contrário do que ocorre com as simulação da Seção 5.3.2 que não consideram o modelo dinâmico, nem sempre um ganho maior implica um tempo de acomodação menor. Nestas simulações foi impossível atingir a meta de 8s para o tempo de acomodação. O menor valor conseguido para o caso clássico foi de 10,3s para  $\lambda = 1,6$ . Testou-se diversos valores para o ganho desse controlador e o resultado está na Figura 5.9a. Percebe-se que o tempo de acomodação não varia linearmente com o ganho e, teoricamente, outros mínimos podem existir. Contudo, a partir de  $\lambda = 3,5$  a trajetória das características visuais saem da imagem, evidenciado na Figura 5.9. Numa situação real, isso causaria a falha da tarefa. Para valores menores que  $\lambda = 1,6$ , a trajetória das características na imagem é melhor, porém os tempos de acomodação tendem a aumentar.

As trajetórias das características visuais na imagem (5.10a, 5.10b e 5.10c) mostram que logo no início da tarefa as características se movem em função do aumento dos ângulos de rotação e arfagem. Tal aumento dos ângulos causa uma variação do erro, que gera uma variação no sinal de controle e que, novamente, implica ainda mais mudanças nos ângulos do robô. Como consequência, os primeiros instantes são marcados por uma oscilação que pode ser verificada na evolução do ganho variável (5.11a) e no erro filtrado





**FIGURA 5.9** – Variação do tempo de acomodação para diferentes valores do ganho ((a)) e trajetória das características visuais na imagem para  $\lambda = 3,5$  ((b)).

(5.11b), nos sinais de controle (5.12d, 5.12e e 5.12f), erro (5.12a, 5.12b e 5.12c) e postura (5.13a, 5.13d, 5.13b, 5.13e, 5.13c e 5.13f). Nos instantes intermediários, estes valores variam menos e o aspecto oscilatório quase desaparece. Próximo à postura desejada ocorrem novas oscilações pelos mesmos motivos.

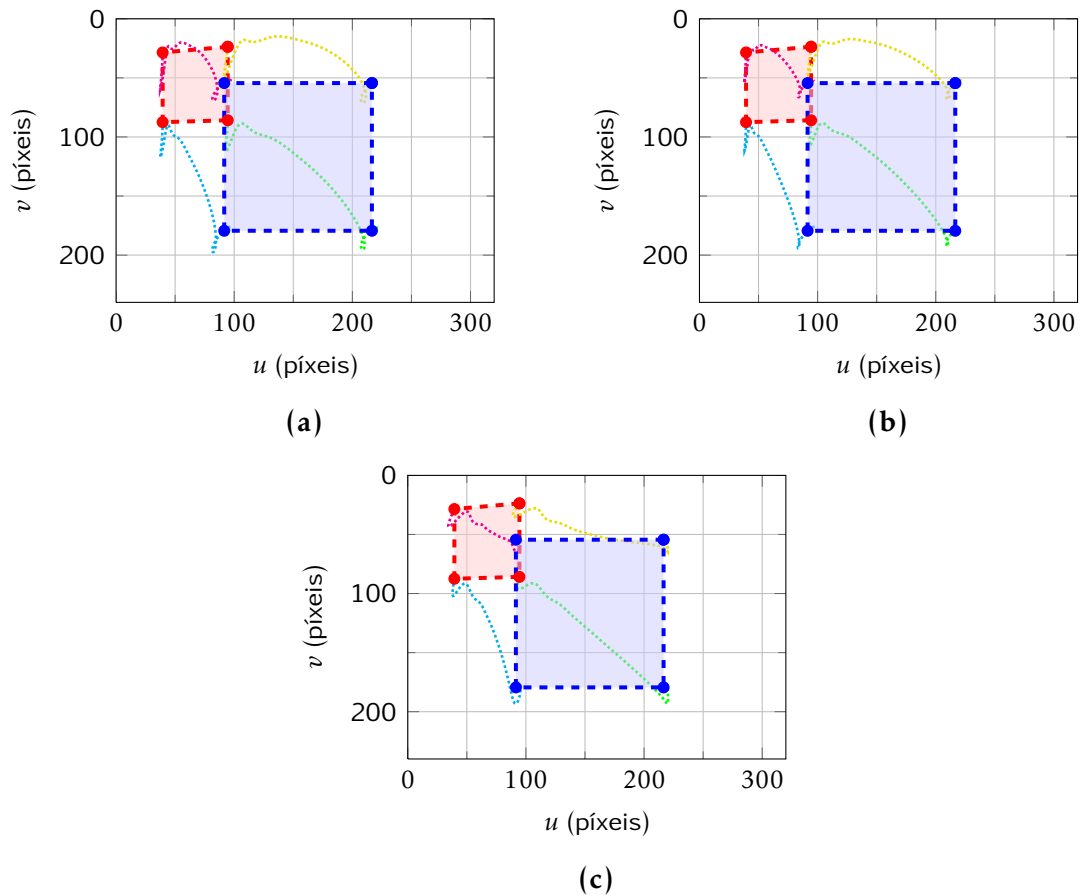
Apesar dos tempos de acomodação maiores, os sinais de controle são também mais elevados que os obtidos com as simulações que não consideram as propriedades dinâmicas do veículo. Esse fato era esperado, já que a degradação da trajetória (menos semelhantes a retas) no plano da imagem implica movimentos cartesianos também piores (também menos retilíneos). Realizar tais movimentos exige maiores deslocamentos e, conseqüentemente, maiores velocidades.

Os tempos de acomodação mínimos conseguidos com o ganho variável (5.12b) e o filtro *fuzzy* (5.12c) foram menores do que com o ganho constante (5.12a). Essa redução, entretanto, é relativamente pequena — 0,15 s para o variável e *filtro* 0,55 s para o *filtro*.

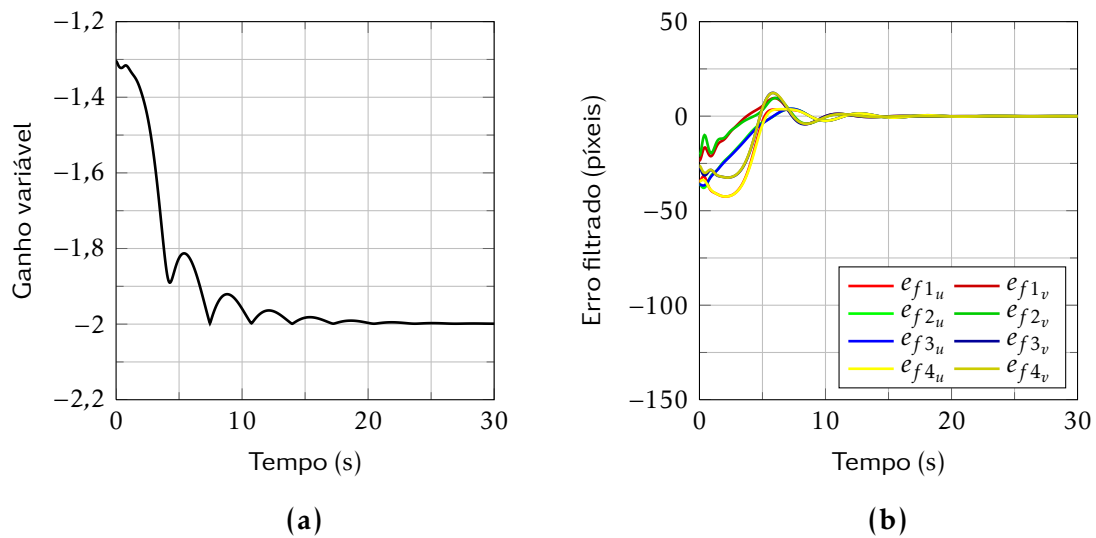
Os dois controladores alternativos obtiveram sinais de controle com amplitude menores que o clássico (0,44 m/s, 0,36 m/s e 0,4 m/s para o ganho proporcional constante,

variável e filtro *fuzzy*, respectivamente) e tempos de acomodação menores. Ambos apresentaram oscilações mais suaves nas primeiras iterações e próximo da postura desejada que com o ganho proporcional constante, como mostram as evoluções da orientação (5.13d, 5.13e e 5.13f. Esse fato contribui para as melhores trajetórias das características visuais na imagem (5.10a, 5.10b e 5.10c).

Por conta dessa oscilação, foi necessário aumentar o valor de  $\epsilon$  na Equação 4.16, pois com o aumento do ângulo de arfagem nos instantes iniciais, o erro real também aumenta. Caso o erro máximo definido em (4.16) seja menor do que o erro real, o sinal de controle resultante será no sentido contrário. Isso causa uma translação de afastamento do veículo, o que não é desejável.



**FIGURA 5.10** – Projeções inicial (vermelho) e desejada (azul) das características visuais na imagem para as simulações com o modelo dinâmico de um quadricóptero. (a) ganho proporcional constante com  $\lambda = 1,6$ ; (b) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) filtro *fuzzy* com  $\lambda = 2,8$ .



**FIGURA 5.11** – (a) evolução do ganho variável e (b) erro filtrado para as simulações com sistema cinemático completamente atuado com seis graus de liberdade.

