

**CONTROLE SERVO-VISUAL PARA APROXIMAÇÃO DE PORTAS
POR ROBÔS MÓVEIS EQUIPADOS COM DUAS CÂMERAS**

MARIANA COSTA BERNARDES

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA**

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**CONTROLE SERVO-VISUAL PARA APROXIMAÇÃO DE PORTAS
POR ROBÔS MÓVEIS EQUIPADOS COM DUAS CÂMERAS**

Mariana Costa Bernardes

*Manuscrito submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Mestre em Engenharia Elétrica*

Banca Examinadora

Prof. Geovany Araújo Borges, ENE/UnB
Orientador

Prof. Geraldo Figueiredo da Silveira Filho,
CTI/MCT
Examinador externo

Prof. João Yoshiyuki Ishihara, ENE/UnB
Examinador interno

Prof. Adolfo Bauchspiess, ENE/UnB
Suplente

Para Tiago

AGRADECIMENTOS

Quero agradecer às pessoas que de alguma forma se fizeram presentes ao longo destes três anos de mestrado. E não só aos que me ajudaram efetivamente na construção dessa Dissertação, mas também aos amigos e colegas que compartilharam comigo idéias, fomentaram discussões, que se preocuparam, que foram solidários, que torceram por mim. Foram eles que fizeram possível este momento e por isso expresso a todos minha mais profunda gratidão.

Ao meu orientador e amigo, Prof. Geovany Araújo Borges, agradeço pela confiança depositada em meu trabalho e principalmente pelo constante incentivo, sempre indicando a direção a ser tomada a cada obstáculo encontrado.

Agradeço também aos colegas do Laboratório de Robótica e Automação, que sempre demonstraram amizade, otimismo e total disponibilidade para ajudar no que fosse preciso. De forma especial, meu muito obrigado à toda família Beckmann.

À minha família, cujo afeto incondicional e apoio nos momentos difíceis me deram forças para seguir adiante, dedico poucas palavras porém aquelas que me são mais caras. Obrigada por existirem!

Em especial, gostaria de agradecer também ao Tiago que por vezes deve ter detestado a mim e a este trabalho devido aos vários momentos sacrificados e ausências constantes. Agradeço carinhosamente pelo companheirismo, paciência e compreensão.

Por fim, agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por tornar possível a realização deste trabalho através do financiamento concedido.

RESUMO

CONTROLE SERVO-VISUAL PARA APROXIMAÇÃO DE PORTAS POR ROBÔS MÓVEIS EQUIPADOS COM DUAS CÂMERAS

Autor: Mariana Costa Bernardes

Orientador: Prof. Geovany Araújo Borges, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Brasília, abril de 2009

Classicamente, o controle servo visual é aplicado em abordagens determinísticas que fazem uso direto dos parâmetros extraídos da imagem ou que os utilizam para determinar a posição do robô no plano cartesiano e com isso calcular a ação de controle. Tais métodos, no entanto, são bastante suscetíveis a ruídos de medição e pouco tolerantes a falhas como obstrução ou perda do rastreamento das características visuais. Estes problemas poderiam ser minimizados com uma abordagem estocástica, fazendo-se uso do modelo de evolução das características visuais e de informação sensorial adicional para estimar os valores de tais parâmetros. Este trabalho busca obter uma abordagem de controle servo visual baseado em retas combinado a um processo simultâneo de reconstrução tridimensional por Filtro de Kalman Estendido. Propõe-se então um controlador para robôs móveis omnidirecionais, que utiliza um sistema binocular de câmeras como principal fonte de informação, associado a medidas de odometria fornecidas pelo robô para estimar o modelo das retas no espaço 3D. As estimativas obtidas são utilizadas no cálculo da ação de controle servo visual e com isso busca-se atenuar os problemas apresentados acima, presentes nas abordagens usuais. O método proposto é validado sobre um robô móvel omnidirecional numa tarefa de aproximação de portas e seu desempenho é comparado ao de um controlador cujas características são extraídas diretamente das imagens.

ABSTRACT

VISUAL SERVOING FOR DOCKING OF MOBILE ROBOTS EQUIPPED WITH TWO CAMERAS

Author: Mariana Costa Bernardes

Supervisor: Prof. Geovany Araújo Borges, ENE/UnB

Programa de Pós-graduação em Engenharia Elétrica

Brasília, april of 2009

Classically, visual servoing is applied in deterministic approaches that use parameters extracted directly from images or that use these parameters to calculate the robot's position in cartesian plane to define control action. However, these methods are very sensitive to measurement noise and not tolerant to fail situations like lost of tracking or obstruction of visual features. Such problems could be minimized with a stochastic approach, using the evolution model of image features and additional sensor information to estimate the parameters. This work purposes a line based visual servoing approach combined with a simultaneous tridimensional reconstruction process with Extended Kalman Filter. We propose a controller for omnidirectional mobile robots that use a binocular camera system as the main information source, combined with odometry measurements provided by the robot to estimate the 3D line model. The control signal is calculated with the obtained estimates in order to attenuate the problems presented above, observed in usual visual servoing approaches. The proposed method is evaluated on an omnidirectional mobile robot in a docking task and its performance is compared to a controller whose visual features are extracted directly from images.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.1.1	HISTÓRICO DO CONTROLE SERVO-VISUAL	1
1.1.2	VISÃO EM ROBÓTICA MÓVEL	2
1.1.3	NAVEGAÇÃO DE ROBÔS MÓVEIS E O PROBLEMA DE <i>docking</i>	4
1.2	MOTIVAÇÃO	6
1.2.1	ROBÓTICA MÓVEL NO LARA	6
1.2.2	CONTROLE SERVO-VISUAL DE TRAJETÓRIA E MAPEAMENTO LOCAL SIMUL- TÂNEOS	7
1.3	DEFINIÇÃO DO PROBLEMA E OBJETIVOS	7
1.4	RESULTADOS ALCANÇADOS	9
1.5	APRESENTAÇÃO DO MANUSCRITO	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	CONTROLE SERVO-VISUAL	11
2.1.1	CLASSIFICAÇÃO DOS TIPOS DE CONTROLE SERVO-VISUAL	11
2.1.2	CARACTERÍSTICAS DA IMAGEM	14
2.1.3	CONTROLADOR SERVO-VISUAL	15
2.2	REPRESENTAÇÃO DE OBJETOS NA IMAGEM	17
2.2.1	PONTOS E RETAS NO ESPAÇO	18
2.2.2	PONTOS E RETAS NA IMAGEM	20
2.2.3	MODELO GEOMÉTRICO DE PROJEÇÃO	21
2.2.4	PROJEÇÃO DA IMAGEM	22
3	CONTROLE SERVO-VISUAL PARA APROXIMAÇÃO DE PORTAS	25
3.1	ESPAÇOS \mathcal{F} E \mathcal{T} PARA O DOCKING DE PORTAS	25
3.2	SISTEMAS DE COORDENADAS \mathcal{O}^c , \mathcal{O}^d E \mathcal{O}^r	26
3.3	CONTROLE SERVO-VISUAL DE VELOCIDADE	29
3.3.1	CONTROLE <i>IBVS</i>	30
3.3.2	CONTROLE <i>PBVS</i>	31
3.4	ESTIMAÇÃO DAS COORDENADAS 3D DAS RETAS	33
3.5	OBTENÇÃO DAS OBSERVAÇÕES VISUAIS	35
3.6	EXTRAÇÃO E RASTREAMENTO DAS RETAS NA IMAGEM	39
4	RESULTADOS EXPERIMENTAIS	41
4.1	INTRODUÇÃO	41
4.2	SIMULAÇÃO	41
4.2.1	ESTIMAÇÃO DOS PARÂMETROS DA PORTA	42
4.2.2	CONTROLE SERVO-VISUAL	45

4.2.3	CONTROLE SERVO-VISUAL E ESTIMAÇÃO SIMULTÂNEOS.....	48
4.3	VALIDAÇÃO EXPERIMENTAL NA PLATAFORMA OMNI	49
4.3.1	SOMENTE ESTIMAÇÃO DOS PARÂMETROS DA PORTA	49
4.3.2	CONTROLE SERVO-VISUAL <i>IBVS</i>	50
4.3.3	CONTROLE SERVO-VISUAL <i>PBVS</i>	51
5	CONCLUSÕES	61
5.1	PROPOSTAS PARA TRABALHOS FUTUROS	62
	REFERÊNCIAS BIBLIOGRÁFICAS	63
	ANEXOS	69
A.	CONFIGURAÇÃO DE CORPOS RÍGIDOS E TRANSFORMAÇÕES DE COOR- DENADAS	71
B.	PRODUTO VETORIAL E SUAS PROPRIEDADES	75
C.	DECOMPOSIÇÃO EM VALORES SINGULARES PARA RESOLUÇÃO DE EQUA- ÇÕES LINEARES HOMOGÊNEAS	79
D.	CONFIGURAÇÕES DE TESTES UTILIZADOS NOS EXPERIMENTOS	81

LISTA DE FIGURAS

1.1	Exemplos de aplicações de controle servo-visual na robótica	3
1.2	Exemplos de trabalhos em <i>docking</i> servo visual	5
1.3	Plataforma móvel Omni e seus componentes.....	6
1.4	Esquemático do problema de <i>docking</i> proposto.....	8
1.5	Principais elementos do sistema proposto.	9
2.1	Tipos de hierarquia de controle servo-visual.....	12
2.2	Tipos de configuração de câmera.	13
2.3	Tipos de estratégia de controle servo-visual	14
2.4	Exemplo de características da imagem para o caso bidimensional (a) e tridimensional (b). Em ambos, s representa o vetor de características da imagem.....	15
2.5	Possíveis escolhas para \hat{L}_s^\dagger e comportamento correspondente numa aplicação <i>IBVS</i> : pontos brancos representam as posições iniciais enquanto pontos e linhas cinzas representam, respectivamente, a posição desejada e uma possível trajetória.	17
2.6	Representação da reta 3D utilizando coordenadas de Plücker.	19
2.7	Determinação da profundidade da porta d	20
2.8	Modelo para geometria da câmera	22
2.9	Esquemático da projeção de um ponto na imagem.....	22
3.1	Diagrama em blocos da solução proposta.	26
3.2	Relação entre os sistemas de coordenadas do robô e das câmeras.....	27
3.3	Retas extraídas de imagem da câmera esquerda e resultado correspondente na etapa de rastreamento. (a) As retas em verde representam todas as retas extraídas da imagem. (b) Os retângulos azuis representam as regiões de busca e as duas retas azuis são o resultado do rastreamento.	39
4.1	Ambiente de simulação em Matlab.....	41
4.2	Imagens simuladas para as câmeras durante trajetória retilínea com uma das retas posicionada sempre ao centro da imagem da câmera esquerda. As linhas azuis correspondem às projeções das laterais da porta enquanto as linhas verdes são projeções da estimação.	43
4.3	Erros de estimação das laterais esquerda (a) e direita (b) da porta em simulação do método monocular.....	44
4.4	Erros de estimação das laterais esquerda (a) e direita (b) da porta em simulação do método binocular.	44
4.5	Trajetórias e sinais de controle gerados pelo controlador <i>PBVS</i> partindo respectivamente das posições 1, 2 e 3 descritas na Figura D.1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.....	46

4.6	Trajetórias e sinais de controle gerados pelo controlador <i>IBVS</i> partindo da posição 4 descritas na Figura D..1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.....	47
4.7	Trajetórias e sinais de controle gerados pelo controlador <i>IBVS</i> com $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_{s^*}^\dagger)$ partindo respectivamente das posições 1, 2 e 3 descritas na Figura D..1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.	48
4.8	Seqüência de imagens da câmera esquerda. As linhas em magenta correspondem à projeção da estimativa atual do EKF.	50
4.9	Resultado do experimento de estimação com robô guiado por <i>joystick</i> . (a)Estimação dos parâmetros 3D da lateral esquerda da porta. (b)Estimação dos parâmetros 3D da lateral direita da porta.	50
4.10	Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle <i>IBVS</i> simultâneos partindo da posição 1 descrita na Figura D..2. As linhas verdes são as laterais esquerda (L) e direita (R) da porta e as azuis são a referência de controle.....	53
4.11	Resultados do experimento de estimação e controle <i>IBVS</i> simultâneos partindo da posição 1 descrita da Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) Erro de controle referente aos parâmetros a, b e c da projeção da reta esquerda em ambas as câmeras. (f) Erro de controle referente aos parâmetros a, b e c da projeção da reta direita em ambas as câmeras. $t_1 = 5,41s$ corresponde ao instante de início da ação de controle.	53
4.12	Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle <i>IBVS</i> simultâneos partindo da posição 2 descrita na Figura D..2. As linhas verdes são as laterais esquerda (L) e direita (R) da porta e as azuis são a referência de controle.....	54
4.13	Resultados do experimento de estimação e controle <i>IBVS</i> simultâneos partindo da posição 2 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) Erro de controle referente aos parâmetros a, b e c da projeção da reta esquerda em ambas as câmeras. (f) Erro de controle referente aos parâmetros a, b e c da projeção da reta direita em ambas as câmeras. $t_1 = 18,32$ corresponde ao instante de início da ação de controle.	54
4.14	Situação em que o controle <i>IBVS</i> converge para a referência no espaço da imagem com posição diferente da desejada para a <i>docking</i> . (a) Robô com pequeno erro final de orientação. (c) e (d) Imagens das câmeras. As retas plotadas correspondem às referência de controle. No caso, ambas coincidem com as laterais da porta apesar de o robô estar ligeiramente fora do alvo de <i>docking</i>	55
4.15	Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 1 descrita na Figura D..2. As linhas em magenta são as projeções da estimativa e as linhas azuis são a projeção da referência de controle.....	55

4.16	Resultados do experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 1 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 12,02$ corresponde ao instante de início da ação de controle.....	56
4.17	Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 2 descrita na Figura D..2. As linhas em magenta são as projeções da estimativa e as linhas azuis são a projeção da referência de controle.....	56
4.18	Resultados do experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 2 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 6,03$ corresponde ao instante de início da ação de controle.....	57
4.19	Seqüência de pares de imagens durante o experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 3 descrita na Figura D..2. As linhas em magenta são as projeções das estimativas e as linhas azuis são a projeção da referência de controle. (a) Início da ação de controle. (b) Instante t_2 em que uma das laterais sai do campo de visão da câmera direita. (c) Instante t_3 em que a reta perdida volta a ser rastreada. (d) Imagens ao fim do controle.....	58
4.20	Resultados do experimento de estimação e controle <i>PBVS</i> simultâneos partindo da posição 3 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 9,61$ corresponde ao instante de início da ação de controle. $t_2 = 33,12$ corresponde ao instante em que o <i>tracking</i> perdeu a reta direita. $t_3 = 91,03$ corresponde ao instante em que a reta direita volta a ser rastreada.	59
A..1	Representação de um sistema coordenado global \mathcal{O}^w e um sistema coordenado local \mathcal{O}^o , atrelado a um corpo rígido.....	71
A..2	Representação de uma transformação de sistemas coordenados.	72
B..1	Representação geométrica do produto vetorial e do produto escalar triplo.....	76
D..1	Configurações iniciais e final para a postura do robô nas avaliações em simulação.....	82
D..2	Configurações iniciais e final para a postura do robô nas avaliações experimentais.	82

LISTA DE TABELAS

4.1	Estados Iniciais	45
4.2	Desvio padrão de medição	45
4.3	Desvio padrão de odometria	45
4.4	Resultados Finais da Estimação.....	50
D.1	Coordenadas e orientação das posições de testes com relação ao centro da porta	81
D.2	Coordenadas e orientação das posições dos experimentos com relação ao centro da porta	81