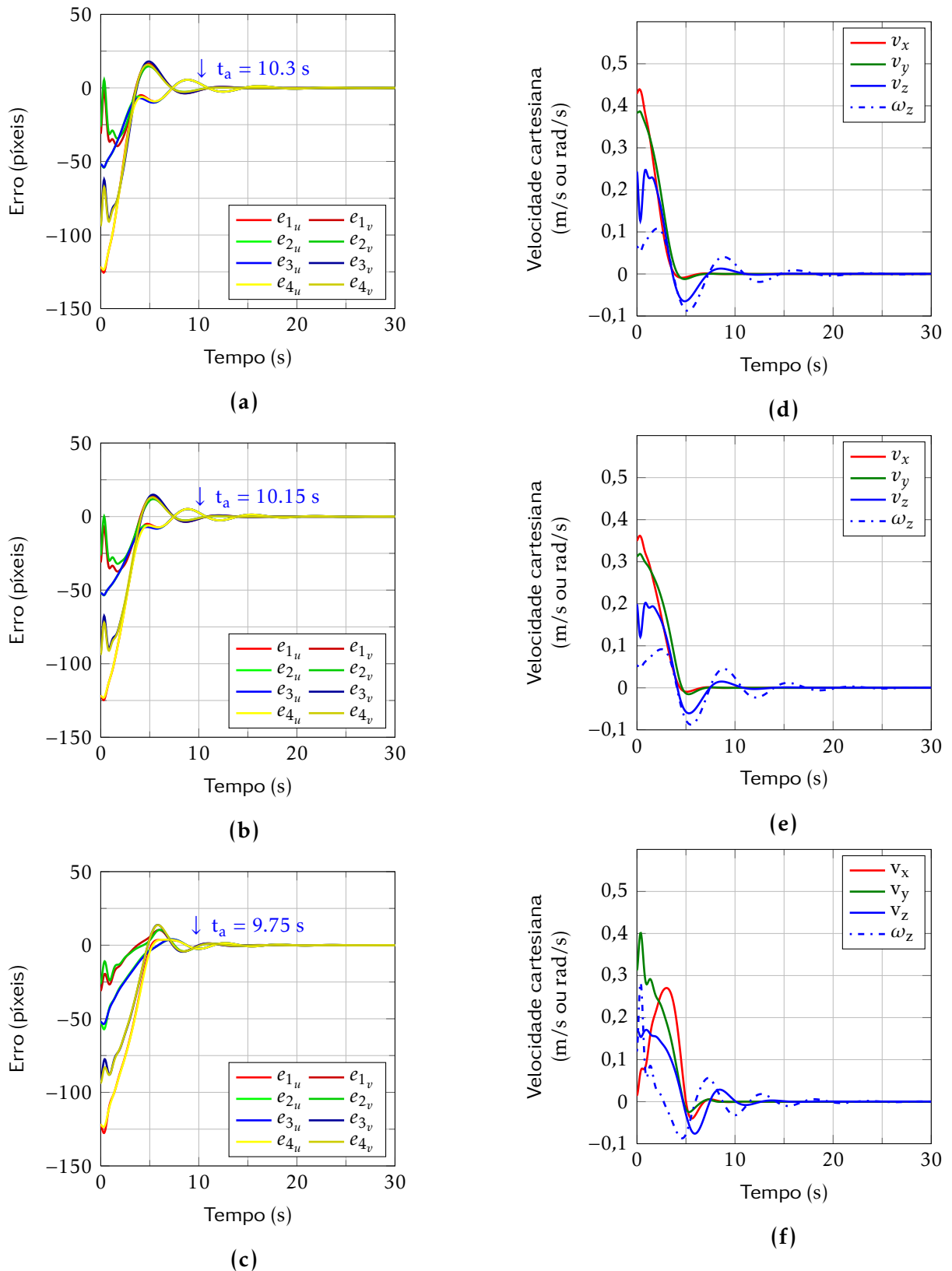
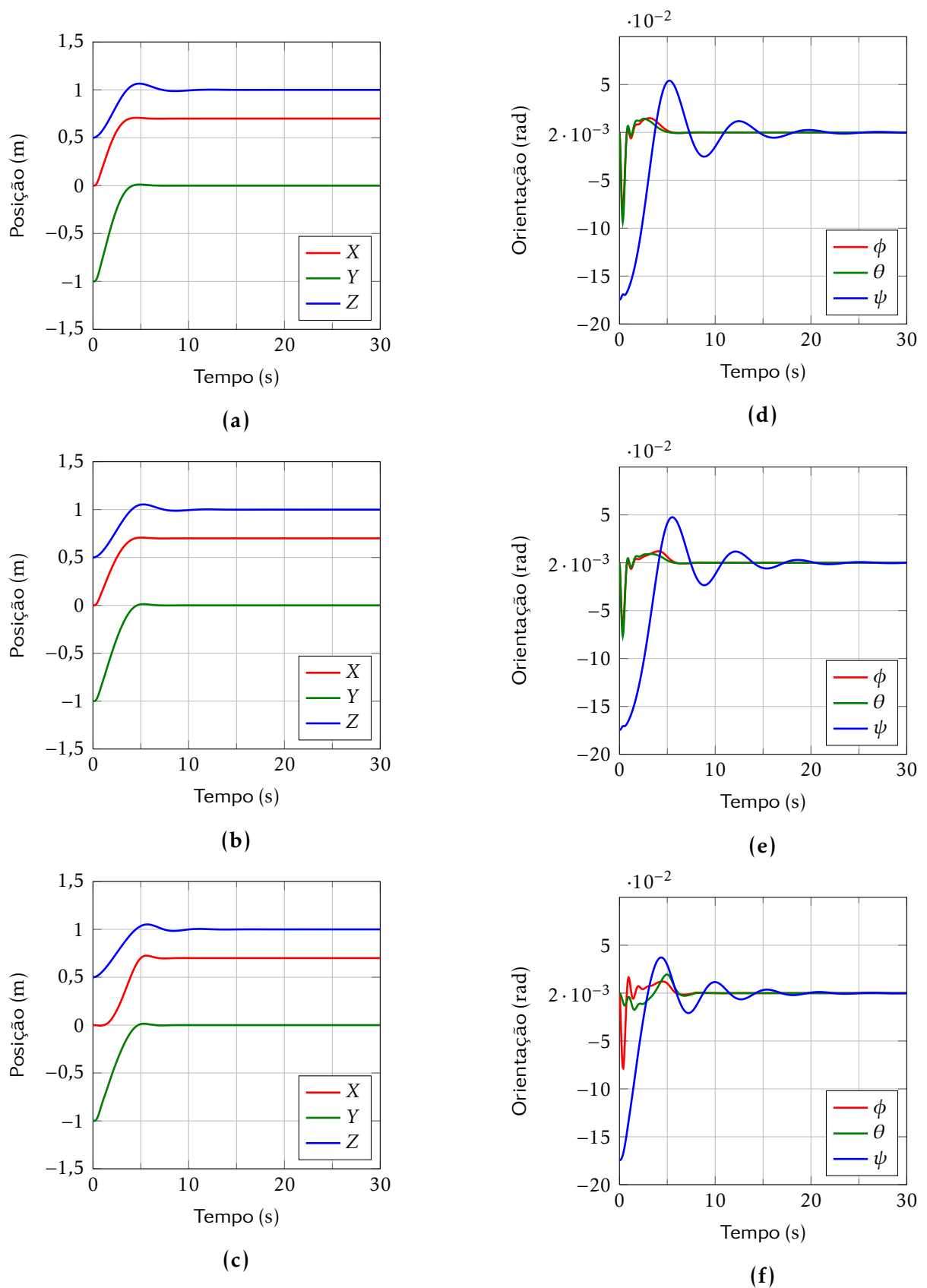


FIGURA 5.12 – Erro e velocidades cartesianas para as simulações com o modelo dinâmico do quadrrrotor. (a) e (d) ganho proporcional constante com  $\lambda = 1,6$ ; (b) e (e) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 2,8$ .



**FIGURA 5.13** – Posição e orientação para as simulações com sistema cinemático completamente atuado com seis graus de liberdade. (a) e (d) ganho proporcional constante com  $\lambda = 1,6$ ; (b) e (e) ganho variável com  $\lambda(0) = 2, \lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 2,8$ .

## 5.4 Dados reais

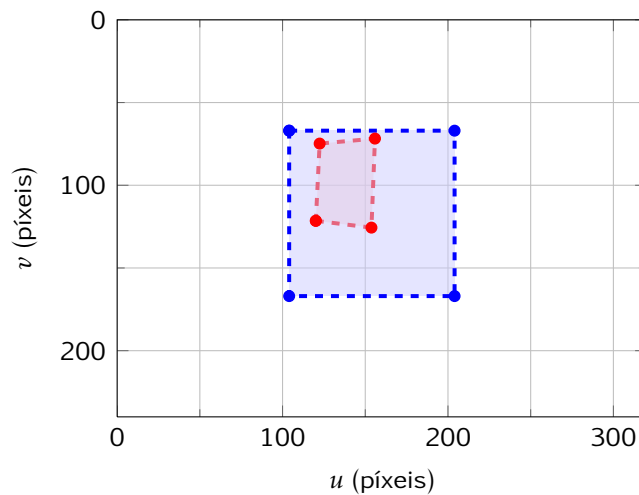
### 5.4.1 Aplicação no AR.Drone

Após as simulações, validaram-se os três controladores no AR.Drone. Os conhecimentos adquiridos nas etapas anteriores facilitam a sintonia dos controladores do veículo real. Os problemas específicos desse sistema foram relatados na Seção 5.2.1.1 e foram levados em consideração.

As posições inicial e desejada das características visuais são semelhantes às definidas na Seção 5.3.1. No entanto, fizeram-se algumas mudanças para facilitar a realização dos experimentos. A posição inicial foi colocada mais ao centro da imagem. O objetivo dessas mudanças foi facilitar o posicionamento inicial do veículo — feito por um operador — antes da servovisão. Como esse posicionamento inicial é feito manualmente por um operador, há uma variação entre a posição inicial entre experimentos. A posição desejada foi afastada das extremidades da imagem para evitar que as características visuais saíssem da imagem durante a execução da tarefa, especialmente para o controlador com ganho proporcional. Os valores da posição desejada das características visuais estão expostos na Tabela 5.4. A Figura 5.14 mostra a posição inicial e desejada para as características visuais.

**TABELA 5.4** – Coordenadas da posição desejada das características visuais para a aplicação no quadrrrotor.

Coordenada	Valor desejado
$u_1$	204
$v_1$	67
$u_2$	104
$v_2$	67
$u_3$	104
$v_3$	167
$u_4$	204
$v_4$	167

**FIGURA 5.14** – Posição inicial (vermelho) e desejada (azul) das características da imagem para a aplicação no quadrrrotor.

Para se obter as mudanças supracitadas nas posições inicial e desejada das características, as posturas foram alteradas de acordo com o necessário. Primeiro aumentou-se a distância do VANT para o objeto-alvo na postura desejada em 20 cm. Segundo,

alterou-se o ângulo de guinada inicial para  $\psi = -\pi/5$  rad e a altura inicial para  $Z = 1$  m. A postura inicial é, portanto,  $\mathbf{r}_i = [0 \quad -1 \quad 1 \quad 0 \quad 0 \quad -\pi/5]^T$  e a desejada é  $\mathbf{r}_d = [0,5 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]^T$ . A Figura 5.15 mostra o AR.Drone no ambiente da tarefa, com os quatro círculos do objeto-alvo e, no canto inferior esquerdo, a imagem capturada pela câmera frontal embarcada no veículo.