LISTA DE SÍMBOLOS

Símbolos Latinos

r	vetor de postura de robô	$[m]$ e $[rad]$
d	distância	$[m]$
p	coordenadas de um ponto	$[m]$
u	vetor de orientação da reta	$[m]$
h	vetor normal da reta	$[m]$
t	instante de tempo	$[s]$
T	período de amostragem	$[s]$
f	distância focal	$[m]$
D	fator de escala para projeção na imagem	$[pixel/m]$
v	vetor velocidade de translação	$[m/s]$
y	vetor de observações visuais	$[rad]$ e $[m]$
w	erro de observações visuais	$[rad]$ e $[m]$
r	erro de medição	$[pixel]$
q	erro de predição	$[m]$
s	erro de odometria	$[m/s]$ e $[rad/s]$

Símbolos Gregos

λ	coordenadas de Plücker de uma reta	$[m]$
θ	ângulo de orientação da reta	$[rad]$
β	ângulo de orientação da reta projetada	$[rad]$
ω	vetor velocidade de rotação	$[rad/s]$
ν	vetor de velocidades	$[m/s]$ e $[rad/s]$

Grupos Adimensionais

K	matriz de calibração da câmera
R	matriz de rotação
t	vetor de translação
s	vetor de características da imagem
f	parâmetros de características da imagem
e	erro de controle
L	matriz de interação
K	ganho do controlador
D	matriz de dinâmica do sistema
A	matriz do sistema sobredeterminado
p	posição dos pólos do sistema em malha fechada
m	dimensão do espaço de configuração
k	dimensão do espaço de características
b	parâmetro do controlador
P	matriz de covariâncias da estimativa
R	matriz de incerteza das observações visuais
S	matriz de incerteza das velocidades
Q	matriz de incerteza de predição
G	ganho de Kalman

Subscritos

<i>x</i>	referente ao eixo coordenado x
<i>y</i>	referente ao eixo coordenado y
<i>z</i>	referente ao eixo coordenado z
π	referente ao plano de projeção

Sobrescritos

<i>k</i>	referente ao instante de amostragem k
<i>i</i>	referente à imagem
<i>c</i>	referente à câmera esquerda
<i>d</i>	referente à câmera direita
<i>r</i>	referente ao robô
*	referente ao objetivo de controle
<i>s</i>	referente às características da imagem
$\hat{}$	estimativa
	coordenada homogênea

Siglas

LARA	Laboratório de Robótica e Automação
UnB	Universidade de Brasília
<i>PBVS</i>	<i>Position Based Visual Servoing</i>
<i>IBVS</i>	<i>Position Based Visual Servoing</i>
CTI	Centro de Tecnologia da Informação Renato Archer
CenPRA	Centro de Pesquisas Renato Archer
LASI	Laboratório de Sistemas Inteligentes
USP	Universidade de São Paulo
<i>AURORA</i>	<i>Autonomous Unmanned Remote Monitoring Robotic Airship</i>
<i>SLAM</i>	<i>Simultaneous Localization and Mapping</i>
<i>LIRMM</i>	<i>Laboratoire d'Informatique de Robotique et le Microélectronique de Montpellier</i>
FKE	Filtro de Kalman Estendido
GNM	Minimização de Gauss-Newton

1 INTRODUÇÃO

The beginning of knowledge is the discovery of something we do not understand.

Frank Herbert (1920 - 1986)

1.1 CONTEXTUALIZAÇÃO

1.1.1 Histórico do controle servo-visual

A autonomia e a capacidade de adaptação de um robô estão intimamente ligadas à habilidade de captar informações sobre o ambiente ao seu redor e de reagir a suas mudanças enquanto realiza uma determinada tarefa. Isso só se faz possível graças à incorporação de informação sensorial proveniente de uma ou mais fontes em seu laço de controle. Diversos tipos de sensores têm seu uso consagrado em aplicações de robótica, mas nenhum deles é capaz de proporcionar uma descrição tão rica do ambiente de trabalho quanto os sensores visuais [1]. A visão é uma forma de sensoreamento muito útil na robótica pois proporciona uma maneira de perceber o ambiente de trabalho sem contato físico e com muitos detalhes, a custo bastante acessível.

Os primeiros trabalhos que utilizaram sistemas de visão para o controle de robôs datam da década de 60 e faziam uso da informação visual em malha aberta, de forma que o sistema de aquisição de imagens e o controle dos atuadores eram independentes e agiam alternadamente. Essa abordagem em malha aberta comprometia a precisão da tarefa de controle pois acumulavam-se erros provenientes do sistema de visão, do posicionamento do robô e das transformações entre eixos coordenados.

Estes erros podem, no entanto, ser compensados utilizando-se uma arquitetura em que o sistema de visão fica dentro da malha de controle em um laço de realimentação de forma a se detectar a diferença entre a postura prevista e a postura real do robô. O uso, em malha fechada, de informações provenientes de visão computacional para controlar os movimentos de um robô é denominado na literatura por controle servo-visual. O controle servo-visual é a fusão de várias áreas mais elementares como teoria de controle, processamento de imagens, cinemática, dinâmica e tempo real com o intuito de controlar um robô para manipular ou mover-se no ambiente utilizando a visão como forma principal de realimentação.

O primeiro trabalho realizado em controle servo-visual é considerado o proposto por Shirai e Inoue [2] em 1973, embora o termo controle servo-visual só tenha sido formalizado anos mais tarde, em 1979 por Hill e Park [3]. Neste primeiro artigo, os autores expõem as vantagens de se utilizar realimentação visual dentro do laço de controle para aumentar a precisão e a robustez do sistema. No entanto, a área de controle servo-visual pouco progrediu entre o início dos anos 80 e meados da década de 90, principalmente devido às dificuldades de se processar as imagens em tempo hábil para sua utilização na malha de controle. Os últimos anos, porém, foram marcados por um crescente interesse da comunidade científica no uso de controladores baseados em sistemas de visão. Isso deve-se principalmente aos avanços recentes em microeletrônica que trouxeram equipamentos mais acessíveis e poderosos computacionalmente para o processamento de imagens.

Um grande marco para os estudos na área foi a publicação em 1987 do trabalho de Weiss *et al.* [4], em que se fez pela primeira vez a distinção entre as abordagens de controle servo-visual baseado em posição (*Position Based Visual Servoing* ou *PBVS*) e baseado em imagem (*Image Based Visual Servoing* ou *IBVS*). Estas duas abordagens são explicadas em detalhes no Capítulo 2. Em 1989, Fedemma e Mitchell [5] demonstraram o funcionamento do primeiro sistema de controle *IBVS*, formalizado posteriormente por Rives *et al.*[6], que investigaram abordagens *IBVS* para diversos tipos de primitivas como pontos, retas e elipses. Neste trabalho também foi apresentada a abordagem de controle servo-visual por Função Tarefa (*Task Function*), descrita com mais detalhes em [7].

O método *PBVS* progrediu com o estudo de Wang e Wilson [8], que propuseram o uso de um Filtro de Kalman Estendido para estimar a postura do robô a partir de uma série de pontos. Em 1993, nova contribuição ao controle *PBVS* foi feita por Papanikolopoulos e Khosla [9], que propuseram o rastreamento de um objeto por um braço manipulador utilizando idéias de controle adaptativo aplicadas à abordagem servo-visual. Prosseguindo com os avanços em controle servo-visual de manipuladores, Maru *et al.* [10] propuseram a utilização de visão estéreo em substituição à visão monocular usada nos trabalhos desenvolvidos até então. Com isso, buscavam extrair informação de profundidade das imagens e assim, evitar erros de aproximação que podiam comprometer a precisão do controlador *IBVS*. Outro importante trabalho contemplando sistemas de visão estéreo foi publicado por Hager [11], em que se propõe a modularização de tarefas de maior complexidade em várias outras tarefas menores e mais genéricas de forma a se obter leis de controle mais simples. Em 1998, Malis e Chaudette [12] introduziram o primeiro sistema híbrido *PBVS* e *IBVS*, buscando evitar os problemas de garantia de estabilidade da abordagem *IBVS* e eliminar a necessidade de um modelo 3D da abordagem *PBVS*. No entanto, para o caso de alvos não-planares, o método proposto exige a utilização de pelo menos oito pontos, além de apresentar certa sensibilidade à ruídos na imagem, pois faz uso direto das características visuais na lei de controle, sem qualquer tipo de estimação prévia.

As aplicações de controle servo-visual propostas até o momento abrangem uma enorme gama de aplicações: braços mecânicos para soldar, manipular ou mover objetos em ambientes industriais [13, 14], colheitadeiras automatizadas de frutas [15], pilotagem de veículos autônomos [16], pouso de aeronaves [17], robótica cirúrgica [18] e até mesmo aplicações mais inusitadas como pingue-pongue e malabarismos com robôs [19, 20]. A figura 1.1 mostra alguns exemplos do uso de controle servo-visual em diversas aplicações de robótica. Apesar da grande versatilidade do controle servo-visual, sua aplicação em áreas como robótica móvel ainda é incipiente se comparada à enorme quantidade de publicações voltadas especificamente ao controle de robôs manipuladores. Isso se deve ao grande aumento na complexidade dos problema quando se está em um ambiente de trabalho dinâmico e não-controlado. Uma coletânea detalhada das aplicações desenvolvidas ao longo dos anos em controle servo-visual de manipuladores pode ser vista em [21].

1.1.2 Visão em robótica móvel

Até o momento, a robótica tem tido muito menos impacto nas aplicações em que o ambiente de trabalho e a disposição dos objetos não podem ser controlados do que em ambientes que podem ser moldados para se adaptar ao robô como, por exemplo, os fabris. O uso de robôs em tarefas rotineiras que demandam grande interação com o ambiente ainda é bastante restrito, principalmente devido à



(a) Manipulador Comau SMART-3 S, PRISMA Lab., Università di Napoli Federico II, Nápoles, Itália
 (b) Colheitadeira, Agricultural and Environment Engineering Research, Aubière, França
 (c) Cycab, Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, França



(d) Dirigível Aurora AS800, Centro de Tecnologia da Informação Renato Archer, Campinas, Brasil
 (e) Braço cirúrgico, Institut de Recherche sur les Cancers de l'Appareil Digestif, Strasbourg, França
 (f) Manipulador Puma 560, Georgia Tech, Atlanta, Estados Unidos

Figura 1.1: Exemplos de aplicações de controle servo-visual na robótica

grande dificuldade em lidar com espaços de trabalho desconhecidos. Essa questão é extremamente importante no estudo de robôs móveis, pois estes precisam ser capazes de se mover pelo ambiente enquanto o exploram e identificam.

Há pouco mais de duas décadas, tem-se observado um forte crescimento da pesquisa em robótica móvel, envolvendo principalmente aspectos de navegação autônoma, em que muita pesquisa foi realizada nos temas de planejamento de rotas, locomoção, localização e mapeamento. Porém, os métodos inicialmente desenvolvidos que obtiveram maior sucesso envolviam o uso de radares a laser (laser *rangefinders*). Estes instrumentos costumam ser relativamente caros e não são capazes de representar o ambiente 3D com a mesma riqueza de detalhes de uma alternativa bem mais acessível: as câmeras de vídeo. No entanto, as dificuldades de processamento de imagens presentes na época eram desanimadoras, resultando em poucos avanços nos estudos de controle servo-visual.

Apenas recentemente, com os avanços registrados nas áreas de estimação e filtragem estocástica, reconhecimento de padrões e visão computacional, observou-se uma retomada do interesse da comunidade científica pela área de controle servo-visual em geral, e conseqüentemente, por suas aplicações na robótica móvel. Estudos visando o uso de informações visuais para tarefas de navegação e controle de robôs autônomos já foram propostos em problemas de robótica subaquática [22], aérea [23] e terrestre [24], sendo a última sub-área a mais explorado na literatura devido à enorme gama de aplicações. Dentre as várias publicações envolvendo robótica terrestre, destacam-se vários trabalhos com robôs não-holonômicos. Em [25], Masutani *et al.* abordam o problema de posicionamento de

um robô não-holonômico com câmera embarcada em frente a um *landmark*. O trabalho de Tsakiris *et al.* [26] estende as estratégias até então usadas em manipuladores para aplicações em robôs não-holonômicos com sistema de câmera e pan-tilt. Em [27], Ma *et al.* descrevem um método para seguimento de curvas por um carro. Já o artigo de Chen *et al.* [28] explora o uso de uma câmera observando o ambiente de trabalho de uma posição fixa no teto num sistema de controle servo-visual para robôs não-holonômicos.

Assim como no restante do mundo, onde predominam as pesquisas em manipuladores, no Brasil ainda não existem muitas frentes de pesquisa na área de controle servo-visual aplicado à robótica móvel. Destacam-se, no entanto, os trabalhos realizados no Centro de Tecnologia da Informação Renato Archer (CTI, antigo CenPRA) e no Laboratório de Sistemas Inteligentes (LASI) da Escola Politécnica da USP. No CenPRA, alguns trabalhos de controle servo-visual já foram desenvolvidos em robótica terrestre [29], embora o maior foco seja nas aplicações em robótica aérea, através do projeto *AURORA*, onde se vem desenvolvendo uma plataforma robótica aérea semi-autônoma no formato de um dirigível [30, 31, 32], enquanto no LASI foi desenvolvido um sistema de visão omnidirecional aplicado no controle de um robô terrestre, que deve manter-se a uma distância constante de determinado alvo móvel [33, 34].

1.1.3 Navegação de robôs móveis e o problema de *docking*

Um robô movendo-se num ambiente de trabalho requer, como habilidades básicas de navegação, a capacidade de detectar e evitar obstáculos, rastrear objetos de uma cena, navegar ao longo de corredores e paredes, atravessar portas e se aproximar de superfícies com uma dada orientação, por exemplo, ao entrar num elevador ou ao se aproximar de uma base de recarga. Assim sendo, a habilidade de mover-se até um objeto de interesse e abordá-lo de forma segura para realizar uma determinada tarefa, conhecida na literatura por *docking*, é pré-requisito básico no desenvolvimento de uma plataforma robótica autônoma.

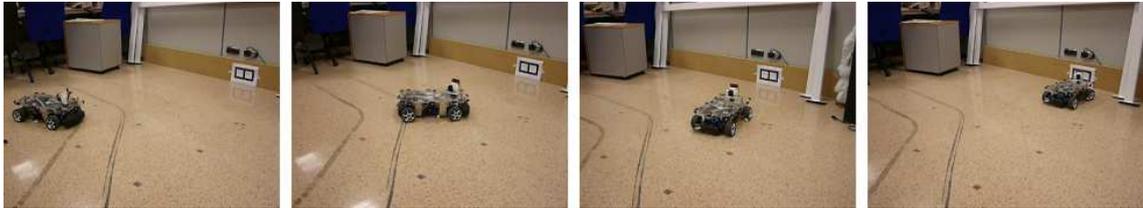
O *docking* pode ser definido como o deslocamento do robô a partir de uma postura atual para uma postura desejada seguindo uma trajetória segura (*cf.* Figura 1.2). A posição e orientação final do robô devem ser adequada às tolerâncias requeridas pela tarefa a ser realizada e o robô deve mover-se o mais próximo possível da superfície alvo, sem colidir com ela. Os tipos de alvo para o *docking* variam conforme a tarefa a ser realizada, indo desde uma estação de recarga de baterias, um objeto a ser trabalhado por um manipulador ou um ponto localizado "no infinito" para seguimento de corredores até alvos em movimento, como por exemplo um outro robô móvel em aplicações de robótica cooperativa.

Por estarem intimamente ligados ao tópico de rastreamento de objetos, os problemas de *docking* costumam envolver o uso de informação visual o que torna as estratégias de controle servo-visual bastante atraentes para solucioná-los. O interesse por estudos na área é um fenômeno recente e explica-se pelos crescentes avanços nas pesquisas em robôs autônomos. Como os primeiros trabalhos publicados datam do fim da década de 90, este ainda não é um assunto consolidado na literatura e várias abordagens diferentes vêm sendo propostas e investigadas.

Uma das vertentes atuais explora o problema de *docking* servo-visual utilizando abordagens por



(a) Aproximação de objeto por manipulador [35]



(b) Posicionamento de robô móvel em frente a um *landmark* [36]

Figura 1.2: Exemplos de trabalhos em *docking* servo visual

aprendizado, baseadas em redes neurais e algoritmos de *Q-learning* em que o veículo autônomo deve posicionar-se com relação a um *landmark* colocado no ambiente [36] ou aproximar-se de um objeto alvo que derá ser apanhado pelo robô [37, 35]. Apesar dos bons resultados apresentados, este método tem como grande inconveniente a necessidade de uma etapa de treinamento para cada tipo de situação antes que o robô consiga trabalhar em modo autônomo. Uma possibilidade explorada em [38] e [39] consiste em utilizar informações de fluxo óptico para implementar um controle *IBVS*. No primeiro trabalho, tal estratégia foi aplicada para o *docking* de um veículo aéreo em forma de dirigível em uma estação de apoio. O segundo artigo propõe um sistema remoto de *docking* utilizando um controlador servo-visual com lógica fuzzy. A principal desvantagem destas propostas deve-se à necessidade de conhecimento *a priori* do alvo, o que reduz bastante a flexibilidade do sistema. Outra abordagem muito explorada envolve um controle *PBVS* a partir da posição cartesiana e orientação do robô com relação à um sistema coordenado cuja origem é um ponto do objeto alvo. Esse método foi utilizado com sucesso em [40] para estacionar um robô de vigilância numa base de recarga, em [41] para o *docking* de um robô sub-aquático em uma estação de apoio submersa e em [42] para seguimento de corredores utilizando o ponto de fuga formado por duas retas horizontais como alvo de controle. No entanto, tais estudos baseiam-se em um ou mais pontos como características da imagem, desperdiçando-se com isso, muita informação sobre o ambiente que poderiam ser utilizadas para melhorar a confiabilidade e robustez do sistema.

Uma alternativa interessante ao uso de pontos é trabalhar com retas, pois estas são mais robustas às perturbações e erros de medição inerentes às câmeras de vídeo. Entretanto, a grande maioria dos trabalhos de controle servo-visual se baseia em pontos por estes serem abundantes nas imagens e apresentarem representação e equações de estado mais simples se comparados às retas. Uma tentativa de contornar as dificuldades do uso de retas tem sido a adoção da representação de segmentos de retas por coordenadas de Plücker. Esta representação para retas tridimensionais possui propriedades bastante interessantes no espaço projetivo das câmeras e vem sendo cada vez mais utilizada em trabalhos de controle servo-visual. Os trabalhos de Rives *et al.* [43] e Weng *et al.* [44] foram alguns dos primeiros a explorar esse tipo de representação num contexto de robótica móvel, desenvolvendo métodos

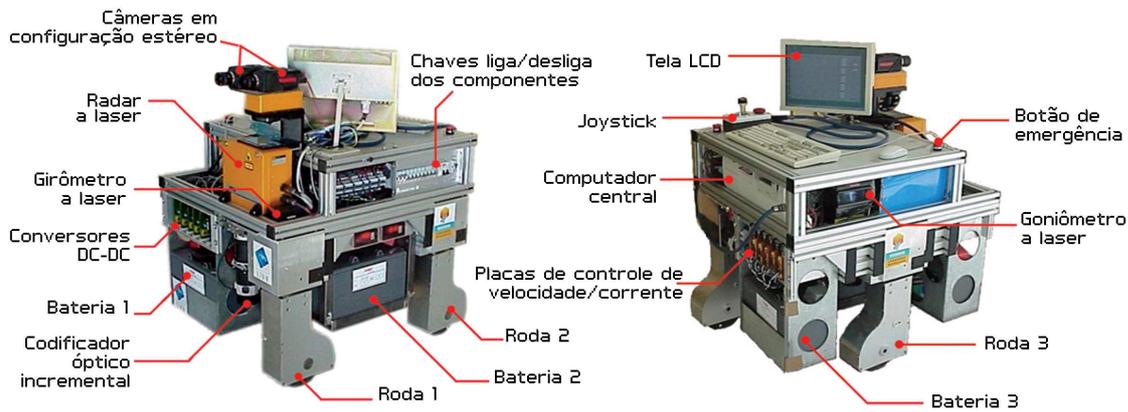


Figura 1.3: Plataforma móvel Omni e seus componentes

de estimação das coordenadas de Plücker a partir de imagens consecutivas de uma câmera em movimento, enquanto em [45] é proposta uma solução fechada para estimação de movimento e estrutura por correspondência de retas. Um método estocástico para estimação das coordenadas de Plücker à partir de imagens é proposto em [46], onde aborda-se o problema de localização e mapeamento simultâneos (*Simultaneous Localization and Mapping* ou *SLAM*) a partir das imagens de um sistema de visão monocular. Apesar de tratarem apenas do problema de determinação dos parâmetros tridimensionais de retas, tais trabalhos constituem uma importante base teórica para o estudo de controle servo-visual. O uso de coordenadas de Plücker aplicado em controle servo-visual foi abordado por Andreff em [47, 48] e aplicado num caso particular de *docking* em que o alvo é um triedro ortogonal. O método desenvolvido nos trabalhos de Andreff permite o projeto de controladores bastante simples e efetivos, porém o sistema não é robusto nos casos em que o alvo sai do campo de visão da câmera, além de ser necessário fazer uso de um feixe laser para se recuperar informação de profundidade.

1.2 MOTIVAÇÃO

1.2.1 Robótica Móvel no LARA

O Laboratório de Robótica e Automação (LARA) vem desenvolvendo trabalhos na área de robótica móvel terrestre com estudos em cartografia e localização simultâneos, cooperação de robôs móveis, controle de marcha para robôs quadrúpedes e planejamento probabilístico de rotas. A principal plataforma de testes utilizada no LARA, chamada Omni, vem sendo desenvolvida desde 1998 para fins de robótica móvel em ambientes internos estruturados. Sua estrutura mecânica foi projetada e contruída na tese de Le Corre [49], no *LIRMM (Laboratoire d'Informatique de Robotique et le Microélectronique de Montpellier)*, em Montpellier, na França. Desde então, a plataforma vem sofrendo modificações tanto na sua arquitetura de *hardware* quando de *software* a fim de satisfazer os requisitos de diversos trabalhos. O Omni é um robô omnidirecional de três rodas acionado por motores de corrente contínua e dotado de uma série de dispositivos como codificadores ópticos incrementais e absolutos, girômetro a laser, *ladar* a laser e sistema de câmeras estéreo (cf. Figura 1.3). Um estudo detalhado da arquitetura e dispositivos presentes atualmente na plataforma Omni encontra-se em [50].

Diversos estudos em robótica móvel já foram realizados sobre a plataforma Omni, dentre os quais destacam-se o problema de mapeamento bidimensional e localização robusta abordado em [51], extração de retas a partir da leitura de radares a laser [52], estimação de pose de um robô móvel usando mapas geométricos [53] e planejamento probabilístico de rotas [54]. Entretanto, os trabalhos até então desenvolvidos fazem uso de equipamentos extremamente caros como o *ladar* a laser, indo de encontro à tendência atual de utilizar atributos visuais e sensores de custo mais acessível como forma de popularizar a robótica móvel. Assim sendo, é interessante desenvolver no LARA trabalhos de robótica terrestre que envolvam o uso dispositivos de menor custo, como por exemplo, câmeras de vídeo.

1.2.2 Controle servo-visual de trajetória e mapeamento local simultâneos

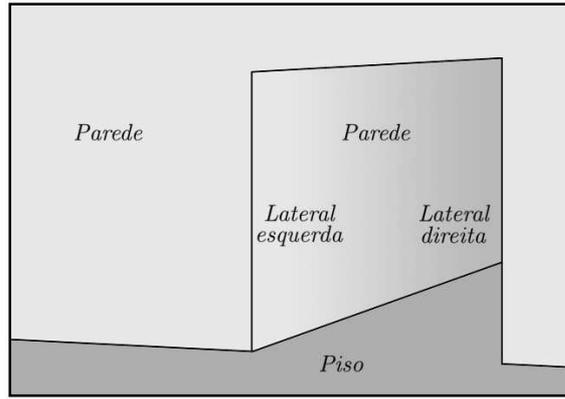
Classicamente, o problema de navegação de robôs móveis envolve diversos temas como localização, mapeamento, planejamento de rotas e controle de trajetória [55]. Por se tratarem de temas por si só suficientemente complexos, esses vários tópicos costumam ser tratados de forma dissociada na literatura e geralmente são empregados separadamente em um sistema integrado de navegação.

Contudo, ao tratar controle de trajetória e mapeamento local separadamente, perde-se de vista a possibilidade de melhoria no desempenho de ambos que poderia ser obtida se, por exemplo, a tarefa de controle de trajetória levasse em consideração ações que permitissem uma melhoria na estimação do mapa local. Por outro lado, o sistema de controle pode levar em consideração incertezas do mapa local de modo a impor cautela à tarefa de controle de trajetória. Em um sistema de controle servo-visual, uma abordagem de controle e mapeamento simultâneos é extremamente conveniente pois o mapa local construído permite que o controlador continue atuando mesmo em situações em que as características observadas na imagem sejam momentaneamente obstruídas ou saiam do campo de visão da câmera. Isso confere maior robustez ao sistema, tornando-o mais confiável para tarefas com grandes exigências de desempenho e conhecimento do ambiente como, por exemplo, uma aplicação de *docking*.

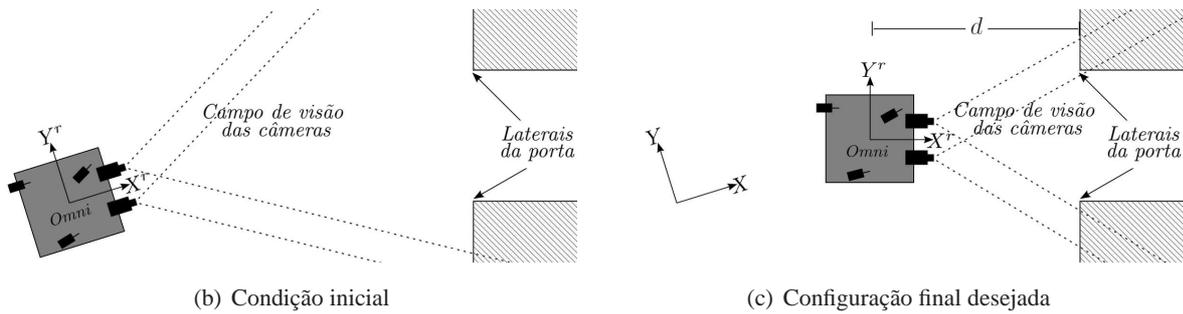
Sendo assim, este trabalho busca obter uma abordagem de mapeamento local e controle de trajetória simultâneos para robôs móveis cuja avaliação experimental é feita em uma tarefa de *docking* sobre a plataforma Omni, disponível no laboratório LARA. Para manter a abrangência da solução proposta, o método desenvolvido deve utilizar um mínimo de informação *a priori*, permitindo assim seu emprego em exploração de ambientes sem intervenção humana. Por serem elementos muito presentes em ambientes estruturados, a utilização apenas de retas como características a serem rastreadas na imagem torna o problema extensível a várias outras aplicações de navegação de robôs como, por exemplo o seguimento de paredes e corredores.

1.3 DEFINIÇÃO DO PROBLEMA E OBJETIVOS

O problema em estudo consiste em realizar uma tarefa de *docking* de um robô móvel omnidirecional em frente à uma porta qualquer, utilizando apenas dados provenientes de um sistema de odometria e visão. A posição inicial da porta é desconhecida, assim como a postura do robô com relação a ela.



(a) Esboço de imagem provida pela câmera



(b) Condição inicial

(c) Configuração final desejada

Figura 1.4: Esquemático do problema de *docking* proposto

As duas laterais da porta devem estar inicialmente no campo de visão das câmeras e na inicialização do sistema, deve-se indicar a referência de controle e quais dos segmentos de retas extraídos do primeiro par de imagens corresponde a essas laterais. A partir de então, empregando uma estratégia de controle servo-visual, o robô deverá se aproximar da porta sem qualquer intervenção humana, de forma a se posicionar perpendicularmente e a uma distância d pré-determinada (cf. Figura 1.4(c)). Ao final da ação de controle a porta deverá ficar próxima o suficiente a ponto de, se o robô seguir em linha reta, atravessá-la sem colisão.

Uma possível abordagem para problemas de *docking* similares ao proposto envolve estimar a postura do robô e dos marcadores com relação a um sistema de referência global \mathcal{O}^w e, a partir destas informações, determinar a ação de controle. No entanto, optou-se por outra estratégia, em que considera-se apenas o posicionamento relativo entre robô e objetos da cena. Com isso, elimina-se a necessidade de estimar a postura do robô explicitamente, bastando determinar os parâmetros da porta com relação ao seu sistema de coordenadas \mathcal{O}^r . Como este se encontra sempre em movimento, as coordenadas dos objetos evoluem temporalmente, devendo ser constantemente atualizadas, reconstruindo o ambiente em torno do robô à medida que ele se desloca.

Usualmente, o problema de controle servo-visual considera que os atributos de imagem usados como medições são bem caracterizados por parâmetros conhecidos *a priori*. Por exemplo, usa-se marcadores compostos de um certo número de pontos cujas coordenadas são consideradas conhecidas e constantes. No caso específico de uma porta, os atributos seriam suas duas laterais que em uma imagem capturada pela câmera do robô correspondem a retas verticais. Os parâmetros que representam estas duas retas no espaço de trabalho determinam o quão distante o robô se encontra da posição

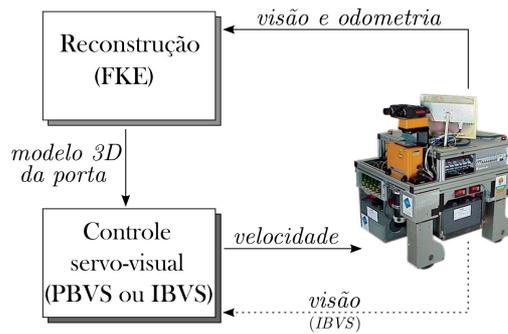


Figura 1.5: Principais elementos do sistema proposto.

final que é o objetivo de controle.

No problema aqui proposto, o desafio torna-se ainda maior devido ao fato de tais parâmetros serem desconhecidos inicialmente. Existe, portanto, um problema de reconstrução local que consiste na estimação dos parâmetros tridimensionais das laterais da porta. Estes parâmetros farão parte de um vetor de estados a ser estimado empregando um filtro estocástico não-linear do tipo FKE (Filtro de Kalman Estendido). Essa abordagem é similar àquela usada para localização e mapeamento de ambientes, mas como o número de estruturas do ambiente é fixo antecipadamente (no caso, duas retas), prefere-se então dizer que o problema em questão é de reconstrução, e não de mapeamento.