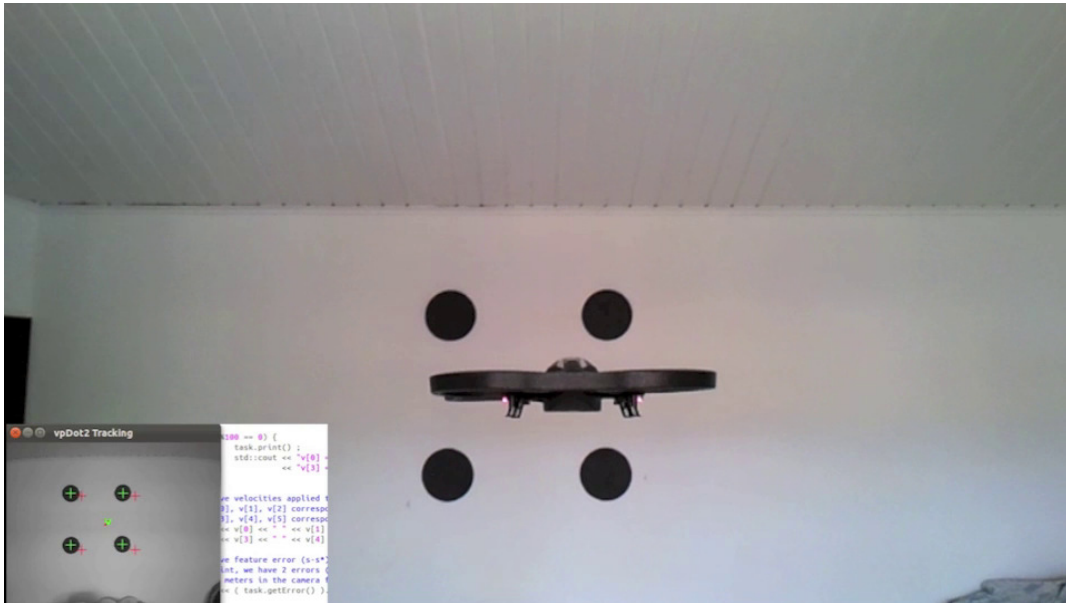


FIGURA 5.15 – O AR.Drone no ambiente de trabalho e a imagem capturada pela câmera frontal do veículo (canto inferior esquerdo).

O AR.Drone não fornece sua posição no espaço. Portanto, para auxiliar e verificar o posicionamento inicial e desejado do veículo, marcou-se o ambiente da tarefa (chão e paredes) com indicações das respectivas posições de  $X$ ,  $Y$  e  $Z$  e do ângulo de guinada. Antes do início da tarefa de controle é necessário colocar o veículo em voo manualmente. Ele é controlado remotamente por um operador usando o teclado do computador até essas marcações. Depois passa-se o comando ao controlador servo visual. Por ser feito de forma manual e visual, a precisão de tal posicionamento feito pelo operador é grosseira. Portanto, o único meio de verificação de erro confiável é por através da imagem.

Os sinais de controle de translação na direção  $x$  e  $y$  são definidos em função dos



ângulos de rolagem e arfagem, respectivamente. O de translação em  $z$  é definido em metros por segundo e o de velocidade de guinada em radianos por segundo. Os gráficos apresentados nesta seção seguem estas propriedades. Usa-se a mesma escala nos gráficos para facilitar a comparação entre os controladores. A malha de controle é executada a uma taxa de 100 Hz, contudo a câmera frontal do veículo opera a uma taxa de 15 quadros por segundo.

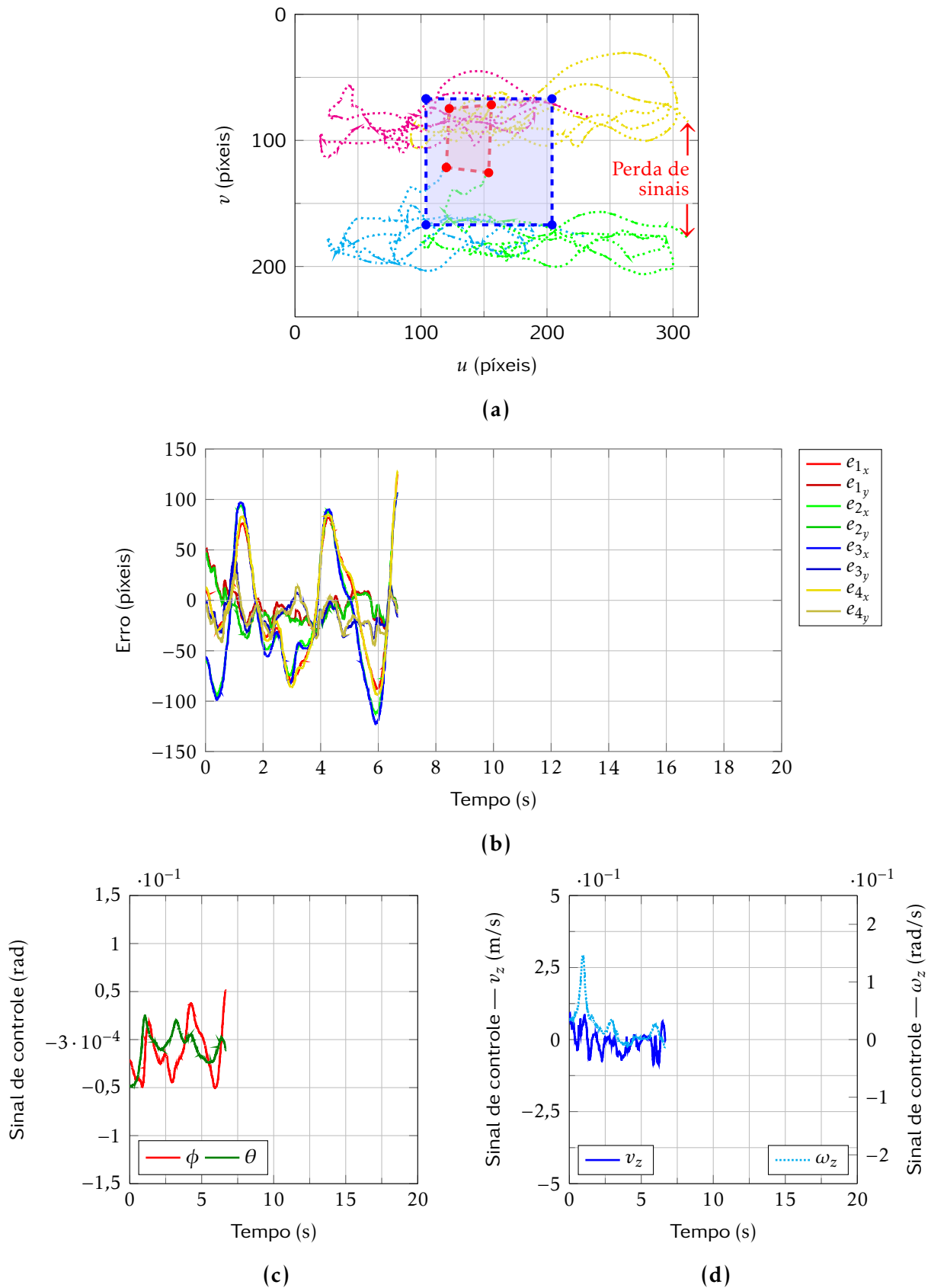
Valores mais baixos do ganho (como  $\lambda = 1,6$ , mostrado na Figura 5.16) não conseguem manter o quadricóptero próximo do objetivo (quando o erro é relativamente pequeno). Também têm problemas em superar perturbações externas e o *drift* do próprio veículo e a tarefa falha rapidamente. Ganhos maiores (como  $\lambda = 3$ ) conseguem inicialmente manter o veículo em torno do objetivo ao custo de uma trajetória das características na imagem insatisfatória (5.17a). As altas amplitudes dos sinais de controle aumentam o efeito do deslocamento das referências em decorrência dos ângulos de rolagem e arfagem. Esse aumento dos ângulos amplifica o movimento oscilatório descrito anteriormente e acaba por fazer com que as características saiam da imagem causando mais uma vez a falha da tarefa.

O ganho variável não melhorou a trajetória na imagem e em alguns instantes ficou próximo de deixar algumas das características visuais saírem da imagem (5.17b) — porém isso não chegou a acontecer nos experimentos. Já o controlador com erro filtrado apresenta um desempenho melhor nesse quesito. O movimento das características visuais ficam restritos em uma região menor em torno da posição desejada do que nos demais casos (5.17c). A evolução dos erros (5.18a, 5.18b e 5.18c) corrobora com esse resultado, com o erro do filtro *fuzzy* apresentando amplitudes menores que nos demais controladores.

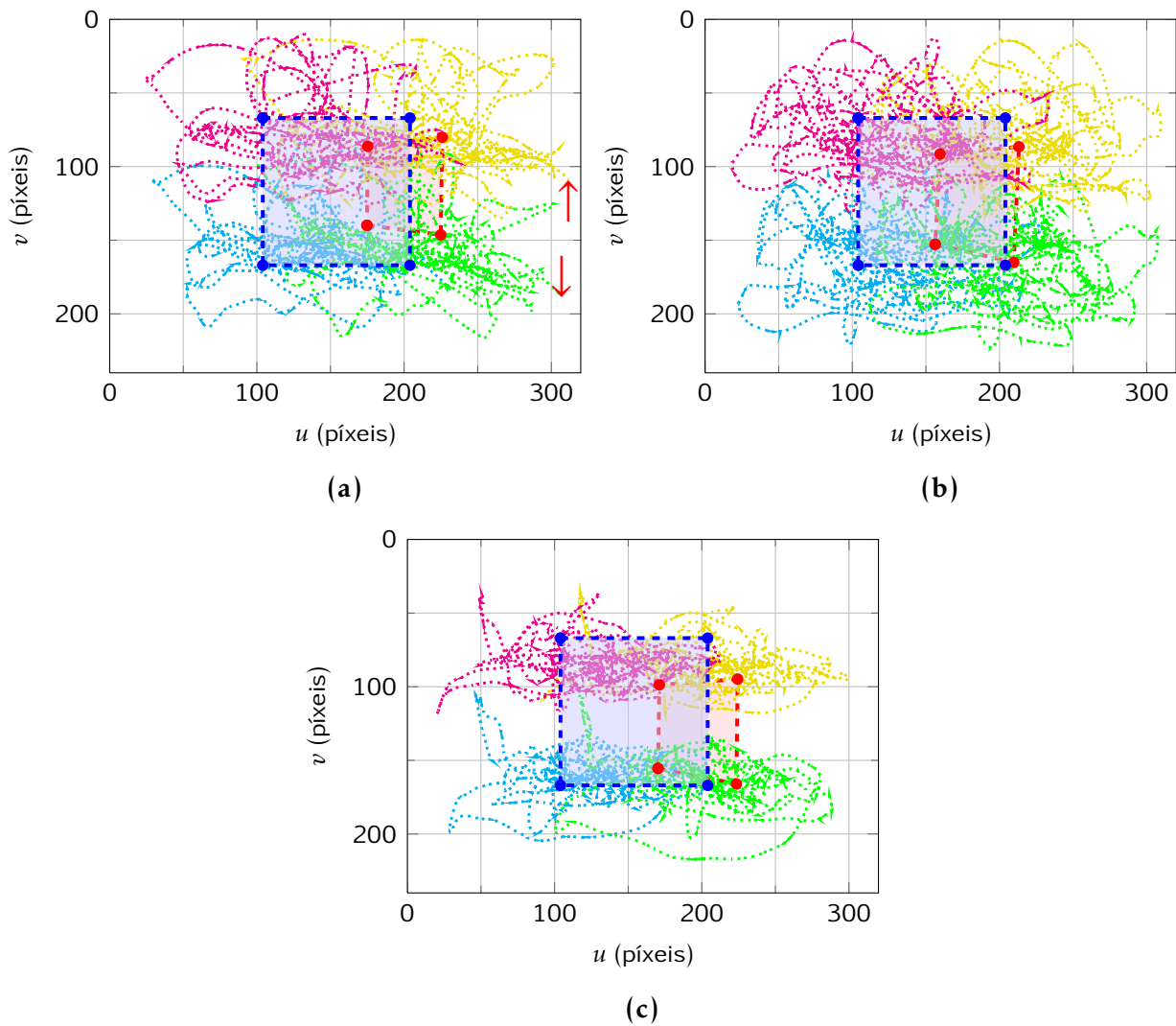
Como esperado, o esforço de controle dos controladores alternativos (5.19b, 5.19e, 5.19c e 5.19f) é consideravelmente menor do que o do ganho constante: 0,41 m/s, 0,14 m/s e 0,13 m/s para  $v_z$  com ganho constante, ganho variável e filtro *fuzzy*, respectivamente. Em comparação com as simulações, nota-se um sinal de velocidade inicial para  $\omega_z$  elevado. Isso é justificado pelo do aumento do ângulo de guinada na postura inicial para a aplicação no AR.Drone, como descrito na Seção 5.4.1.

Um resultado inesperado foi a trajetória das características visuais ter sido melhor para o caso com erro filtrado do que o variável. Sabendo que no filtro *fuzzy* o decaimento do erro não é exponencial e com base nas simulações, deveria ocorrer o contrário. Isso provavelmente se deve a uma pior sintonia conseguida para o controlador variável.

Nenhum dos controladores conseguiu manter o erro constantemente dentro do objetivo definida de 5 %. Por esse motivo, não se indicam os tempos de acomodação nos gráficos.



**FIGURA 5.16** – Aplicação do IBVS ao quadrirrotor com ganho proporcional constante  $\lambda = 1,6$ . (a) projeções inicial e final das características visuais; (b) erro das características visuais; (c) sinal de controle de rolagem e arfagem; (d) sinal de controle de velocidade vertical e velocidade de guinada.



**FIGURA 5.17** – Projeções inicial (vermelho) e desejada (azul) das características visuais na imagem para a aplicação no quadricóptero. (a) ganho proporcional constante com  $\lambda = 3$ ; (b) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) filtro *fuzzy* com  $\lambda = 3$ .

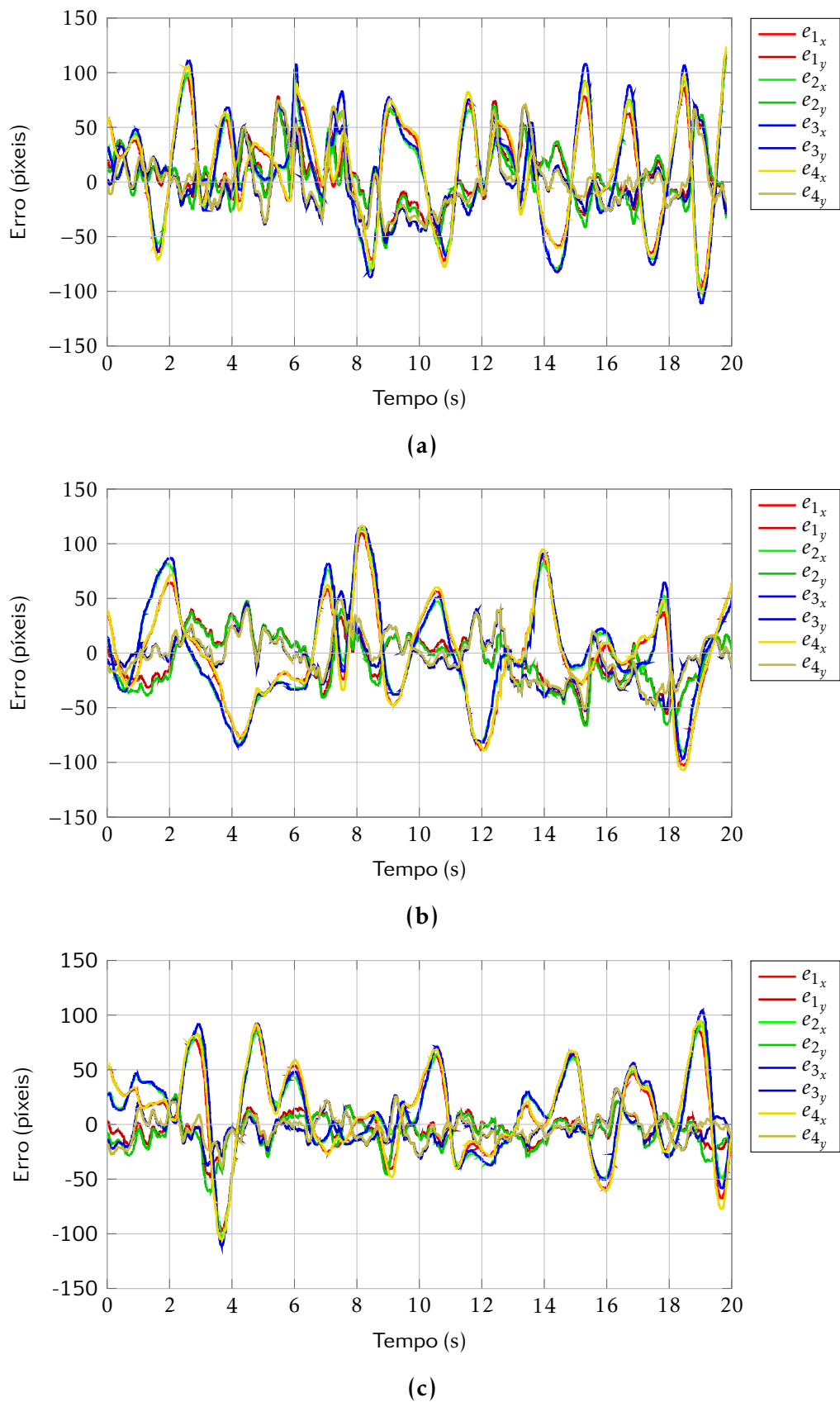
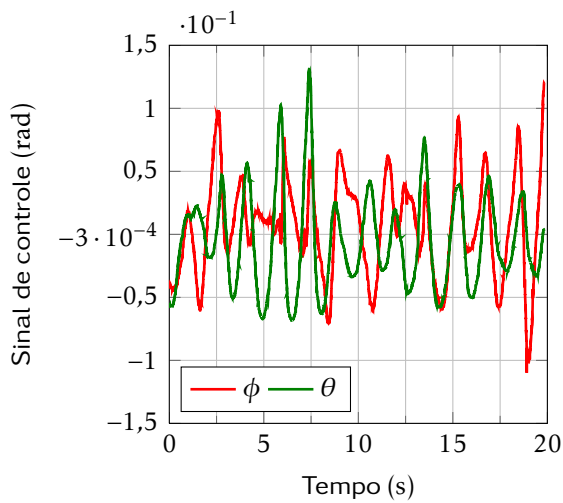
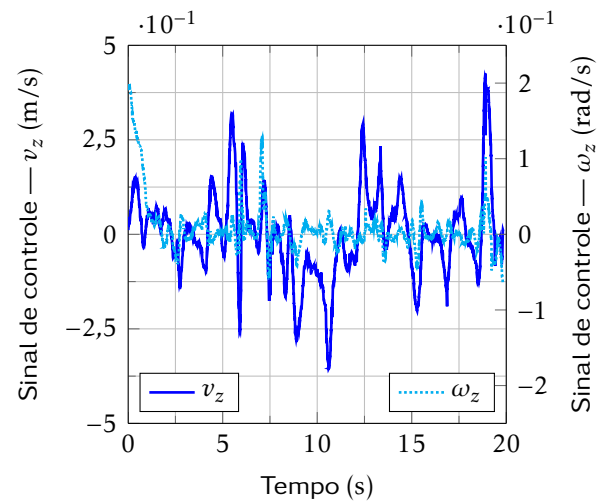


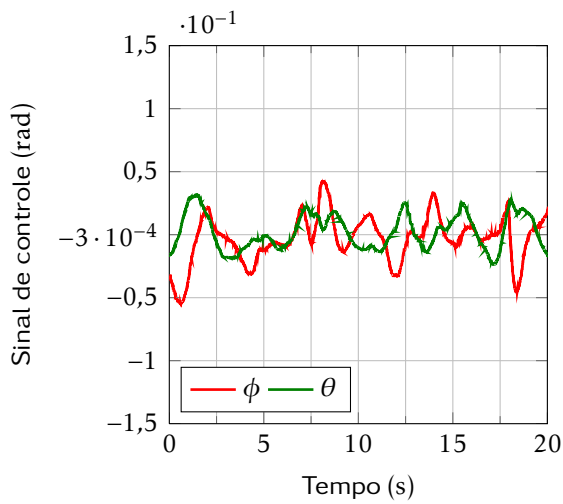
FIGURA 5.18 – Erro para a aplicação no quadricóptero. (a) ganho proporcional constante com  $\lambda = 3$ ; (b) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) filtro *fuzzy* com  $\lambda = 3$ .