O sistema de controle deve ser capaz de levar o robô a uma determinada configuração diante da porta fazendo uso das imagens adquiridas pela câmera, em malha fechada. Essas imagens são processadas, gerando medidas utilizadas na etapa de correção do FKE e no cálculo do sinal de controle. A ação de controle é determinada de forma a minimizar o erro entre as características visuais atuais e a referência, calculada a partir de informações fornecidas previamente pelo usuário, como a distância d e a largura da porta em questão. No entanto, a informação de largura não é estritamente necessária, e poderia ser obtida diretamente das estimações do FKE, deixando a implementação ainda mais versátil e flexível. Um esquemático da arquitetura do sistema é apresentado na Figura 1.5, em que evidencia-se o problema de reconstrução e controle simultâneos.

Resumindo em linhas gerais, o objetivo deste trabalho é obter uma abordagem de reconstrução estocástica e controle de trajetória simultâneos para robôs omnidirecionais utilizando estratégias de controle servo-visual. A metodologia proposta deve ser avaliada em uma tarefa de *docking* em frente a portas, sem informação *a priori* sobre a localização do alvo e utilizando como fonte de medidas apenas o sistema de visão e sensores de odometria e validada tanto em simulação computacional quanto sobre uma plataforma robótica real. O processo de reconstrução utiliza retas 3D para representar o objeto de interesse, de forma que a solução aqui desenvolvida é extensível a outras aplicações que também façam uso de retas como características visuais.

1.4 RESULTADOS ALCANÇADOS

Ao final deste trabalho, chegou-se a um novo método estocástico para estimação de parâmetros 3D de retas e aplicação de duas diferentes abordagens de controle servo-visual baseado em retas, sendo

que uma é realizada com retas no espaço da imagem (controle 2D) e a outra se utiliza da representação das retas no espaço tridimensional (controle 3D). Ambas foram avaliadas numa tarefa de *docking* em frente à uma porta e obtiveram resultados satisfatórios tanto nas simulações computacionais quanto na validação sobre uma plataforma robótica real.

1.5 APRESENTAÇÃO DO MANUSCRITO

O Capítulo 2 é dividido em duas seções. A Seção 2.1 trata conceitos básicos de controle servo-visual, suas principais classificações e propriedades enquanto a Seção 2.2 apresenta alguns dos modelos matemáticos para pontos e retas, o modelo geométrico usual para câmeras e a transformação de projeção. O Capítulo 3 refere-se à abordagem de controle proposta neste trabalho. As Seções 3.1 e 3.2 apresentam definições necessárias para descrever o problema em estudo. Na Seção 3.3 faz-se o projeto dos controladores *IBVS* e *PBVS* baseados em retas e na Seção 3.4 descreve-se o filtro estocástico usado para realizar a estimação dos parâmetros tridimensionais das retas. A Seção 3.5 apresenta o método de cálculo das observações visuais e a Seção 3.6 descreve brevemente o procedimento de extração e tracking utilizado para processamento das imagens. Finalmente, no Capítulo 4, tem-se os principais resultados obtidos em avaliação computacional e em validação sobre a plataforma robótica Omni. Por fim, o Capítulo 5 encerra com as conclusões a cerca desta dissertação e propostas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

It is the theory that decides what we can observe.

Albert Einstein (1879 - 1955)

2.1 CONTROLE SERVO-VISUAL

O termo controle servo-visual refere-se ao uso, em malha fechada, de informações provenientes de visão computacional para determinar a ação de controle de um robô com o objetivo de realizar uma tarefa. A formulação deste tipo de problema consiste em usar informações provenientes de imagens para controlar a postura do robô em relação a um objeto alvo presente na cena, ou a um conjunto de características visuais extraídas da imagem. O uso de câmeras como sensores em tarefas de controle tem sido bastante estudado e explorado nas últimas décadas devido a algumas características bastante interessantes dos sensores visuais, dentre as quais destacam-se:

- serem sensores de medição sem contato, não interferindo no ambiente de trabalho, nem estando sujeitos a desgastes por atrito ou pressão;
- capacidade de mimetizar o sentido humano, tornando os algoritmos de controle mais simples e intuitivos;
- amplo campo de cobertura, estendendo-se a partir de alguns centímetros e podendo chegar até vários metros, o que permite sua utilização em uma grande gama de aplicações;
- possuir taxa de amostragem compatível com aplicações de tempo real;
- baixo custo se comparados às outras alternativas disponíveis, como lasers e sensores de ultrassom.

No entanto, o uso de imagens na malha de controle também apresenta inconvenientes, sendo o principal deles, a latência existente no processo de aquisição, processamento e interpretação das informações contidas nas imagens. Os atrasos gerados pela espera do processamento das informações visuais podem causar problemas de instabilidade no sistema de controle e devem ser investigadas soluções adequadas para contorná-los. Contudo, o avanço recente das pesquisas em áreas como processamento digital de imagens e visão computacional, aliados às novas tecnologias em micro-eletrônica que permitiram o aumento no desempenho dos processadores, têm minimizado progressivamente os tempos de atraso no processamento de informações visuais.

Esta seção apresenta os principais conceitos de controle servo-visual e as abordagens mais utilizadas na literatura.

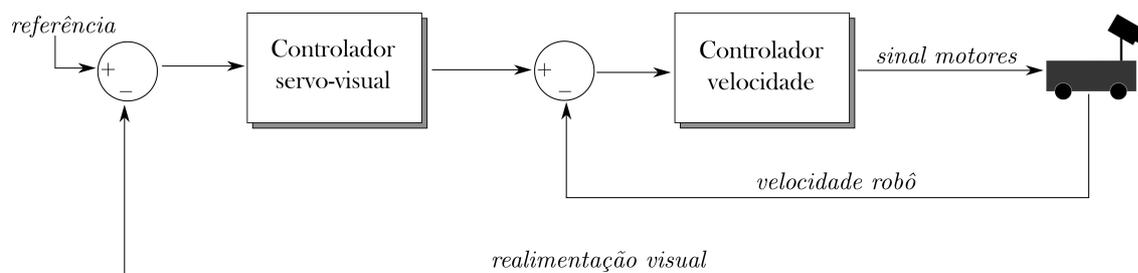
2.1.1 Classificação dos tipos de controle servo-visual

Existem diversas formas de controlar um robô a partir das informações extraídas de um sistema de câmeras. Essas diferentes abordagens podem ser classificadas [56] quanto ao arranjo hierárquico

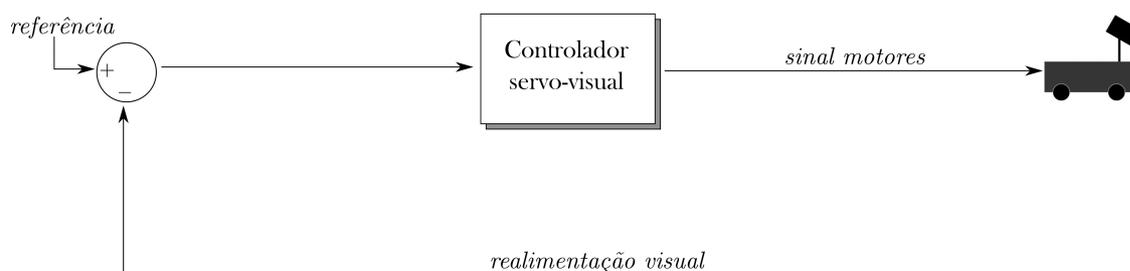
do controlador, quanto à configuração do sistema de câmeras e quanto à estratégia de controle.

2.1.1.1 Arranjo hierárquico do controlador

As duas possíveis configurações hierárquicas do controlador servo-visual são a arquitetura *look-and-move* e o controle servo-visual direto (cf. Figura 2.1). A estrutura da arquitetura *look-and-move* é hierárquica e usa o sistema de visão para fornecer referências de velocidade ou torque para um outro controlador de mais baixo nível, responsável por atuar diretamente sobre os motores. Já o controle servo-visual direto é não-hierárquico, pois elimina o controle de baixo nível, usando diretamente as informações visuais para estabilizar o robô. Apesar da arquitetura mais simples, o controle servo-visual direto apresenta algumas desvantagens [56]. Em primeiro lugar, a baixa taxa de amostragem dos sistemas de visão computacional tornam o controle direto de sistemas com dinâmica não-linear um problema extremamente complicado. Em segundo lugar, a maioria dos robôs já possui controladores de mais baixo nível implementados, sendo desnecessário que o controlador servo-visual aja diretamente nos atuadores. Por fim, a arquitetura *look-and-move* permite que o controle servo-visual esteja dissociado das singularidades cinemáticas do robô, facilitando a implementação da lei de controle. Devido a estes fatores, a grande maioria dos sistemas de controle servo-visual adota a arquitetura hierárquica, o único arranjo abordado neste trabalho.



(a) Arquitetura *look-and-move*



(b) Arquitetura de controle servo-visual direto

Figura 2.1: Tipos de hierarquia de controle servo-visual.

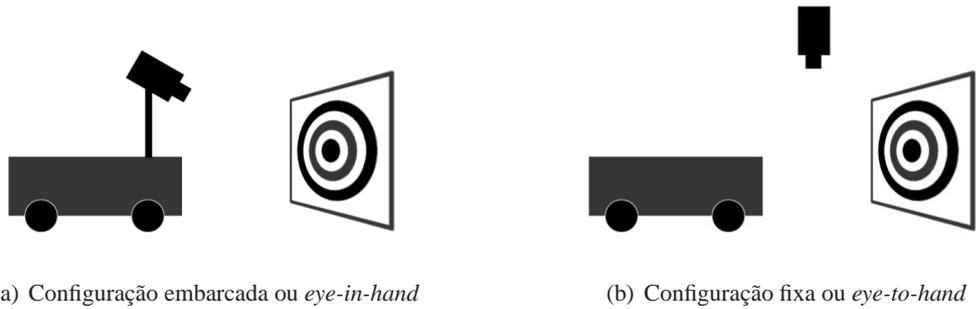


Figura 2.2: Tipos de configuração de câmera.

2.1.1.2 Configuração do sistema de câmeras

As configurações de câmera podem ser classificadas em fixa ou embarcada (*cf.* Figura 2.2). Na configuração fixa, também conhecida como *eye-to-hand*¹ [20], [28], a câmera fica imóvel num ponto fixo no sistema de coordenadas global, de onde registra imagens do movimento do robô no ambiente de trabalho. Na configuração embarcada, também chamada de *eye-in-hand* [40, 24, 37], a câmera é fixada sobre o robô e move-se junto com ele, sendo necessário determinar como o movimento do robô afeta a posição da câmera. Enquanto a configuração *eye-to-hand* tem uma visão global da tarefa mas com pouca precisão, a configuração *eye-in-hand* tem uma visão parcial da cena, porém mais precisa, principalmente para tarefas de aproximação (ou *docking*). As duas configurações de câmera não são excludentes, sendo que existem trabalhos que fazem uso de formações híbridas, contendo tanto câmeras fixas quanto embarcadas.

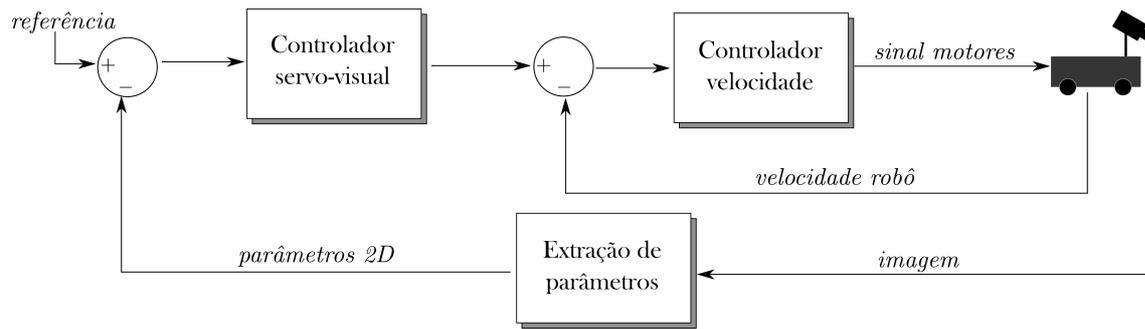
2.1.1.3 Estratégia de controle

Quanto à estratégia de controle, destacam-se principalmente duas metodologias: o controle baseado na imagem (*Image Based Visual Servoing* ou *IBVS*) e o controle baseado na posição (*Position Based Visual Servoing* ou *PBVS*), esquematizados da Figura 2.3. No *IBVS*, o controle é feito diretamente a partir das informações extraídas das imagens fornecidas pelo sistema de câmeras. O sinal de erro do controlador é definido como a diferença entre as medições da imagem atual e as medições da imagem final de referência. É, portanto, um controle 2D, em que a referência a ser seguida é o conjunto de parâmetros extraídos da imagem captada pela câmera quando o robô se encontra na posição final desejada. Essa abordagem pode reduzir atrasos computacionais e erros devido ao modelamento do sistema e calibração da câmera. No entanto, o desafio para o controlador é maior devido aos acoplamentos e não-linearidades da planta quando se utiliza esta abordagem [56].

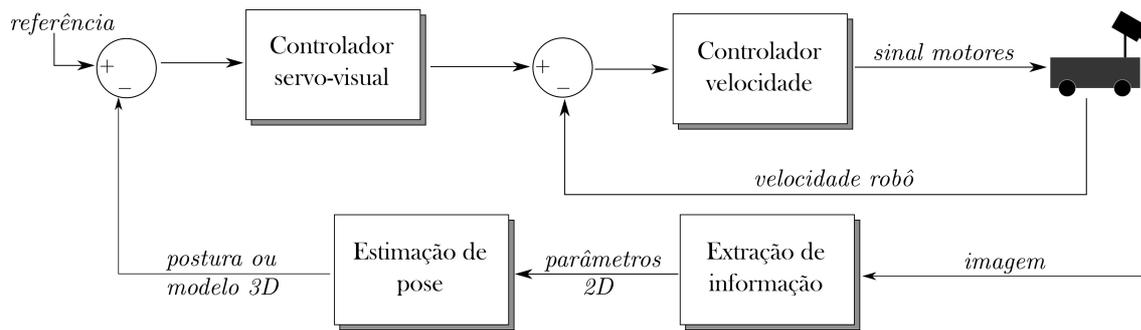
Já no *PBVS*, parâmetros são extraídos das imagens e utilizados para estimar a pose ou o modelo 3D do alvo em relação ao robô. Com esses valores, define-se a medida de erro entre a postura atual e a desejada para fins de cálculo do sinal de controle. Desta forma, trata-se de um controle 3D, em que a referência a ser seguida é a postura ou o modelo 3D final desejado para o alvo no espaço de trabalho.

A abordagem *IBVS* apresenta menor custo computacional pois elimina a necessidade de calcular

¹Essa expressão tem origem no controle servo-visual de manipuladores.



(a) Abordagem IBVS



(b) Abordagem PBVS

Figura 2.3: Tipos de estratégia de controle servo-visual

os parâmetros 3D do alvo, além de ser robusta aos erros provenientes da modelagem e calibração da câmera. Por outro lado, o uso da estratégia *IBVS* pode implicar em maior dificuldade de estabilização devido à ausência do modelo tridimensional do alvo em um problema de controle potencialmente complexo e não-linear. Outro potencial problema desta abordagem é a existência de mínimos locais que levam o sinal de controle a zero em posição diferente da desejada [57].