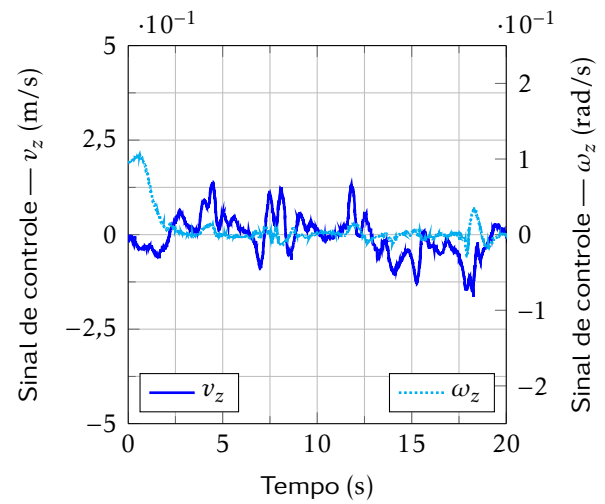
(a)



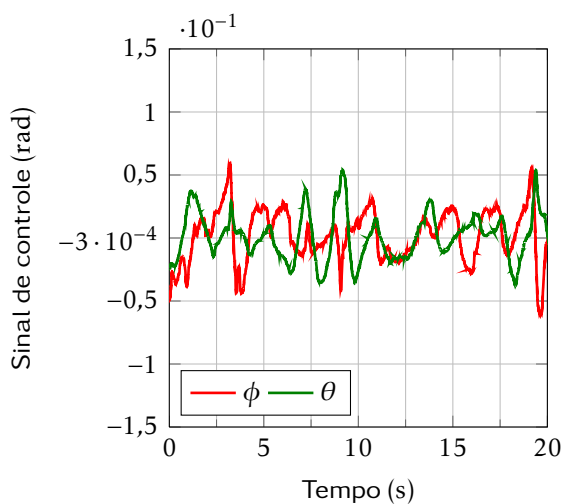
(d)



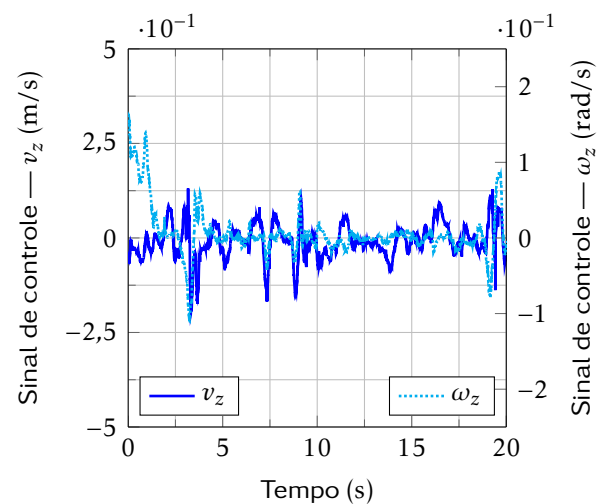
(b)



(e)



(c)



(f)

**FIGURA 5.19** – Sinal de controle para a aplicação no quadricóptero. (a) e (d) ganho proporcional constante com  $\lambda = 3$ ; (b) e (e) ganho variável com  $\lambda(0) = 2$ ,  $\lambda(\infty) = 1,2$  e  $\lambda'(0) = 700$ ; (c) e (f) filtro *fuzzy* com  $\lambda = 3$ .

## 5.5 Considerações

Este capítulo aplicou as três abordagens de IBVS discutidas neste texto. Os resultados dos experimentos simulados e reais mostraram que as duas alternativas indicadas para melhorar o IBVS possuem desempenho superior ao ganho constante. A influência das propriedades dinâmicas do robô piora consideravelmente o desempenho de todos os controladores, principalmente na aplicação no VANT real.

O Capítulo 6 faz uma breve revisão do que foi discutido neste texto. Apresenta possíveis soluções para alguns problemas encontrados nos experimentos e oferece algumas propostas de trabalhos futuros.

## 6 Conclusão

Este texto apresentou os resultados de um estudo sobre o controle de robôs móveis aéreos multirrotores usando visão computacional. Especificamente, tratou da metodologia de regulação implícita da postura de robôs usando características da imagem como referências — o controle servo visual baseado em imagem. Apresentou alguns dos principais resultados da área por meio de uma revisão abrangente da literatura, desde os primórdios da técnica até os avanços recentes. Mostrou as principais abordagens da servovisão: 2D ou controle servo visual baseado em imagem, que usa as características da imagem diretamente na lei de controle (IBVS); e 3D ou controle servo visual baseado em posição (PBVS), que usa informação visuais para estimar a postura tridimensional do robô, definindo o erro no espaço Euclidiano. Citou, também, algumas formulações híbridas como o controle particionado e o controle servo visual 2 1/2 D, que tentam aproveitar as vantagens de cada método. A prova rigorosa da estabilidade global do controle servo visual baseado em imagem clássico é difícil, especialmente pela restrição de visibilidade. No entanto, na literatura prova-se que é possível definir uma vizinhança em torno de um ponto onde a estabilidade do IBVS é garantida.

Algumas das principais características da servovisão 2D foram mostradas. O fato do IBVS ser robusto a ruídos na imagem, erros nos parâmetros intrínsecos da câmera, não requerer estimação da postura e demandar menor processamento computacional,



faz dessa técnica uma alternativa viável para aplicação em VANTs quadricópteros com instrumentos de baixa precisão. Considerando o baixo custo desse tipo de veículo e as possibilidades de pesquisa que oferecem, têm se tornado uma plataforma comum em laboratórios, o que instigou a robótica aérea.

As propriedades dinâmicas dos quadricópteros impõem restrições à aplicação da estrutura clássica cinemática da servovisão. É comum encontrar na literatura estudos que incorporam o modelo dinâmico do veículo no controlador ou usam características visuais e modelos de projeção menos comuns. Tais abordagens, porém, aumentam a complexidade geral do sistema de controle. Este trabalho sugeriu a aplicação de duas técnicas para melhorar o desempenho do IBVS sem aumentar sua complexidade: o ganho variável e o filtro *fuzzy*. O ganho variável causa um decaimento exponencial do ganho proporcional do controlador em função do erro — erros maiores recebem ganhos menores e vice-versa. Esta técnica não degrada a trajetória no espaço da imagem, pois preserva o decaimento exponencial do erro. Por outro lado, requer a sintonia de três parâmetros. O filtro *fuzzy* mantém o ganho proporcional constante na lei de controle mas obtém um efeito semelhante ao do ganho variável, já que filtra os valores do erro com base numa função de pertinência triangular. Assim, o filtro diminui a diferença entre os erros das diferentes características visuais e tende a gerar sinais de controle com menor variação entre iterações. A sintonia do filtro *fuzzy* é mais simples pois possui apenas um parâmetro que pode ser determinado automaticamente. Em contrapartida, o decaimento do erro não é exponencial e, teoricamente, piora a trajetória das características visuais. Simularam-se todas essas alternativas com modelos de quadricópteros que não consideram as propriedades dinâmicas destes veículos. Essas simulações permitiram observar a influência de cada controlador no desempenho do

sistema.

Uma vez verificadas as vantagens das técnicas sugeridas, novas simulações foram feitas considerando as propriedades dinâmicas de um quadricóptero. Mais uma vez verificou-se que o ganho variável e o filtro *fuzzy* eram capazes de reduzir o esforço de controle e o tempo de acomodação em relação ao ganho proporcional constante. A inclusão do modelo dinâmico introduziu oscilações nos momentos iniciais e finais da tarefa, mas a amplitude dessas oscilações foram menores nos dois controladores sugeridos — isso foi ainda mais evidente para o caso filtrado.

Por fim, as técnicas supracitadas de controle foram validadas em uma plataforma experimental real de baixo custo disponível comercialmente, o AR.Drone. Esse sistema é composto de instrumentos pouco precisos e câmeras ruidosas, além de ser sensível a perturbações externas. O controlador servo visual clássico não conseguiu manter as características visuais próximas da posição desejada por muito tempo. As oscilações causadas pelos elevados sinais de controle evidenciaram a dificuldade desta técnica quando aplicada em quadricópteros. Os dois controladores alternativos foram capazes de manter as características visuais em torno da referência. Contudo, o caso variável não melhorou a trajetória no espaço da imagem, enquanto o filtro *fuzzy* foi capaz de limitar o deslocamento visual em uma região menor. Teoricamente, e como foi evidenciado nas simulações, o controlador com filtro *fuzzy* deveria ser pior que o variável nesse aspecto pois não garante um decaimento exponencial do erro. Esse resultado inesperado pode ser devido a uma pior sintonia do caso variável.