Apesar de o controle baseado em imagem ser o único que utiliza as medidas das imagens diretamente como referência, os controladores baseados em posição fazem uso das imagens em suas estimações. Assim sendo, em ambas estratégias de controle, a taxa de aquisição das câmeras é fator importante a ser observado.

2.1.2 Características da Imagem

Nos algoritmos de controle servo-visual, a imagem geralmente não é utilizada diretamente como medida de informação. É necessária uma etapa anterior, quando são extraídas dela as informações relevantes para o problema em questão. Entende-se por informações relevantes aquelas necessárias para alimentar o sistema de controle e que são conhecidas na literatura como características da imagem ou *image features*. Embora possam ser obtidas a partir da imagem completa (caso do fluxo óptico ou do histograma), é bastante comum que as características da imagem estejam relacionadas a propriedades dos objetos inseridos no ambiente de trabalho, como uma porta, um corredor, ou até mesmo um *landmark* colocado na cena. Grande parte das pesquisas de controle servo-visual baseiam-se em objetos que sejam facilmente identificados e separados do restante da imagem. Tal processo, conhecido

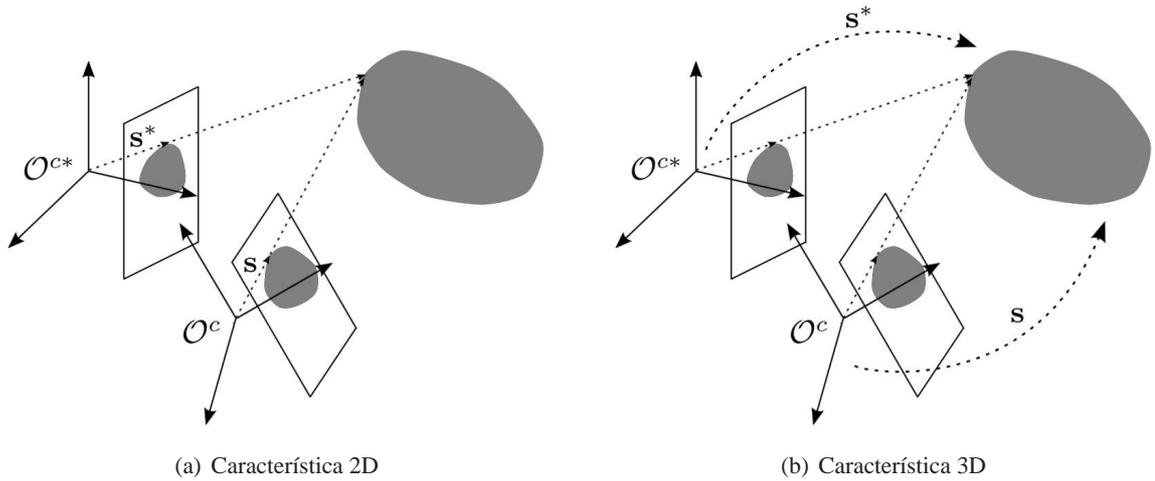


Figura 2.4: Exemplo de características da imagem para o caso bidimensional (a) e tridimensional (b). Em ambos, s representa o vetor de características da imagem.

por segmentação, tem sido objeto de estudo de vários trabalhos e diversas abordagens diferentes já foram propostas, a maioria utilizando filtros para extração de regiões com alto contraste como bordas e vértices.

Assim sendo, define-se o termo *característica da imagem* como sendo um elemento geométrico extraído a partir do conjunto de pixels de uma ou mais imagens. Esta característica, dependendo da abordagem de controle, pode ser um elemento bidimensional, correspondente à projeção na imagem de objetos da cena (contornos do alvo, bordas de estradas, pontos das extremidades, etc.); ou um elemento tridimensional que os representa geometricamente no espaço (pontos, retas, elipses, cilindros, etc.). Ambas possibilidades são exemplificadas na Figura 2.4. Uma boa característica da imagem é aquela que pode ser extraída de diferentes pontos de vista, de forma robusta e sem ambigüidades [58] e sua escolha irá depender do problema específico de controle a ser resolvido.

Os parâmetros de uma característica da imagem são definidos como sendo valores reais f_i calculados a partir de uma ou mais características da imagem, como por exemplo, as coordenadas de um ponto, o tamanho dos semi-eixos de uma elipse ou o comprimento de um segmento de reta. Dado um conjunto de k parâmetros de características da imagem, pode-se definir um vetor de características da imagem $\mathbf{s} = [f_1 \ \dots \ f_k]^T \in \mathcal{F} \subseteq \mathbb{R}^k$, onde \mathcal{F} representa o espaço de características da imagem.

2.1.3 Controlador Servo-Visual

O objetivo do controlador servo-visual é minimizar um erro $\mathbf{e}(t)$ tipicamente definido como

$$\mathbf{e}(t) = \mathbf{s}^*(t) - \mathbf{s}(\mathbf{m}(t), \mathbf{a}), \quad (2.1)$$

em que $\mathbf{s}^*(t)$ é o vetor de características da imagem desejado e $\mathbf{s}(\mathbf{m}(t), \mathbf{a})$ é o vetor de características da imagem atual, computadas a partir do conjunto $\mathbf{m}(t)$ de pixels extraídos da imagem e de \mathbf{a} , que representa as informações conhecidas do sistema como, por exemplo, parâmetros intrínsecos das câmeras ou o modelo 3D do sistema [59]. Os vetores \mathbf{s}^* e \mathbf{s} são designados de acordo com a estratégia de controle adotada, podendo ser parâmetros 3D estimados a partir das imagens, no caso

da abordagem *PBVS*, ou parâmetros 2D extraídos diretamente da imagem, no caso *IBVS*.

Seja $\mathbf{r} \in \mathbb{R}^m$ o vetor de posições e orientações do robô num dado instante. O espaço de configuração do robô, representado por \mathcal{T} , é definido como o conjunto de todas posturas possíveis para o robô, tal que $\mathbf{r} \in \mathcal{T}$. Para um vetor \mathbf{s} de características da imagem, a Matriz de Interação (\mathbf{L}_s), também conhecida como Jacobiana da Imagem, é uma transformação linear do espaço tangente a \mathcal{T} em \mathbf{r} para o espaço tangente a \mathcal{F} em \mathbf{s} [56], ou seja,

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{t}} = \mathbf{L}_s \dot{\mathbf{r}}, \quad (2.2)$$

em que $\mathbf{L}_s \in \mathbb{R}^{k \times m}$ e

$$\mathbf{L}_s = \left[\frac{\partial \mathbf{s}}{\partial \mathbf{r}} \right] = \begin{bmatrix} \frac{\partial f_1(\mathbf{r})}{\partial r_1} & \dots & \frac{\partial f_1(\mathbf{r})}{\partial r_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_k(\mathbf{r})}{\partial r_1} & \dots & \frac{\partial f_k(\mathbf{r})}{\partial r_m} \end{bmatrix}, \quad (2.3)$$

com m igual à dimensão do espaço de configuração \mathcal{T} e k igual à dimensão do espaço de características \mathcal{F} .

A relação estabelecida em (2.2) descreve como o vetor de características da imagem é afetado pela mudança na postura do robô. No entanto, nas aplicações de controle servo-visual normalmente utiliza-se a relação inversa: o cálculo de $\dot{\mathbf{r}}$ a partir de $\dot{\mathbf{s}}$. Contudo, nos casos em que a Matriz de Interação é não-singular ou quando $k \neq m$, a matriz \mathbf{L}_s^{-1} não existe. Assim sendo, quando necessário, faz-se uso de sua pseudo-inversa de Moore-Penrose (\mathbf{L}_s^\dagger)

$$\dot{\mathbf{r}} = \mathbf{L}_s^\dagger \dot{\mathbf{s}} \quad (2.4)$$

em que \mathbf{L}_s^\dagger é dada por

$$\mathbf{L}_s^\dagger = (\mathbf{L}_s^T \mathbf{L}_s)^{-1} \mathbf{L}_s^T, \quad (2.5)$$

que é a melhor estimativa para a \mathbf{L}_s^{-1} no sentido de mínimos quadrados [60].

A matriz \mathbf{L}_s^\dagger depende tanto de parâmetros obtidos diretamente das imagens quanto de informações 3D das primitivas. Como os primeiros são influenciados por erros de calibração e ruído nas imagens e as últimas geralmente são desconhecidas *a priori*, é necessário escolher um modelo de aproximação $\hat{\mathbf{L}}_s^\dagger$ para \mathbf{L}_s^\dagger [7]. Dentre as diversas escolhas possíveis de $\hat{\mathbf{L}}_s^\dagger$, destacam-se [57]:

- $\hat{\mathbf{L}}_s^\dagger = \mathbf{L}^\dagger$, em que \mathbf{L}^\dagger é estimada numericamente durante o movimento da câmera sem levar em conta descrição analítica da Matriz de Interação (caso do uso de redes neurais, por exemplo). Essa abordagem é bastante interessante nos casos em que o conhecimento do problema é mínimo. No entanto, não há garantia de estabilidade, o que pode resultar em trajetórias sinuosas, levando características da imagem para fora do campo de visão da câmera (*cf.* Figura 2.5)(a).
- $\hat{\mathbf{L}}_s^\dagger = \mathbf{L}_s^\dagger$, em que $\hat{\mathbf{s}}$ é o vetor de características estimado a partir das imagens e/ou outras informações disponíveis ao sistema. Nesta abordagem, a matriz $\hat{\mathbf{L}}_s^\dagger$ é atualizada a cada interação do laço de controle. Quanto menor o erro de estimação, mais próximo fica-se da condição ideal

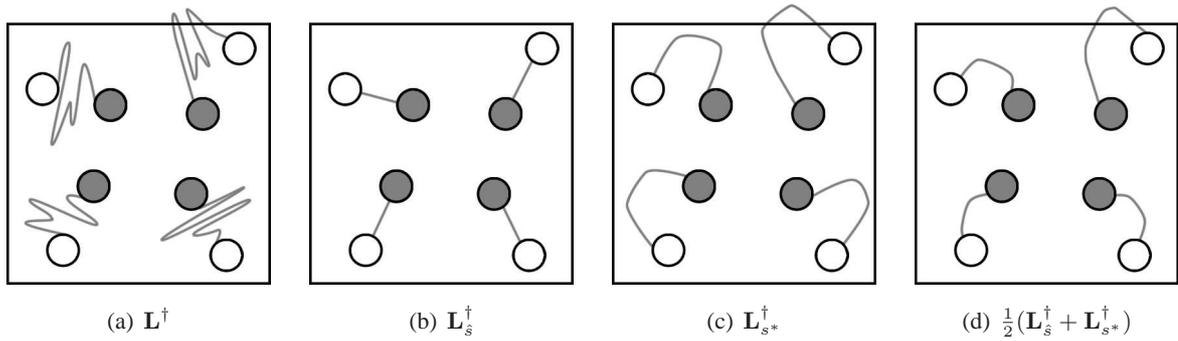


Figura 2.5: Possíveis escolhas para $\hat{\mathbf{L}}_s^\dagger$ e comportamento correspondente numa aplicação *IBVS*: pontos brancos representam as posições iniciais enquanto pontos e linhas cinzas representam, respectivamente, a posição desejada e uma possível trajetória.

$\hat{\mathbf{L}}_s^\dagger \mathbf{L}_s = \mathbf{I}$, situação em que tem-se um sistema desacoplado com cada parâmetro alcançando o valor desejado seguindo o menor caminho (cf. Figura 2.5)(b). Contudo, tal abordagem pode implicar em movimentos inadequados da câmera, levando a configurações de mínimo local.

- $\hat{\mathbf{L}}_s^\dagger = \mathbf{L}_{s^*}^\dagger$, em que s^* é o vetor de características desejado ao fim do controle. Este modelo tem valor constante durante toda a execução e portanto, é determinado previamente numa etapa *off-line*. Devido à necessidade de adotar valores aproximados para as informações 3D, a convergência só é assegurada em uma pequena vizinhança em torno da posição desejada.(cf. Figura 2.5)(c).
- $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_{s^*}^\dagger)$, em que utiliza-se uma média dos dois modelos anteriores (cf. Figura 2.5)(d). Na prática, a escolha dessa aproximação costuma levar a boas performances com trajetórias razoavelmente suaves [61].

2.2 REPRESENTAÇÃO DE OBJETOS NA IMAGEM

Pontos e retas são as primitivas de uso mais comum em visão computacional devido à facilidade de serem extraídos das imagens e por estarem presentes na maioria dos ambientes de trabalho. O ponto é, de todas as primitivas geométricas, a mais elementar. Sua representação é simples e de fácil extração a partir das imagens, tornando-o a primitiva mais utilizada em aplicações de visão computacional. Já as retas são muito importantes para representar objetos das cenas. Modelos de objetos poliédricos são construídos a partir de retas, assim como uma trajetória para um robô no espaço 3D pode ser representada por uma cadeia de retas [62].

Esta seção discute as principais representações geométricas de pontos e retas no espaço e na imagem, além de fazer uma breve descrição do modelo geométrico associado ao processo de captura da imagem e seus principais parâmetros.

2.2.1 Pontos e retas no espaço

Neste trabalho, pontos e retas definidos no espaço bidimensional da imagem são chamados *pontos na imagem* e *retas na imagem* (ou 2D) para diferenciá-los dos pontos e retas definidos no espaço tridimensional da cena, os chamados *pontos no espaço* e *retas no espaço* (ou 3D).

Um ponto no espaço pode ser representado por um vetor pertencente a \mathbb{R}^3 . Assim sendo, um ponto \mathcal{P} pertencente à cena e descrito no sistema de coordenadas da câmera \mathcal{O}^c é representado pelo vetor

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \quad (2.6)$$

Alternativamente, \mathcal{P} pode ser representado por um vetor $\tilde{\mathbf{p}}$ pertencente a \mathbb{R}^4 e que satisfaça a condição $\mathbf{k}^T \tilde{\mathbf{p}} = 1$, com $\mathbf{k} = (0, 0, 0, 1)^T$, ou seja, o vetor de coordenadas homogêneas [63]

$$\tilde{\mathbf{p}} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}. \quad (2.7)$$

A distância de \mathcal{P} à origem do sistema \mathcal{O}^c é dada por

$$d = \sqrt{x^2 + y^2 + z^2} = \|\mathbf{p}\| = \sqrt{\|\tilde{\mathbf{p}}\|^2 - 1}. \quad (2.8)$$

As retas possuem diversas representações possíveis no espaço da cena. Dentre as diversas representações já propostas [62], destacam-se as representações de Denavit-Hartenberg e Hayati-Roberts por serem mínimas. Elas necessitam de apenas 4 parâmetros para descrever os 4 graus de liberdade que as retas possuem no espaço 3D. No entanto, tais representações não são muito intuitivas nem relacionam-se facilmente com as representações para pontos e planos, tornando as representações não-mínimas mais atraentes para várias aplicações [64].

Uma representação bastante natural para a reta é dada por um ponto \mathcal{P}_0 de coordenadas \mathbf{p}_0 e um vetor de orientação \mathbf{n} , na forma

$$\mathbf{p}_0 = (p_{0x} \quad p_{0y} \quad p_{0z})^T, \quad (2.9)$$

$$\mathbf{n} = (n_x \quad n_y \quad n_z)^T. \quad (2.10)$$

Então a reta \mathcal{L} pode ser parametrizada pela variável $\mu \in \mathbb{R}$, de forma que seja expressa como um conjunto de pontos no espaço tridimensional

$$\mathcal{L} = \{ \mathcal{P} \mid \mathbf{p} = \mathbf{p}_0 + \mu \mathbf{n}, \quad -\infty < \mu < \infty \}, \quad (2.11)$$

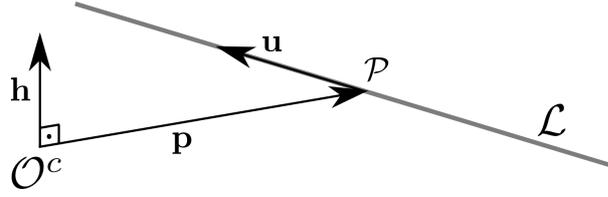


Figura 2.6: Representação da reta 3D utilizando coordenadas de Plücker.

No entanto, esta representação sofre de falta de unicidade pois são possíveis infinitas escolhas de \mathbf{p}_0 e infinitas representações para a orientação \mathbf{n} que irão descrever a mesma reta \mathcal{L} . O vetor de orientação \mathbf{n} também é indeterminado quanto ao sinal, pois \mathbf{n} e $-\mathbf{n}$ descrevem a mesma reta.

Assim sendo, é mais conveniente uma outra representação com 6 parâmetros definida por Plücker [65] que garante unicidade e é particularmente interessante por ser feita no espaço projetivo, assim como a transformação de imagens na câmera, que também é projetiva (cf. Seção 2.2.3).

A representação de retas 3D por coordenadas de Plücker (cf. Figura 2.6) consiste de um vetor $\lambda \in \mathbb{R}^6$, dividido em duas componentes na forma

$$\lambda = \begin{pmatrix} \mathbf{u} \\ \mathbf{h} \end{pmatrix}, \quad (2.12)$$

em que $\mathbf{u} = (u_x \ u_y \ u_z)^T$ é um vetor unitário de orientação da reta \mathcal{L} e $\mathbf{h} = (h_x \ h_y \ h_z)^T$ é dado por

$$\mathbf{h} = \mathbf{p} \times \mathbf{u}, \quad (2.13)$$

em que \mathbf{p} é um ponto qualquer pertencente à reta. De (2.13) conclui-se que \mathbf{h} é um vetor normal ao plano que passa por \mathcal{L} e pela origem do sistema de eixos coordenados, conhecido como plano de interpretação. Como conseqüências dessa representação, tem-se as restrições

$$\mathbf{h}^T \mathbf{u} = 0 \quad \text{e} \quad \|\mathbf{u}\| = 1. \quad (2.14)$$

Deve ser observado que o vetor \mathbf{h} independe de qual ponto \mathbf{p} se escolha sobre a reta e que sua norma $\|\mathbf{h}\|$ é igual à distância entre a reta e a origem do sistema de coordenadas, ou seja, é igual à profundidade da reta. A independência de \mathbf{h} quanto à escolha de \mathbf{p} é demonstrada a seguir. Um ponto qualquer sobre uma reta \mathcal{L} com parâmetros em coordenadas de Plücker $\lambda = \begin{pmatrix} \mathbf{u} \\ \mathbf{h} \end{pmatrix}$ deve obedecer à relação

$$\mathbf{p} = \mathbf{p}_0 + \mu \mathbf{u}, \quad (2.15)$$

em que \mathbf{p}_0 é um ponto qualquer pertencente à mesma reta e $\mu \in \mathbb{R}$. Substituindo (2.15) em (2.13), tem-se

$$\mathbf{h} = (\mathbf{p}_0 + \mu \mathbf{u}) \times \mathbf{u} = \mathbf{p}_0 \times \mathbf{u} + \underbrace{\mu \mathbf{u} \times \mathbf{u}}_{\mathbf{0}} = \mathbf{p}_0 \times \mathbf{u}, \quad (2.16)$$

que é independente de μ e, portanto, independe da escolha de \mathbf{p} .

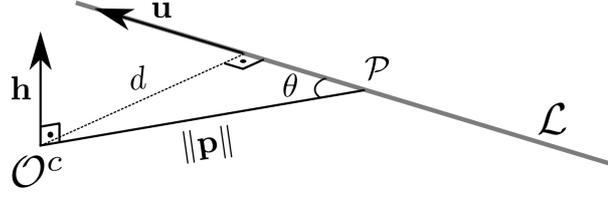


Figura 2.7: Determinação da profundidade da porta d

A prova de que a norma $\|\mathbf{h}\|$ corresponde à distância entre a reta e o eixo coordenado vem da própria definição do produto vetorial que define que o vetor resultante é perpendicular ao plano formado pelos vetores iniciais e tem módulo dado por

$$\|\mathbf{h}\| = \|\mathbf{p} \times \mathbf{u}\| = \|\mathbf{p}\|\|\mathbf{u}\| \sin(\theta), \quad (2.17)$$

em que θ é o ângulo entre \mathbf{p} e \mathbf{u} tal que $0 \leq \theta \leq \pi$ (cf. Figura 2.7).

Como $\|\mathbf{u}\| = 1$, (2.17) resulta em

$$\|\mathbf{h}\| = \|\mathbf{p}\| \sin(\theta) = d. \quad (2.18)$$

Para tornar a representação por coordenadas de Plücker única, resta eliminar a ambiguidade do sinal do vetor de orientação. Uma maneira simples de fazer isso é exigir que o sinal de \mathbf{u} seja determinado de forma que $u_z < 0$. Caso $u_z = 0$, escolhe-se o sinal tal que $u_y < 0$. E finalmente, se ambos $u_z = u_y = 0$, faz-se $u_x = -1$. Com isso, uma reta será determinada de forma única pela representação de Plücker. Entretanto, há uma singularidade no caso em que $\mathbf{h} = 0$, que acontece quando \mathcal{L} passa pela origem e se projeta como um ponto na imagem. Assume-se que isso não ocorre pois trata-se de um caso degenerado de visão computacional.

Uma grande vantagem do uso da representação de Plücker em problemas de visão computacional é o fato de a projeção ℓ da reta \mathcal{L} ter uma relação bastante direta com o vetor \mathbf{h} . Conforme o demonstrado no Capítulo 3, o vetor formado pelos parâmetros da reta ℓ é igual ao vetor \mathbf{h} normalizado, representado por $\underline{\mathbf{h}}$. Com isso, a componente \mathbf{h} é trivialmente determinada a partir das imagens da câmera.

2.2.2 Pontos e retas na imagem

Pontos na imagem são representados por um vetor pertencente a \mathbb{R}^2 e composto por suas coordenadas (x, y) em relação ao sistema de eixos coordenados \mathcal{O}^i , fixo na imagem. Assim sendo, um ponto \mathcal{P} projetado na imagem em \mathcal{P}^i é representado por:

$$\mathbf{p}^i = \begin{pmatrix} x \\ y \end{pmatrix}. \quad (2.19)$$

Uma representação alternativa é utilizar um vetor $\tilde{\mathbf{p}}^i$ pertencente a \mathbb{R}^3 , tal que $\mathbf{k}^T \tilde{\mathbf{p}}^i = 1$, com $\mathbf{k} = (0, 0, 1)^T$, ou seja, o vetor de coordenadas homogêneas

$$\tilde{\mathbf{p}}^i = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}. \quad (2.20)$$

A distância de \mathcal{P}^i à origem é dada por

$$d = \sqrt{x^2 + y^2} = \sqrt{\|\mathbf{p}^i\|^2 - 1}. \quad (2.21)$$

Uma reta na imagem é representada por sua equação

$$ax + by + c = 0. \quad (2.22)$$

No entanto, esta representação não é única, pois (2.22) multiplicada por um fator de escala ou com o sinal invertido continua representando a mesma reta na imagem. Assim sendo, uma reta na imagem pode ser representada pelo vetor

$$\boldsymbol{\lambda}^i = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \quad (2.23)$$

contanto que $\boldsymbol{\lambda}^i \neq \mathbf{k}$ pois a e b não podem ser iguais a zero simultaneamente e $\|\boldsymbol{\lambda}^i\| = 1$ e $a > 0$ para evitar multiplicidade de representação.

A distância de $\boldsymbol{\lambda}^i$ até a origem é dada por

$$d = \frac{|c|}{\sqrt{a^2 + b^2}} = \frac{|\mathbf{k}^T \boldsymbol{\lambda}^i|}{\sqrt{1 - (\mathbf{k}^T \boldsymbol{\lambda}^i)^2}}. \quad (2.24)$$

2.2.3 Modelo Geométrico de Projeção

Existem diversos modelos para dispositivos para captura de imagens, sendo o modelo *pinhole*² o mais simples deles. Seu nome deve-se ao seu modo de funcionamento, que faz uso de um minúsculo orifício de entrada para focalizar os raios de luz e com isso projetar as imagens capturadas sobre uma superfície. Os primeiros modelos desse tipo de câmera datam do século XVI e foram progressivamente substituídos por modelos mais sofisticados que, graças à adoção de lentes, diafragmas e outras inovações, permitem regular diversas características da imagem como nitidez, foco e profundidade de campo. No entanto, o modelo *pinhole* de projeção perspectiva, inicialmente proposto por Brunelleschi no início do século XV, ainda é matematicamente conveniente pois apesar de sua simplicidade, este modelo fornece aproximações bastante razoáveis do processo de captura de imagem [66].

A geometria do modelo *pinhole* está representada na Figura 2.8(a). Observa-se o plano π , chamado plano imagem, sobre o qual os objetos do mundo real são projetados. A linha tracejada horizontal (eixo Z^c) é o eixo óptico e a lente da câmera é posicionada perpendicularmente a ele na

²A designação *pinhole* vem do termo em inglês, *pin-hole*, que significa "buraco de alfinete".

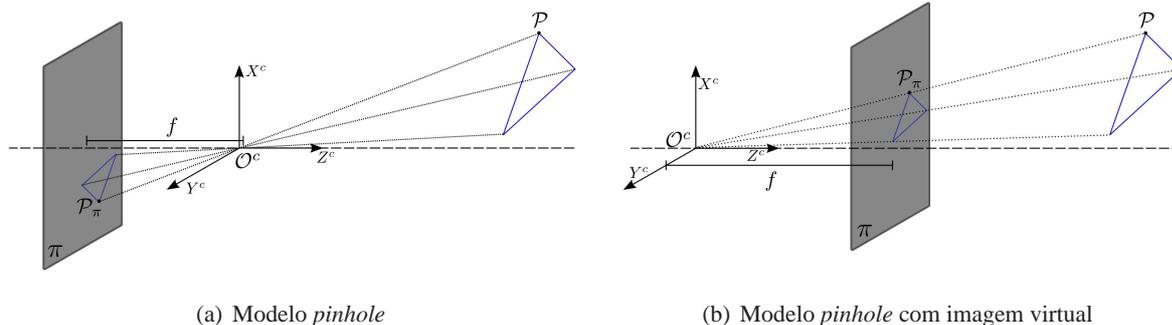


Figura 2.8: Modelo para geometria da câmera

origem do sistema de coordenadas \mathcal{O}^c , também chamada de centro óptico. O comprimento focal f é um parâmetro das lentes.

Neste modelo, considera-se que todos os raios de luz captados do ambiente atravessam a lente da câmera em um único ponto, o centro óptico. Os raios de luz são então projetados sobre o plano π , formando uma imagem invertida dos objetos. Para fins de simplificação do modelo, costuma-se considerar uma imagem virtual que está associada a um plano colocado à frente do centro óptico à mesma distância f (cf. Figura 2.8(b)).

2.2.4 Projeção da imagem

O modelo matemático de uma câmera pode ser caracterizado com a ajuda de duas transformações [67]: uma projeção que leva um ponto do espaço 3D para o espaço 2D projetado sobre o plano imagem, e uma transformação de coordenadas do sistema \mathcal{O}^c para o sistema \mathcal{O}^i , geralmente acrescida de uma conversão de unidades (cf. Figura 2.9).

Assim sendo, um ponto \mathcal{P} pertencente à cena e descrito no sistema de coordenadas da câmera \mathcal{O}^c por $\mathbf{p} = (x, y, z)^T$ é projetado no plano imagem em \mathcal{P}_π , cujas coordenadas representadas em \mathcal{O}^c podem ser encontradas à partir da Figura 2.8(b), utilizando semelhança de triângulos [1, 68]:

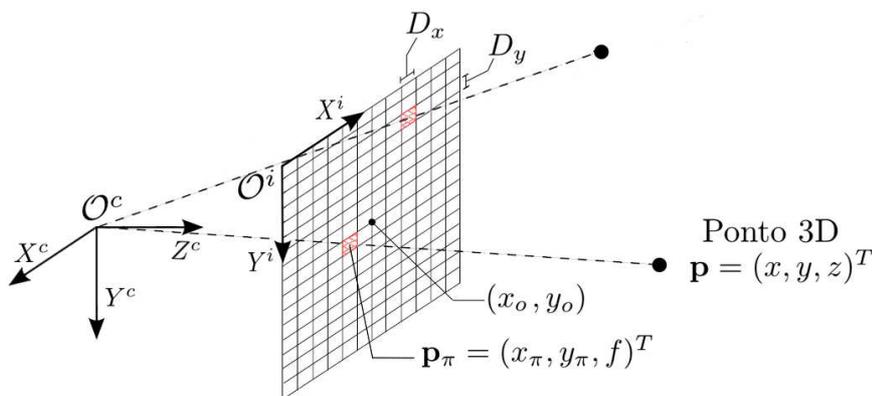


Figura 2.9: Esquemático da projeção de um ponto na imagem

$$\mathbf{p}_\pi = \begin{pmatrix} \frac{fx}{z} \\ \frac{fy}{z} \\ f \end{pmatrix} = f \frac{\mathbf{P}}{z}. \quad (2.25)$$

A transformação do ponto \mathcal{P}_π do sistema \mathcal{O}^c para o sistema \mathcal{O}^i é dada por uma rotação uma translação (*cf.* Anexo A., para mais detalhes sobre transformação de sistemas de coordenadas).

$$\mathbf{R}_c^i = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{t}_c^i = \begin{pmatrix} x_o \\ y_o \\ 1 \end{pmatrix},$$

em que x_o e y_o são as coordenadas do centro da imagem medidas em *pixels* e representadas no sistema \mathcal{O}^i .

Como o sistema original tem unidades no sistema métrico e os pontos na imagem são dados em *pixels*, acrescenta-se à transformação uma mudança de escala dada pela matriz diagonal \mathbf{D} , cujos elementos D_x e D_y representam os fatores de escala na horizontal e na vertical, respectivamente. Com efeito, os *pixels* de uma câmera raramente são quadrados [67], o que justifica a existência de fatores de escala diferentes nas direções x e y .

Com a combinação das duas transformações descritas, chega-se ao seguinte modelo final

$$\begin{aligned} \tilde{\mathbf{p}}^i &= \begin{pmatrix} D_x & 0 & 0 \\ 0 & D_y & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{p}_\pi + \begin{pmatrix} x_o \\ y_o \\ 1 \end{pmatrix}, \\ \tilde{\mathbf{p}}^i &= \begin{pmatrix} -D_x & 0 & x_o \\ 0 & D_y & y_o \\ 0 & 0 & \frac{1}{f} \end{pmatrix} \mathbf{p}_\pi, \\ \tilde{\mathbf{p}}^i &= \mathbf{K} \frac{\mathbf{P}}{z}, \end{aligned} \quad (2.26)$$

em que \mathbf{K} é conhecida como a matriz de calibração da câmera e é dada por

$$\mathbf{K} = \begin{pmatrix} -fD_x & 0 & x_o \\ 0 & fD_y & y_o \\ 0 & 0 & 1 \end{pmatrix}.$$

3 CONTROLE SERVO-VISUAL PARA APROXIMAÇÃO DE PORTAS

I can't understand why people are frightened of new ideas. I'm frightened of the old ones.