## 6.1 Perspectivas futuras

As perspectivas futuras e recomendações relacionadas a seguir foram escolhidas com o objetivo de complementar este trabalho:

1. realizou-se todo o trabalho descrito neste texto sem conhecimento da relação entre o sinal de controle do AR.Drone (os ângulos de rolagem, arfagem e guinada) e as velocidades espaciais decorrentes de tais ângulos. Como o controle servo visual calcula sinais de velocidade, é importante conhecer a relação entre as duas grandezas. É interessante, também, testar as técnicas de controle usadas neste trabalho em um veículo menos sensível a perturbações externas;
2. não foi possível avaliar o desempenho dos controladores em relação à posição e orientação espacial. Uma alternativa para obter essa informação é usar fusão dos dados da IMU com informações visuais para estimar a postura do veículo. Bourquardez *et al.* (2009) desenvolveram um trabalho que usa essa técnica com quadricópteros;
3. o processo de sintonia dos controladores foi empírico. Os valores obtidos nas simulações serviram como ponto de partida para a aplicação no AR.Drone. Para o caso variável isso foi particularmente difícil, já que requer a configuração de três parâmetros. Seria interessante comparar o desempenho dos controladores sintonizados por meio de alguma outra técnica conhecida;
4. algumas técnicas de controle avançado vêm sendo usadas para controle servo visual. Dentre suas vantagens está a possibilidade de tratar restrições tanto no espaço da tarefa e da imagem quanto dos atuadores. São os casos do controle ótimo

(DANÈS; BELLOT, 2006) e do controle servo visual preditivo (ALLIBERT; COUR-TIAL; CHAUMETTE, 2010), por exemplo, além de outras citadas no Capítulo 1. Tais técnicas devem evitar que as tarefas servo visuais falhem em decorrência da perda de características na imagem, como ocorreu em alguns dos experimentos realizados neste trabalho.

# Referências Bibliográficas

ALLIBERT, G.; COURTIAL, E.; CHAUMETTE, F. Predictive Control for Constrained Image-Based Visual Servoing. **IEEE Transactions on Robotics**, v. 26, n. 5, p. 933–939, out. 2010.

ALLIBERT, G.; COURTIAL, E.; TOURÉ, Y. Visual predictive control for manipulators with catadioptric camera. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2008. p. 510–515.

AZINHEIRA, J.; RIVES, P.; CARVALHO, J.; SILVEIRA, G.; PAIVA, E. de; BUENO, S. Visual servo control for the hovering of all outdoor robotic airship. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2002. v. 3, n. May, p. 2787–2792.

BAJRACHARYA, M.; MAIMONE, M.; HELMICK, D. Autonomy for Mars Rovers: Past, Present, and Future. **Computer**, v. 41, n. 12, p. 44–50, dez. 2008.

BAKER, C.; DOLAN, J. Street smarts for boss. **IEEE Robotics & Automation Magazine**, v. 16, n. 1, p. 78–87, mar. 2009.

BENHIMANE, S.; MALIS, E. Homography-based 2D Visual Tracking and Servoing. **The International Journal of Robotics Research**, v. 26, n. 7, p. 661–676, jul. 2007.

BENSALAH, F. **Estimation du mouvement par vision active**. 149 p. Tese (Doutorado em Tratamento de Sinais e Telecomunicações) — L'Université de Rennes I, Rennes, 1996.

BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual Navigation for Mobile Robots: A Survey. **Journal of Intelligent and Robotic Systems**, v. 53, n. 3, p. 263–296, maio 2008.

BOURQUARDEZ, O.; CHAUMETTE, F. Visual servoing of an airplane for auto-landing. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, Ieee, p. 1314–1319, out. 2007.

BOURQUARDEZ, O.; MAHONY, R.; GUENARD, N.; CHAUMETTE, F.; HAMEL, T.; ECK, L. **Kinematic Visual Servo Control of a Quadrotor aerial vehicle**. Rennes, 2007. 38 p.

BOURQUARDEZ, O.; MAHONY, R.; GUENARD, N.; CHAUMETTE, F.; HAMEL, T.; ECK, L. Image-Based Visual Servo Control of the Translation Kinematics of a Quadrotor Aerial Vehicle. **IEEE Transactions on Robotics**, v. 25, n. 3, p. 743–749, 2009.

- BRISTEAU, P.-J.; CALLOU, F.; VISSIÈRE, D.; PETIT, N. The Navigation and Control Technology Inside the AR.Drone Micro UAV. In: SERGIO, B. (Ed.). **IFAC World Congress**. Milão: [s.n.], 2011. p. 1477–1484.
- CEREN, Z.; ALTUĞ, E. Image Based and Hybrid Visual Servo Control of an Unmanned Aerial Vehicle. **Journal of Intelligent and Robotic Systems**, v. 65, n. 1-4, p. 325–344, set. 2011.
- CHAUMETTE, F. Potential problems of stability and convergence in image-based and position-based visual servoing. In: KRIEGMAN, D.; HAGER, G.; MORSE, A. (Ed.). **The confluence of vision and control**. Springer London, 1998, (Lecture Notes in Control and Information Sciences, v. 237). cap. 5, p. 66–78.
- CHAUMETTE, F. Image Moments: A General and Useful Set of Features for Visual Servoing. **IEEE Transactions on Robotics**, v. 20, n. 4, p. 713–723, ago. 2004.
- CHAUMETTE, F.; HUTCHINSON, S. Visual servo control. I. Basic approaches. **IEEE Robotics & Automation Magazine**, v. 13, n. 4, p. 82–90, dez. 2006.
- CHAUMETTE, F.; HUTCHINSON, S. Visual servo control. II. Advanced approaches [Tutorial]. **IEEE Robotics & Automation Magazine**, v. 14, n. 1, p. 109–118, mar. 2007.
- CHAUMETTE, F.; RIVES, P.; ESPIAU, B. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. In: **International Conference on Robotics and Automation**. IEEE Comput. Soc. Press, 1991. p. 2248–2253.
- CORKE, P. MATLAB toolboxes: robotics and vision for students and teachers. **IEEE Robotics & Automation Magazine**, v. 14, n. 4, p. 16–17, dez. 2007.
- CORKE, P. **Robotics, Vision and Control**. Berlim, Heidelberg: Springer Berlin Heidelberg, 2011. 1–31 p. (Springer Tracts in Advanced Robotics, v. 73).
- CORKE, P. I. Visual Control Of Robot Manipulators — A Review. In: K. Hashimoto (Ed.). **Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback**. Singapore: World Scientific, 1993, (World Scientific Series in Robotics and Automated Systems). cap. 1, p. 1–31.
- CORKE, P. I. Spherical image-based visual servo and structure estimation. In: **IEEE International Conference on Robotics and Automation**. Anchorage: IEEE, 2010. p. 5550–5555.
- CORKE, P. I.; HUTCHINSON, S. A. A new partitioned approach to image-based visual servo control. **IEEE Transactions on Robotics and Automation**, v. 17, n. 4, p. 507–515, 2001.
- DANÈS, P.; BELLOT, D. Towards an LMI Approach to Multicriteria Visual Servoing in Robotics. **European Journal of Control**, v. 12, n. 1, p. 86–110, abr. 2006.
- DE COSTA SOUSA, J.; SETNES, M. Fuzzy predictive filters in model predictive control. **IEEE Transactions on Industrial Electronics**, v. 46, n. 6, p. 1225–1232, 1999.
- DEMENTHON, D. F.; DAVIS, L. S. Model-based object pose in 25 lines of code. **International Journal of Computer Vision**, v. 15, p. 123–141, 1995.

- ESPIAU, B.; CHAUMETTE, F.; RIVES, P. A new approach to visual servoing in robotics. **IEEE Transactions on Robotics and Automation**, v. 8, n. 3, p. 313–326, jun. 1992.
- FEDDEMA, J.; MITCHELL, O. Vision-guided servoing with feature-based trajectory generation. **IEEE Transactions on Robotics and Automation**, v. 5, n. 5, p. 691–700, 1989.
- FOMENA, R. T.; CHAUMETTE, F. Visual Servoing from Spheres using a Spherical Projection Model. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2007. p. 2080–2085.
- GANS, N. R.; HUTCHINSON, S. A.; CORKE, P. I. Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control. **The International Journal of Robotics Research**, v. 22, n. 10, p. 955–981, out. 2003.
- GARCIA, E.; JIMENEZ, M.; De Santos, P.; ARMADA, M. The evolution of robotics research. **IEEE Robotics & Automation Magazine**, v. 14, n. 1, p. 90–103, mar. 2007.
- GONÇALVES, P.; MENDONÇA, L.; SOUSA, J.; CALDAS PINTO, J. Uncalibrated Eye-to-Hand Visual Servoing Using Inverse Fuzzy Models. **IEEE Transactions on Fuzzy Systems**, v. 16, n. 2, p. 341–353, abr. 2008.
- GROCHOLSKY, B.; KELLER, J.; KUMAR, V.; PAPPAS, G. Cooperative air and ground surveillance. **IEEE Robotics & Automation Magazine**, v. 13, n. 3, p. 16–25, set. 2006.
- GUENARD, N.; HAMEL, T.; MAHONY, R. A Practical Visual Servo Control for an Unmanned Aerial Vehicle. **IEEE Transactions on Robotics**, v. 24, n. 2, p. 331–340, abr. 2008.
- HASHIMOTO, K.; EBINE, T.; KIMURA, H. Visual servoing with hand-eye manipulator—Optimal control approach. **IEEE Transactions on Robotics and Automation**, v. 12, n. 5, p. 766–774, 1996.
- HASHIMOTO, K.; KIMOTO, T.; EBINE, T.; KIMURA, H. Manipulator control with image-based visual servo. In: **IEEE International Conference on Robotics and Automation**. IEEE Comput. Soc. Press, 1991. p. 2267–2271.
- HILL, J.; PARK, W. T. Real time control of a robot with a mobile camera. In: **International Symposium on Industrial Robots**. Washington: [s.n.], 1979. p. 233–246.
- HOSODA, K.; ASADA, M. Versatile visual servoing without knowledge of true Jacobian. In: **International Conference on Intelligent Robots and Systems**. IEEE, 1994. v. 1, p. 186–193.
- HUTCHINSON, S.; HAGER, G.; CORKE, P. I. A tutorial on visual servo control. **IEEE Transactions on Robotics and Automation**, v. 12, n. 5, p. 651–670, 1996.
- IWATSUKI, M.; OKIYAMA, N. A new formulation of visual servoing based on cylindrical coordinate system. **IEEE Transactions on Robotics**, v. 21, n. 2, p. 266–273, abr. 2005.

- JONES, J. Robots at the tipping point: the road to iRobot Roomba. **IEEE Robotics & Automation Magazine**, v. 13, n. 1, p. 76–78, mar. 2006.
- KATRAŠNIK, J.; PERNUŠ, F.; LIKAR, B. A Survey of Mobile Robots for Distribution Power Line Inspection. **IEEE Transactions on Power Delivery**, v. 25, n. 1, p. 485–493, jan. 2010.
- KRAJNÍK, T.; VONASEK, V.; FIŠER, D.; FAIGL, J. AR-Drone as a Platform for Robotic Research and Education. In: OBDRZALEK, D.; GOTTSCHIEBER, A. (Ed.). **Research and Education in Robotics - EUROBOT 2011**. Springer Berlin Heidelberg, 2011, (Communications in Computer and Information Science, v. 161). p. 172–186.
- KUMAGAI, J. Swimming to Europa. **IEEE Spectrum**, v. 44, n. 9, p. 33–40, 2007.
- LIPPIELLO, V.; SICILIANO, B.; VILLANI, L. Eye-in-Hand/Eye-to-Hand Multi-Camera Visual Servoing. In: **IEEE Conference on Decision and Control**. IEEE, 2005. p. 5354–5359.
- MAHONY, R.; KUMAR, V.; CORKE, P. Multirotor Aerial Vehicles: Modeling, Estimation and Control of Quadrotor. **IEEE Robotics & Automation Magazine**, v. 19, n. 3, p. 20–32, set. 2012.
- MALIS, E. Improving vision-based control using efficient second-order minimization techniques. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2004. v. 2, n. April, p. 1843–1848.
- MALIS, E.; CHAUMETTE, F. 2 1/2 D Visual Servoing with Respect to Unknown Objects Through a New Estimation Scheme of Camera Displacement. **International Journal of Computer Vision**, v. 37, n. 1, p. 79–97, 2000.
- MALIS, E.; CHAUMETTE, F.; BOUDET, S. 2 1/2 D visual servoing. **IEEE Transactions on Robotics and Automation**, v. 15, n. 2, p. 238–250, abr. 1999.
- MALIS, E.; CHAUMETTE, F.; BOUDET, S. Multi-cameras visual servoing. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2000. v. 4, n. April, p. 3183–3188.
- MARCHAND, E.; CHAUMETTE, F. Features tracking for visual servoing purpose. In: KRAGIC, D.; CHRISTENSEN, H. (Ed.). **Advances in Robot Vision - From Domestic Environments to Medical Applications**. Sendai: [s.n.], 2004. p. 10–20.
- MARCHAND, E.; SPINDLER, F.; CHAUMETTE, F. ViSP for visual servoing: a generic software platform with a wide class of robot control skills. **IEEE Robotics & Automation Magazine**, v. 12, n. 4, p. 40–52, dez. 2005.
- MARIOTTINI, G. L.; PRATTICCHIZZO, D. Image-based Visual Servoing with Central Catadioptric Cameras. **The International Journal of Robotics Research**, v. 27, n. 1, p. 41–56, jan. 2008.
- MATARIĆ, M. J. **The Robotics Primer**. Cambridge: The MIT Press, 2007. 328 p.
- MICHEL, H.; RIVES, P. **Singularities in the determination of the situation of a robot effector from the perspective view of 3 points**. Sophia-Antipolis, 1993. 22 p.



- MIURA, K.; HASHIMOTO, K.; GANGLOFF, J.; MATHELIN, M. de. Visual Servoing without Jacobian Using Modified Simplex Optimization. In: **IEEE International Conference on Robotics and Automation**. Barcelona: IEEE, 2005. p. 3504–3509.
- MORARI, M. **Visual Servoing via Advanced Numerical Methods**. Londres: Springer London, 2010. (Lecture Notes in Control and Information Sciences, v. 401).
- PAPANIKOLOPOULOS, N.; KHOSLA, P.; KANADE, T. Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision. **IEEE Transactions on Robotics and Automation**, v. 9, n. 1, p. 14–35, 1993.
- PISKORSKI, S.; BRULEZ, N.; ELINE, P.; D’HAEYER, F. **AR.Drone Developer Guide**. [S.l.], 2012. 105 p.
- PLINVAL, H. de; MORIN, P.; MOUYON, P.; HAMEL, T. Visual servoing for underactuated VTOL UAVs: A linear, Homography-Based Approach. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2011. p. 3004–3010.
- SAMSON, C.; Le BORGNE, M.; ESPIAU, B. **Robot Control: The task function approach**. 1. ed. [S.l.]: Oxford University Press, 1991. 364 p. (Oxford Engineering Science Series, v. 22).
- SANTANA, P. H. d. R. Q. e. A.; BORGES, G. A. Modelagem e Controle de Quadrirotóres. In: **Simpósio Brasileiro de Automação Inteligente**. Brasília: [s.n.], 2009.
- SARIPALLI, S.; MONTGOMERY, J.; SUKHATME, G. Visually guided landing of an unmanned aerial vehicle. **IEEE Transactions on Robotics and Automation**, v. 19, n. 3, p. 371–380, jun. 2003.
- SEQUEIRA GONÇALVES, P.; MENDONÇA, L.; SOUSA, J.; CALDAS PINTO, J. Improving visual servoing using fuzzy filters. In: **IEEE International Conference on Fuzzy Systems**. Budapeste: IEEE, 2004. v. 2, p. 1185–1190.
- SHARMA, R.; HUTCHINSON, S. Motion perceptibility and its application to active vision-based servo control. **IEEE Transactions on Robotics and Automation**, v. 13, n. 4, p. 607–617, 1997.
- SHI, J.; TOMASI, C. Good features to track. In: **IEEE Conference on Computer Vision and Pattern Recognition**. IEEE Comput. Soc. Press, 1994. p. 593–600.
- SHIRAI, Y.; INOUE, H. Guiding a robot by visual feedback in assembling tasks. **Pattern Recognition**, v. 5, n. 2, p. 99–108, jun. 1973.
- SICILIANO, B.; SCIAVICCO, L.; VILLANI, L.; ORIOLO, G. **Robotics: Modelling, Planning and Control**. Londres: Springer London, 2009. (Advanced Textbooks in Control and Signal Processing).
- SILVEIRA, G.; CARVALHO, J.; RIVES, P.; AZINHEIRA, J.; BUENO, S.; MADRID, M. Optimal visual servoed guidance of outdoor autonomous robotic airships. In: **American Control Conference**. American Automatic Control Council, 2002. v. 1, p. 779–784.

- SILVEIRA, G.; MALIS, E. Direct Visual Servoing: Vision-Based Estimation and Control Using Only Nonmetric Information. **IEEE Transactions on Robotics**, v. 28, n. 4, p. 974–980, 2012.
- SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. Wiley, 2005. 496 p.
- SZELISKI, R. **Computer Vision: Algorithms and Applications**. Nova Iorque: Springer, 2011. (Texts in Computer Science).
- TAHRI, O.; CHAUMETTE, F. Point-based and region-based image moments for visual servoing of planar objects. **IEEE Transactions on Robotics**, v. 21, n. 6, p. 1116–1127, dez. 2005.
- THUILOT, B.; MARTINET, P.; CORDESSES, L.; GALLICE, J. Position based visual servoing: keeping the object in the field of vision. In: **IEEE International Conference on Robotics and Automation**. IEEE, 2002. v. 2, n. May, p. 1624–1629.
- TSAKIRIS, D. P.; RIVES, P.; SAMSON, C. Extending visual servoing techniques to nonholonomic mobile robots. In: KRIEGMAN, D.; HAGER, G.; MORSE, A. (Ed.). **The confluence of vision and control**. Springer London, 1998, (Lecture Notes in Control and Information Sciences, v. 237). p. 106–117.
- VENABLES, M. Let intelligent robots take the strain [automobile industry]. **Manufacturing Engineer**, v. 85, n. 2, p. 34–35, 2006.
- WEISS, L.; SANDERSON, A.; NEUMAN, C. Dynamic visual servo control of robots: An adaptive image-based approach. In: **IEEE International Conference on Robotics and Automation**. Institute of Electrical and Electronics Engineers, 1985. v. 2, p. 662–668.
- WEISS, L.; SANDERSON, A.; NEUMAN, C. Dynamic sensor-based control of robots with visual feedback. **IEEE Journal on Robotics and Automation**, v. 3, n. 5, p. 404–417, out. 1987.
- WESTMORE, D.; WILSON, W. Direct dynamic control of a robot using an end-point mounted camera and Kalman filter position estimation. In: **IEEE International Conference on Robotics and Automation**. IEEE Comput. Soc. Press, 1991. p. 2376–2384.
- WILSON, W.; WILLIAMS HULLS, C.; BELL, G. Relative end-effector control using Cartesian position based visual servoing. **IEEE Transactions on Robotics and Automation**, v. 12, n. 5, p. 684–696, 1996.
- YOSHIHATA, Y.; WATANABE, K.; IWATANI, Y.; HASHIMOTO, K. Multi-camera visual servoing of a micro helicopter under occlusions. In: **IEEE/RSJ International Conference on Intelligent Robots and Systems**. IEEE, 2007. p. 2615–2620.

# Apêndice A - Representação de corpos no espaço e seu movimento

Sistemas robóticos costumam ser tratados como um corpo rígido. Denominam-se sua posição e orientação no espaço euclidiano por postura. O número mínimo de coordenadas para representar um corpo desta forma é seis: três para posição e três para orientação. A cada corpo associa-se um sistema de coordenadas fixo em relação a ele. Os deslocamentos, tanto angulares quanto lineares, são expressos como relações entre dois sistemas de coordenadas — um deles comumente referido como *fixo* e o outro, *móvel*. Isso significa apenas que o observador é estacionário em relação ao primeiro. Cada sistema de coordenadas  $\mathcal{F}_i$  consiste de uma origem,  $\mathcal{O}_i$ , e três vetores unitários mutuamente ortonormais, denominados base e representados por  $[\hat{x}_i \ \hat{y}_i \ \hat{z}_i]^T$ .

## A.1 Representando posição

Representa-se um ponto em relação ao sistema de coordenadas  $i$  como

$${}^i\mathbf{p} = x\hat{\mathbf{x}}_i + y\hat{\mathbf{y}}_i + z\hat{\mathbf{z}}_i = \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} \hat{\mathbf{x}}_i \\ \hat{\mathbf{y}}_i \\ \hat{\mathbf{z}}_i \end{bmatrix} \quad (\text{A.1})$$

onde  $x$ ,  $y$  e  $z$  são as projeções do vetor entre  $\mathcal{O}_i$  e o ponto  $p$ .