John Cage (1912 - 1992)

Este capítulo apresenta a metodologia utilizada para resolver a tarefa de aproximação de portas por um robô móvel, além do projeto dos dois controladores servo visuais avaliados neste trabalho. Em linhas gerais, propõe-se uma tarefa de navegação que incorpora simultaneamente controle de trajetória e mapeamento local buscando realizar a *docking* com o mínimo de conhecimento *a priori* sobre o alvo, no caso uma porta.

A partir das imagens do ambiente, estimam-se os parâmetros 3D do alvo e estes são utilizados paralelamente para atualizar o controlador servo-visual. A ação de controle, por sua vez, move o robô de forma a aproximá-lo do alvo e ao mesmo tempo, excita o sistema com novos parâmetros para a estimação. A Figura 3.1 apresenta um diagrama em blocos de cada etapa do processo descrito neste capítulo. O procedimento referente ao Bloco 1 é descrito brevemente na Seção 3.6. O Bloco 2 é apresentado na Seção 3.5. O conteúdo do Bloco 3 é dado pela Seção 3.4 e o Bloco 4 está detalhado na Seção 3.3.

3.1 ESPAÇOS \mathcal{F} E \mathcal{T} PARA O DOCKING DE PORTAS

Classicamente, o problema de controle servo-visual considera que objetos relevantes da cena sejam bem caracterizados no espaço de trabalho por modelos geométricos. No caso específico de uma porta, duas possibilidades foram estudadas para sua representação: quatro pontos referentes às extremidades superiores e inferiores de seus umbrais, ou duas retas verticais correspondentes. Comparados a pontos, segmentos de reta têm propriedades interessantes para a construção de uma representação do ambiente pois eles fornecem mais informação da estrutura da cena, podem ser facilmente detectados em ambientes estruturados ou semi-estruturados [69] e ainda apresentam robustez à obstrução parcial. Assim sendo, optou-se pela segunda alternativa, em que utiliza-se o modelo por retas para representar a porta alvo do *docking*.

Dentre as várias representações matemáticas existentes para retas no espaço, optou-se pelo uso das coordenadas de Plücker devido às vantagens apresentadas no Capítulo 2. Assim sendo, uma porta é representada por um conjunto de dois vetores λ , referentes a cada um de seus umbrais. Estes vetores devem ser estimados a partir das imagens obtidas pelas câmeras e das relações determinadas geometricamente entre o movimento do robô e os parâmetros de λ .

O ambiente onde o robô e a porta estão inseridos é um espaço de trabalho do tipo $\mathbb{R}^3 \times SO^3 = SE^3$, de 6 dimensões. No entanto, robôs móveis terrestres movimentam-se apenas no plano do chão. Desta forma, temos que a postura \mathbf{r} do robô está definida no espaço de configuração $\mathcal{T} = \mathbb{R}^2 \times SO^1 = SE^2$, ou seja, tem-se $m = 3$ graus de liberdade. Já as dimensões do espaço de características da imagem $\mathcal{F} = \mathbb{R}^k$ dependem do tipo de características da imagem escolhidas. Caso s seja constituído

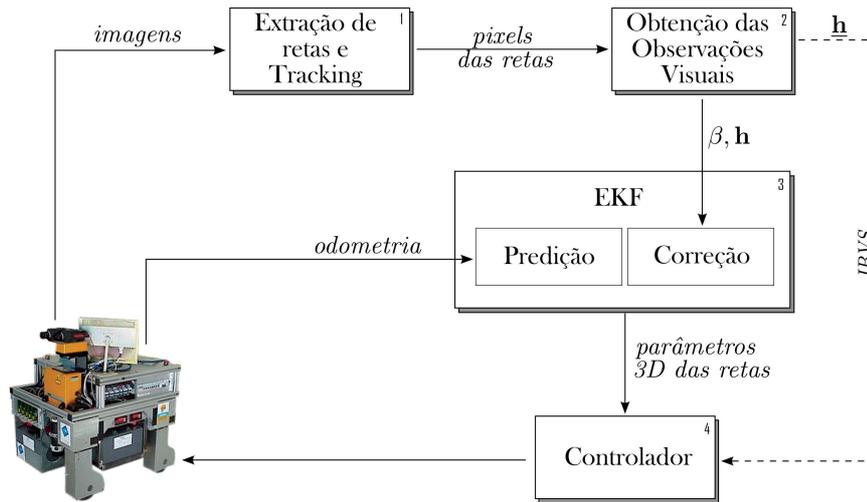


Figura 3.1: Diagrama em blocos da solução proposta.

pelos parâmetros 3D das retas na representação de (2.12), tem-se $k = 6$, e caso s seja dado pela projeção 2D das retas na imagem representadas segundo (2.23), tem-se $k = 3$.

3.2 SISTEMAS DE COORDENADAS \mathcal{O}^C , \mathcal{O}^D E \mathcal{O}^R

Neste trabalho, é utilizado um sistema de visão binocular, constituído de duas câmeras dispostas lado a lado, de forma a se obter obter informação de profundidade (*cf.* Figura 3.2(b)). O sistema de câmeras fica fixo sobre o robô e ambos movem-se em conjunto (configuração *eye-in-hand*). Esta configuração, embora forneça uma visão parcial do espaço de trabalho, é recomendada para a tarefa de *docking* por proporcionar maior precisão. Como robô e sistema de câmeras movem-se em conjunto, a transformação de coordenadas entre robô e câmeras passa a ser um parâmetro fixo do sistema, dependendo unicamente do ajuste inicial das câmeras sobre o suporte do robô.

Convencionou-se utilizar a representação das retas da porta sempre em referência ao sistema de coordenadas da câmera esquerda \mathcal{O}^c , de forma a facilitar a interpretação dos parâmetros das imagens fornecidas pelo sistema de visão computacional. Por isso, o sobrescrito c é dispensável para coordenadas representadas no sistema \mathcal{O}^c . Nesta seção, optou-se por manter o sobrescrito c apenas para que as equações de transformação ficassem mais claras. No entanto ele é omitido no restante do trabalho.

Assim sendo, é necessário converter as informação obtidas na câmera direita que estão no sistema de coordenadas \mathcal{O}^d para o sistema da câmera esquerda. Além do mais, os dados de velocidade fornecidos pela odometria são referentes ao sistema de coordenadas do robô \mathcal{O}^r e também devem ser convertidos para o sistema de coordenadas adotado como padrão. Por isso, é importante definir as relações de transformação de um sistema de coordenadas para outro, de forma a manter consistência entre as representações. As transformações do sistema de coordenadas \mathcal{O}^c para um sistema \mathcal{O}^r são descritas pela combinação de uma matriz de rotação \mathbf{R}_c^r e um vetor de translação \mathbf{t}_c^r . No caso do sistema representado na Figura 3.2(a), \mathbf{R}_c^r e \mathbf{t}_c^r são dados por

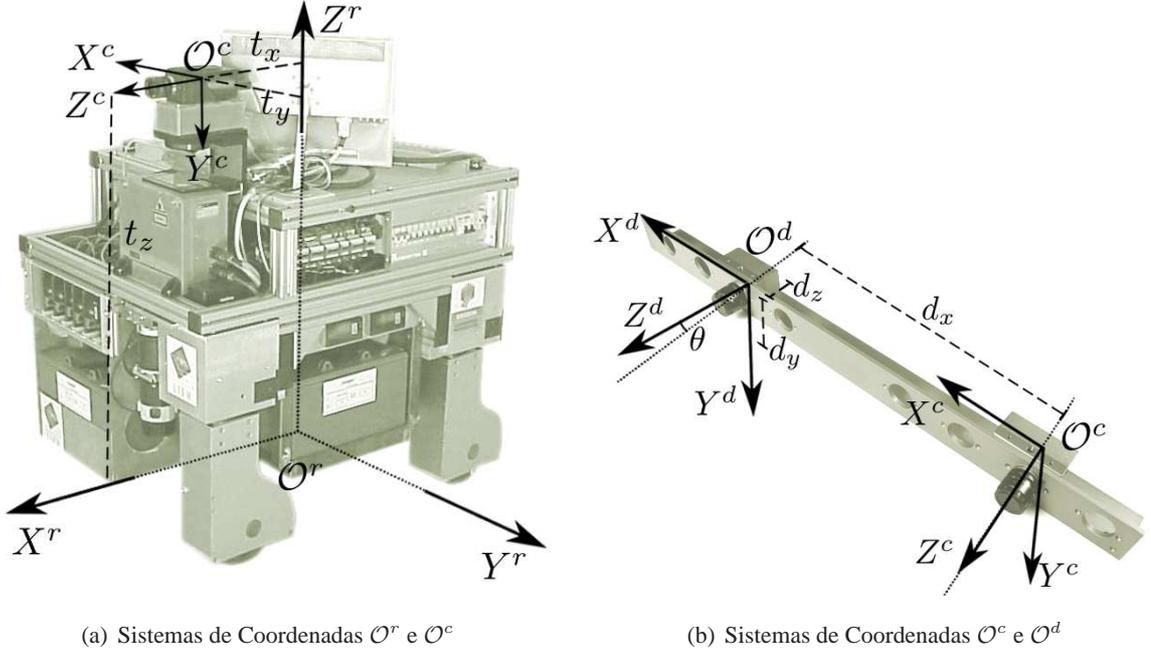


Figura 3.2: Relação entre os sistemas de coordenadas do robô e das câmeras

$$\mathbf{R}_c^r = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{t}_c^r = \begin{pmatrix} t_x \\ -t_y \\ t_z \end{pmatrix}, \quad (3.1)$$

em que t_x , t_y e t_z são as distâncias entre as origens dos dois sistemas em cada eixo

Da mesma forma, para o caso de uma configuração binocular como a representada na Figura 3.2(b), a transformação do sistema O^c para O^d é descrita pela combinação de \mathbf{R}_c^d e \mathbf{t}_c^d dados por

$$\mathbf{R}_c^d = \begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{pmatrix}, \quad \mathbf{t}_c^d = \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix}, \quad (3.2)$$

em que d_x , d_y e d_z são as distâncias entre as origens dos dois sistemas em cada eixo e o ângulo θ representa a diferença de alinhamento entre os eixos ópticos.

Para fazer as transformações inversas, ou seja, de um sistema qualquer O^x para O^c , as matrizes de rotação e translação são dadas por [70]

$$\mathbf{R}_x^c = (\mathbf{R}_c^x)^T, \quad \mathbf{t}_x^c = -(\mathbf{R}_x^c)^T \mathbf{t}_x^r \quad (3.3)$$

A transformação de um ponto de coordenadas \mathbf{p}^c no sistema da camera para outro sistema é dada por [70]

$$\mathbf{p}^x = \mathbf{R}_c^x \mathbf{p}^c + \mathbf{t}_c^x. \quad (3.4)$$

Sendo ν^c o vetor de velocidades em 3D da câmera esquerda definido por

$$\boldsymbol{\nu}^c = \begin{pmatrix} \mathbf{v}^c \\ \boldsymbol{\omega}^c \end{pmatrix}, \quad (3.5)$$

com $\mathbf{v}^c = (v_x^c \ v_y^c \ v_z^c)^T$ referente às velocidades de translação e $\boldsymbol{\omega}^c = (\omega_x^c \ \omega_y^c \ \omega_z^c)^T$ referente às velocidades de translação, a transformação de coordenadas de \mathcal{O}^c para um sistema qualquer \mathcal{O}^x é dada por [56]

$$\begin{aligned} \boldsymbol{\omega}^x &= \mathbf{R}_c^x \boldsymbol{\omega}^c, \\ \mathbf{v}^x &= \mathbf{R}_c^x \mathbf{v}^c + \mathbf{R}_c^x \mathbf{t}_c^x \times \boldsymbol{\omega}^c, \\ \boldsymbol{\nu}^x &= \begin{pmatrix} \mathbf{R}_c^x & \mathbf{R}_c^x \mathbf{A}s(\mathbf{t}_c^x) \\ \mathbf{0} & \mathbf{R}_c^x \end{pmatrix} \boldsymbol{\nu}^c, \end{aligned}$$

em que $\mathbf{A}s(\mathbf{x})$ representa a matriz anti-simétrica do vetor $\mathbf{x} = (x_x, x_y, x_z)^T$, dada pela fórmula

$$\mathbf{A}s(\mathbf{x}) = \begin{pmatrix} 0 & -x_z & x_y \\ x_z & 0 & -x_x \\ -x_y & x_x & 0 \end{pmatrix}. \quad (3.6)$$

No entanto, como discutido na Seção 3.1, um robô móvel terrestre move-se de fato apenas num plano bidimensional, de forma que v_z^r, ω_x^r e ω_y^r são sempre iguais a zero. Robôs terrestres omnidirecionais não sofrem restrições holonômicas e portanto, possuem componentes de velocidade translacional em ambas direções do plano. Assim sendo, tem-se o vetor de velocidades do robô omnidirecional é dado por

$$\dot{\mathbf{r}} = \begin{pmatrix} v_x^r \\ v_y^r \\ \omega_z^r \end{pmatrix}, \quad (3.7)$$

que correspondem às componentes de velocidade translacional do robô nos eixos X^r e Y^r e sua componente de velocidade rotacional no eixo Z^r , definidos de acordo com o sistema coordenado da Figura 3.2(a).

Devido ao acoplamento rígido da configuração *eye-in-hand* utilizada, tem-se a seguinte relação invariante no tempo entre o vetor de velocidades $\boldsymbol{\nu}^c$ e a derivada da postura do robô :

$$\boldsymbol{\nu}^c = \mathbf{M} \dot{\mathbf{r}}, \quad (3.8)$$

com

$$\mathbf{M} = \begin{pmatrix} \mathbf{R}_r^c & \mathbf{R}_r^c \mathbf{A}s(\mathbf{t}_r^c) \\ \mathbf{0}_{3 \times 3} & \mathbf{R}_r^c \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.9)$$

Para robôs a tração diferencial, ao contrário do caso omnidirecional, há presença de restrição de movimento de forma que a componente v_y^r é sempre nula. Com isso, tem-se que $\dot{\mathbf{r}}$ seria um vetor de apenas duas dimensões com componentes v_x^r e ω_z^r e conseqüentemente, $\mathbf{M} \in \mathbb{R}^{6 \times 2}$.