Analogamente, representa-se a posição da origem de um sistema de coordenadas  $i$  em relação a  $j$  por

$${}^j\mathbf{t}_i = \begin{bmatrix} {}^j t_i^x \hat{\mathbf{x}}_j \\ {}^j t_i^y \hat{\mathbf{y}}_j \\ {}^j t_i^z \hat{\mathbf{z}}_j \end{bmatrix}. \quad (\text{A.2})$$

Os componentes desse vetor,  ${}^j t_i^x$ ,  ${}^j t_i^y$  e  ${}^j t_i^z$ , são as coordenadas cartesianas de  $\mathcal{O}_i$  em  $\mathcal{F}_j$ , que são as projeções do vetor  ${}^j\mathbf{t}_i$  nos eixos correspondentes.

## A.2 Representando orientação

Existem diversas formas de representar a orientação de um sistema de coordenadas em relação a outro. Aqui apenas duas são abordadas: a matriz de rotação e a representação com três ângulos.

### A.2.1 Matriz de rotação

A matriz de rotação expressa os vetores da base de  $\mathcal{F}_i$ ,  $[\hat{x}_i \ \hat{y}_i \ \hat{z}_i]^T$ , em relação à base de  $\mathcal{F}_j$ ,  $[\hat{x}_j \ \hat{y}_j \ \hat{z}_j]^T$ . Essa relação fornece uma matriz  $3 \times 3$  cujos componentes são os produtos escalares entre os vetores da base de cada sistema de coordenadas:

$${}^j\mathbf{R}_i = \begin{bmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{bmatrix} \quad (\text{A.3})$$

e

$${}^j\mathbf{p} = {}^j\mathbf{R}_i {}^i\mathbf{p}. \quad (\text{A.4})$$

Na prática, essa matriz é bastante simplificada, já que os vetores da base são unitários e seus produtos escalares são o cosseno do ângulo formado entre eles. Para uma rotação de  $i$  em relação a  $\hat{x}_j$  por um ângulo  $\theta$ , por exemplo,

$$\mathbf{R}_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}. \quad (\text{A.5})$$

Contudo, a matriz de rotação possui nove elementos, enquanto apenas três são necessários para representar a orientação no espaço tridimensional. A representação com três ângulos fornece uma descrição mínima.

### A.2.2 Representação com três ângulos

Esse método, que também recebe o nome de ângulos de Euler, usa a rotação em relação a três eixos para representar a orientação. Os eixos de rotação podem ser

repetidos, mas não sucessivamente. Por exemplo, a representação  $XYX$  significa uma rotação em relação ao eixo  $\hat{x}$  seguida por outra em relação a  $\hat{y}$  e mais uma em relação a  $\hat{x}$ . Há, ainda, outra distinção. Essas rotações podem ser realizadas em relação ao eixo atual ou a um eixo fixo. No primeiro caso, faz-se uma rotação subsequente em relação à nova posição que o sistema de coordenadas se encontra. No segundo, todas as rotações são feitas em relação a um mesmo sistema de coordenadas fixo.

Existem doze tipos distintos de representação para o caso em relação ao eixo atual e mais doze em relação a um eixo fixo. A sequência  $XYZ$  costuma ser chamada de *rolagem-arfagem-guinada* (*roll-pitch-yaw*). Contudo, não há uma única convenção em relação a quais rotações recebem cada nome. Por exemplo, Spong, Hutchinson e Vidyasagar (2005) denomina a sequência  $XYZ$  de guinada-arfagem-rolagem, enquanto (CORKE, 2011) adota rolagem-arfagem-guinada. Esta última é a adotada nesse texto e usada em todas as representações de orientação.

### A.2.3 Postura - combinando translação e rotação

A postura de um corpo é, então, a combinação da sua translação e sua rotação. Contudo, é de se esperar que, com as sucessivas rotações, as equações de rotação, como (A.5) fiquem bem complexas. Uma forma de representar ambos deslocamentos de forma compacta, em uma matriz  $4 \times 4$ , é usar a representação homogênea. Assim, qualquer postura pode ser representada como

$${}^j\mathbf{T}_i = \begin{bmatrix} {}^j\mathbf{R}_i & {}^j\mathbf{t}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}. \quad (\text{A.6})$$

A matriz  ${}^j\mathbf{T}_i$  recebe o nome de matriz de transformação homogênea. Assim,

$$\begin{bmatrix} {}^j\mathbf{p} \\ 1 \end{bmatrix} = {}^j\mathbf{T}_i \begin{bmatrix} {}^i\mathbf{p} \\ 1 \end{bmatrix}. \quad (\text{A.7})$$

### A.3 Velocidade de um ponto fixo em um sistema de coordenadas em movimento

Seja  $\mathbf{S}$  uma matriz antisimétrica de dimensões  $3 \times 3$  e  $\mathbf{a} = [a_1 \ a_2 \ a_3]^T$ . Defina a matriz matriz antisimétrica  $\mathbf{S}(\mathbf{a})$  como

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (\text{A.8})$$

É possível provar (SPONG; HUTCHINSON; VIDYASAGAR, 2005) que, dada uma matriz de rotação variante no tempo  $\mathbf{R}(t)$  e um sistema de coordenadas que se move com velocidade angular  $\omega(t)$ ,

$$\dot{\mathbf{R}}(t) = \mathbf{S}(\omega(t))\mathbf{R}(t) \quad (\text{A.9})$$

e que (removendo a dependência temporal, para simplificar a notação)

$$\mathbf{S}(\omega) \mathbf{p} = \omega \times \mathbf{p}. \quad (\text{A.10})$$

Seja  ${}^i\mathbf{p}$  um ponto rigidamente fixo ao sistema de coordenadas  $\mathcal{F}_i$ , reescrevendo A.7,

$${}^j\mathbf{p} = {}^j\mathbf{R}_i {}^i\mathbf{p} + {}^j\mathbf{t}_i, \quad (\text{A.11})$$

cuja derivada temporal é

$${}^j\dot{\mathbf{p}} = {}^j\dot{\mathbf{R}}_i {}^i\mathbf{p} + {}^j\mathbf{R}_i {}^i\dot{\mathbf{p}} + {}^j\dot{\mathbf{t}}_i. \quad (\text{A.12})$$

Lembrando que o ponto  ${}^i\mathbf{p}$  é fixo em relação a  $\mathcal{F}_i$ , substituindo (A.9) e (A.10),

$${}^j\dot{\mathbf{p}} = \mathbf{S}({}^j\boldsymbol{\omega}_{j,i}) {}^j\mathbf{R}_i {}^i\mathbf{p} + {}^j\dot{\mathbf{t}}_i \quad (\text{A.13})$$

$$= {}^j\boldsymbol{\omega}_{j,i} \times {}^j\mathbf{R}_i {}^i\mathbf{p} + {}^j\dot{\mathbf{t}}_i \quad (\text{A.14})$$

$$= {}^j\boldsymbol{\omega}_{j,i} \times \mathbf{r} + {}^j\mathbf{v}_{j,i}, \quad (\text{A.15})$$

onde  $\mathbf{r} = {}^j\mathbf{R}_i {}^i\mathbf{p}$  é o vetor da origem de  $\mathcal{F}_i$  a  $\mathbf{p}$  expresso em relação a  $\mathcal{F}_j$ ,  $\mathbf{v}$  e  $\boldsymbol{\omega}_{j,i}$  são as velocidades linear e angular do sistema de coordenadas  $\mathcal{F}_i$  em relação a  $\mathcal{F}_j$  expressas em  $\mathcal{F}_j$ .



# Apêndice B - Calibração de câmeras

## B.1 Modelagem da câmera

A relação (2.1) pode ser escrita na forma matricial como

$$\tilde{\mathbf{p}} = \mathbf{M}\tilde{\mathbf{P}} \quad (\text{B.1})$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (\text{B.2})$$

em que  $(\tilde{\cdot})$  indica coordenadas homogêneas. Pode ainda ser fatorado em

$$\tilde{\mathbf{p}} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{P}}. \quad (\text{B.3})$$

É possível, ainda, incorporar a Equação 2.2 e obter a posição em píxeis do ponto na imagem.

$$\tilde{\mathbf{m}} = \underbrace{\begin{bmatrix} \frac{f}{\rho_l} & 0 & u_0 \\ 0 & \frac{f}{\rho_a} & u_v \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\mathbf{P}}. \quad (\text{B.4})$$

A matriz  $\mathbf{K}$  é conhecida como matriz de parâmetros intrínsecos da câmera,

$$\mathbf{K} = \begin{bmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.5})$$

Contudo, o modelo de projeção perspectiva mostrado na Seção 2.1 considera uma câmera perfeita. Na prática, verifica-se que, em decorrência de imperfeições na fabricação e montagem das lentes e do sensor, as imagens apresentam distorções. Tais imperfeições podem ser modeladas como

$$\begin{cases} u = u_0 + p_x x (1 + k_{ud} (x^2 + y^2)) \\ v = v_0 + p_y y (1 + k_{ud} (x^2 + y^2)), \end{cases} \quad (\text{B.6})$$

onde  $k_{ud}$  é um parâmetro de distorção radial usado para converter unidades normalizadas para píxeis.

O problema inverso, determinar as coordenadas normalizadas a partir dos valores em píxeis é resolvido com

$$\begin{cases} x = \frac{\bar{u}}{p_x} \left( 1 + k_{du} \left( \left( \frac{\bar{u}}{p_x} \right)^2 + \left( \frac{\bar{v}}{p_y} \right)^2 \right) \right) \\ y = \frac{\bar{v}}{p_y} \left( 1 + k_{du} \left( \left( \frac{\bar{u}}{p_x} \right)^2 + \left( \frac{\bar{v}}{p_y} \right)^2 \right) \right), \end{cases} \quad (\text{B.7})$$

com  $k_{du} \neq k_{ud}$  também um parâmetro de distorção radial, neste caso, usado para transformar unidades de píxeis para metros.

Existem outros modelos de representação das distorções causadas pelas câmeras. Alguns deles incorporam também distorções tangenciais. Porém, na prática, estas últimas costumam ser bem menores que as primeiras (CORKE, 2011).

O processo de determinação da matriz de parâmetros intrínsecos  $\mathbf{K}$  e dos parâmetros de distorção é conhecido como calibração da câmera. Para detalhes sobre algumas dessas abordagens, ver Szeliski (2011) e Corke (2011).