A transformação entre sistemas de coordenadas para o vetor de parâmetros λ que representa uma reta \mathcal{L} no espaço também pode ser facilmente derivada [71]. De (2.15) tem-se que $\mathbf{u}^x = \frac{\mathbf{p}^x - \mathbf{p}_0^x}{\mu}$. Substituindo \mathbf{p}^x e \mathbf{p}_0^x pela relação (3.4) chega-se a

$$\mathbf{u}^x = \mathbf{R}_c^x \mathbf{u}^c. \quad (3.10)$$

De (2.13), tem-se $\mathbf{h}^x = \mathbf{p}^x \times \mathbf{u}^x$ que, utilizando (3.4) e (3.10), resulta em

$$\begin{aligned} \mathbf{h}^x &= (\mathbf{R}_c^x \mathbf{p}^c + \mathbf{t}_c^x) \times \mathbf{R}_c^x \mathbf{u}^c \\ &= \mathbf{R}_c^x \mathbf{p}^c \times \mathbf{R}_c^x \mathbf{u}^c + \mathbf{t}_c^x \times \mathbf{R}_c^x \mathbf{u}^c \\ &= \mathbf{R}_c^x (\mathbf{p}^c \times \mathbf{u}^c) + \mathbf{t}_c^x \times \mathbf{R}_c^x \mathbf{u}^c, \\ &= \mathbf{R}_c^x \mathbf{h}^c + \mathbf{t}_c^x \times \mathbf{R}_c^x \mathbf{u}^c. \end{aligned} \quad (3.11)$$

Assim, a transformação do vetor de parâmetros λ^c do sistema da câmera para o sistema do robô é dada por

$$\lambda^x = \begin{pmatrix} \mathbf{R}_c^x & \mathbf{0}_{3 \times 3} \\ \mathbf{As}(\mathbf{t}_c^x) \mathbf{R}_c^x & \mathbf{R}_c^x \end{pmatrix} \lambda^c. \quad (3.12)$$

3.3 CONTROLE SERVO-VISUAL DE VELOCIDADE

Como definido na seção (2.1.3), o controle servo visual consiste em minimizar um erro

$$\mathbf{e} = \mathbf{s}^* - \mathbf{s}, \quad (3.13)$$

em que \mathbf{s} e \mathbf{s}^* são respectivamente o vetor de características da imagem e seu valor final desejado, definidos conforme a abordagem de controle escolhida.

Optou-se por uma arquitetura do tipo *look-and-move*, cuja ação de controle é da forma

$$\dot{\mathbf{r}} = -\mathbf{K}\mathbf{e}, \quad (3.14)$$

com $\dot{\mathbf{r}}$ sendo o vetor de velocidades do robô dado por (3.7).

A partir de (2.2) e (3.14), tem-se

$$\dot{\mathbf{e}} = -\dot{\mathbf{s}} = -\mathbf{L}_s \dot{\mathbf{r}} = \mathbf{L}_s \mathbf{K}\mathbf{e}, \quad (3.15)$$

em que considera-se $\dot{\mathbf{s}}^* = \mathbf{0}$. Portanto, a dinâmica do erro é definida pela matriz \mathbf{D} dada por

$$\mathbf{D} = \mathbf{L}_s \mathbf{K}. \quad (3.16)$$

Considerando $\mathbf{D} = p\mathbf{I}_k$, em que p equivale à posição dos pólos do sistema em malha fechada e k é o número de componentes do vetor \mathbf{s} , deve-se escolher os pólos de forma que o controlador servo-sivual seja mais lento que o controlador de velocidade da malha mais interna do controle em cascata.

A partir de (3.16), pode-se obter uma expressão para \mathbf{K} utilizando-se da solução geral para equações matriciais não-homogêneas [60], resultando em

$$\mathbf{K} = p\mathbf{L}_s^\dagger + (\mathbf{I}_3 - \mathbf{L}_s^\dagger \mathbf{L}_s) \mathbf{b}, \quad (3.17)$$

em que \mathbf{L}_s^\dagger é a matriz pseudo-inversa de \mathbf{L}_s dada em (2.3) e \mathbf{b} é um vetor arbitrário de dimensão $3 \times k$. Tem-se que o termo $p\mathbf{L}_s^\dagger$ é por si só uma solução para \mathbf{K} . O termo adicional presente na solução generalizada dada em (3.17) é uma ponderação do erro de aproximação da pseudo-inversa e embora não seja explorado neste trabalho, poderia vir a ser usado para forçar trajetórias que melhorem a excitação do estimador estocástico, por exemplo. Substituindo (3.17) em (3.14), tem-se

$$\dot{\mathbf{r}} = -[p\mathbf{L}_s^\dagger + (\mathbf{I}_3 - \mathbf{L}_s^\dagger \mathbf{L}_s) \mathbf{b}] \mathbf{e}, \quad (3.18)$$

sendo p e \mathbf{b} parâmetros ajustáveis do controlador.

Em ambas abordagens de controle aqui estudadas, o sinal de controle é dado por (3.18), variando-se apenas a forma escolhida para o vetor \mathbf{s} . No caso do controle baseado em posição (*PBVS*) define-se $\mathbf{s} = \begin{pmatrix} \lambda_1^T & \lambda_2^T \end{pmatrix}^T$, correspondente ao vetor de coordenadas 3D das duas retas da porta, enquanto para o controle baseado em imagem (*IBVS*) define-se $\mathbf{s} = \begin{pmatrix} \underline{\mathbf{h}}_1^T & \underline{\mathbf{h}}_2^T \end{pmatrix}^T$, correspondente aos parâmetros a , b e c das retas projetadas na imagem.

3.3.1 Controle *IBVS*

O controle *IBVS* utiliza parâmetros 2D disponíveis da imagem do ambiente como referência para a ação de controle. No caso de um par de retas, uma possível escolha é fazer $\mathbf{s} = \begin{pmatrix} \underline{\mathbf{h}}_1^T & \underline{\mathbf{h}}_2^T \end{pmatrix}^T$. Com esta representação para \mathbf{s} , os parâmetros observados para cada reta são os três coeficientes da equação da reta projetada ℓ . Portanto, para descrever a influência do movimento da câmera na projeção é necessário determinar a relação entre $\dot{\underline{\mathbf{h}}}$ e $\dot{\mathbf{r}}$ [47].

Tem-se que $\underline{\mathbf{h}} = \frac{\mathbf{h}}{\|\mathbf{h}\|}$, logo

$$\dot{\underline{\mathbf{h}}} = \frac{\dot{\mathbf{h}} \|\mathbf{h}\| - \mathbf{h} \frac{d\|\mathbf{h}\|}{dt}}{\|\mathbf{h}\|^2} = \frac{\dot{\mathbf{h}} - \underline{\mathbf{h}} \frac{d\|\mathbf{h}\|}{dt}}{\|\mathbf{h}\|}. \quad (3.19)$$

Como $\|\mathbf{h}\| = \sqrt{\mathbf{h}^T \mathbf{h}}$, então $\frac{d\|\mathbf{h}\|}{dt} = \frac{\mathbf{h}^T \dot{\mathbf{h}}}{\sqrt{\mathbf{h}^T \mathbf{h}}} = \underline{\mathbf{h}}^T \dot{\mathbf{h}}$. Utilizando o resultado de (3.38), tem-se

$$\frac{d\|\mathbf{h}\|}{dt} = \underline{\mathbf{h}}^T (\mathbf{h} \times \boldsymbol{\omega}) + \underline{\mathbf{h}}^T (\mathbf{u} \times \mathbf{v}), \quad (3.20)$$

Pela propriedade de permutação circular dos produtos vetoriais mistos,

$$\underline{\mathbf{h}}^T(\mathbf{h} \times \boldsymbol{\omega}) = \boldsymbol{\omega}^T(\underline{\mathbf{h}} \times \mathbf{h}) = 0, \quad (3.21)$$

e portanto,

$$\frac{d\|\mathbf{h}\|}{dt} = \underline{\mathbf{h}}^T(\mathbf{u} \times \mathbf{v}). \quad (3.22)$$

Substituindo (3.22) em (3.19) obtem-se

$$\begin{aligned} \underline{\dot{\mathbf{h}}} &= \frac{(\mathbf{u} \times \mathbf{v} + \mathbf{h} \times \boldsymbol{\omega}) - \underline{\mathbf{h}}\underline{\mathbf{h}}^T(\mathbf{u} \times \mathbf{v})}{\|\mathbf{h}\|}, \\ \underline{\dot{\mathbf{h}}} &= \underline{\mathbf{h}} \times \boldsymbol{\omega} + \frac{(\mathbf{I}_{3 \times 3} - \underline{\mathbf{h}}\underline{\mathbf{h}}^T)(\mathbf{u} \times \mathbf{v})}{\|\mathbf{h}\|}. \end{aligned} \quad (3.23)$$

Assim sendo, a evolução dos parâmetros da projeção ℓ é dada por

$$\underline{\dot{\mathbf{h}}} = \mathbf{L}_h \boldsymbol{\nu}, \quad \mathbf{L}_h = \begin{pmatrix} \frac{(\mathbf{I}_{3 \times 3} - \underline{\mathbf{h}}\underline{\mathbf{h}}^T)}{\|\mathbf{h}\|} A_s(\mathbf{u}) & A_s(\underline{\mathbf{h}}) \end{pmatrix}, \quad (3.24)$$

em que $\boldsymbol{\nu}$ é dado por (3.5).

Por fim, \mathbf{L}_s relacionando o vetor de características com a velocidade do robô $\dot{\mathbf{r}}$ é definido realizando a transformação dada por (3.8), resultando em

$$\mathbf{L}_s = \mathbf{L}_h \mathbf{M}. \quad (3.25)$$

3.3.2 Controle *PBVS*

O controle *PBVS* típico utiliza a postura \mathbf{r} da câmera expressa no espaço cartesiano como referência para a ação de controle. No entanto, este tipo de controle pode ser feito a partir de um conjunto de parâmetros 3D estimados a partir informações extraídas das imagens. Uma escolha interessante é fazer $\mathbf{s} = \begin{pmatrix} \lambda_1^T & \lambda_2^T \end{pmatrix}^T$, em que os parâmetros observados para cada reta são as três componentes de \mathbf{u} e as três componentes de \mathbf{h} , cujas derivadas no tempo se relacionam com o movimento da câmera conforme descrito em [43] e demonstrado nas relações a seguir.

Suponha um ponto \mathcal{P} qualquer pertencente à reta tridimensional \mathcal{L} , expresso no sistema de coordenadas de uma câmera. Se o movimento da câmera é descrito através do vetor de velocidades $\boldsymbol{\nu}$ definido em (3.5), tem-se que o movimento relativo de \mathcal{P} será dado por (3.26)

$$\dot{\mathbf{p}} = -\mathbf{v} - \boldsymbol{\omega} \times \mathbf{p}. \quad (3.26)$$

Dado um ponto \mathcal{P}_0 de coordenadas \mathbf{p}_0 pertencente a \mathcal{L} e um vetor de orientação \mathbf{u} , um segmento de reta pode ser definido pela equação

$$(\mathbf{p} - \mathbf{p}_0) \times \mathbf{u} = 0 \quad (3.27)$$

em que

$$\mathbf{p} = \mathbf{p}_0 + \mu \mathbf{u}, \quad (3.28)$$

com $\mu \in \mathbb{R}$.

Sem perda de generalidade, toma-se \mathbf{p} que satisfaz $\mathbf{p} = \mathbf{p}_0 + \mathbf{u}$, ou seja, $\mu = 1$. Derivando de acordo com (3.26) obtem-se

$$\dot{\mathbf{u}} = -\mathbf{v} - \boldsymbol{\omega} \times \mathbf{p} + \mathbf{v} + \boldsymbol{\omega} \times \mathbf{p}_0 = -\boldsymbol{\omega} \times (\mathbf{p} - \mathbf{p}_0). \quad (3.29)$$

Portanto,

$$\dot{\mathbf{u}} = \mathbf{u} \times \boldsymbol{\omega}. \quad (3.30)$$

Desenvolvendo (3.27) e derivando-a no tempo, tem-se

$$\dot{\mathbf{p}} \times \mathbf{u} - \dot{\mathbf{p}}_0 \times \mathbf{u} - \dot{\mathbf{u}} \times (\mathbf{p} - \mathbf{p}_0) = 0. \quad (3.31)$$

Substituindo (3.28) e (3.26) em (3.31) chega-se à equação

$$(-\mathbf{v} - \dot{\mathbf{p}}_0 - \boldsymbol{\omega} \times \mathbf{p}_0) \times \mathbf{u} + \mu(-\boldsymbol{\omega} \times \mathbf{u} - \dot{\mathbf{u}}) \times \mathbf{u} = 0. \quad (3.32)$$

Fazendo a substituição de $\dot{\mathbf{u}}$ em (3.32) pela relação dada em (3.30) chega-se à expressão

$$(-\mathbf{v} - \dot{\mathbf{p}}_0 - \boldsymbol{\omega} \times \mathbf{p}_0) \times \mathbf{u} = 0. \quad (3.33)$$

Pode-se re-escrever (3.33) como sendo

$$\dot{\mathbf{p}}_0 \times \mathbf{u} = \mathbf{u} \times \mathbf{v} + \mathbf{u} \times (\boldsymbol{\omega} \times \mathbf{p}_0). \quad (3.34)$$

Da relação encontrada em (2.13) obtem-se a seguinte expressão para $\dot{\mathbf{h}}$:

$$\dot{\mathbf{h}} = \dot{\mathbf{p}}_0 \times \mathbf{u} + \mathbf{p}_0 \times \dot{\mathbf{u}}, \quad (3.35)$$

que, fazendo uso dos resultados encontrados em (3.30) e (3.34), fica na forma

$$\dot{\mathbf{h}} = \mathbf{u} \times \mathbf{v} + \mathbf{u} \times (\boldsymbol{\omega} \times \mathbf{p}_0) + \mathbf{p}_0 \times (\mathbf{u} \times \boldsymbol{\omega}). \quad (3.36)$$

Utilizando a Identidade de Jacobi (B..9),

$$\mathbf{u} \times (\boldsymbol{\omega} \times \mathbf{p}_0) + \mathbf{p}_0 \times (\mathbf{u} \times \boldsymbol{\omega}) = -\boldsymbol{\omega} \times (\mathbf{p}_0 \times \mathbf{u}) = (\mathbf{p}_0 \times \mathbf{u}) \times \boldsymbol{\omega} = \mathbf{h} \times \boldsymbol{\omega}. \quad (3.37)$$

Substituindo (3.37) em (3.36) tem-se finalmente que

$$\dot{\mathbf{h}} = \mathbf{u} \times \mathbf{v} + \mathbf{h} \times \boldsymbol{\omega}. \quad (3.38)$$

Por fim, a relação obtida para a evolução dos parâmetros de \mathcal{L} é dado por

$$\dot{\lambda} = \begin{pmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{h}} \end{pmatrix} = \mathbf{L}_\lambda \boldsymbol{\nu}, \quad \mathbf{L}_\lambda = \begin{pmatrix} \mathbf{0}_{3 \times 3} & A_s(\mathbf{u}) \\ A_s(\mathbf{u}) & A_s(\mathbf{h}) \end{pmatrix}. \quad (3.39)$$

em que $\boldsymbol{\nu}$ é dado por (3.5).

Por fim, \mathbf{L}_s relacionando o vetor de características com a velocidade do robô $\dot{\mathbf{r}}$ é definido realizando a transformação dada por (3.8), resultando em

$$\mathbf{L}_s = \mathbf{L}_\lambda \mathbf{M}.$$

O Jacobiano da imagem obtido para \mathcal{L} (3.39) apresenta uma característica bastante interessante: a velocidade de translação não influencia o comportamento do parâmetro de orientação \mathbf{u} . Isso permite controlar o sistema utilizando leis de controle bastante simples.

3.4 ESTIMAÇÃO DAS COORDENADAS 3D DAS RETAS

A lei de controle definida por (3.18) depende explicitamente das coordenadas de Plücker λ referentes às retas que representam a porta. No entanto, tais parâmetros não são conhecidos *a priori*, sendo necessário estimá-los. Partindo-se apenas das informações fornecidas por uma única imagem, é impossível determinar as coordenadas de Plücker das retas verticais da porta. Além do mais, tais imagens apresentam ruídos e imprecisões, de forma que não se pode afirmar com exatidão os parâmetros que elas fornecem. Utilizar métodos estocásticos para estimar tais parâmetros permite incorporar à estimativa imprecisões de medição inerentes do processo de captura da imagem além de o sistema continuar operando por predição nos instantes em que as retas da porta saem momentaneamente do campo de visão das câmeras.

A influência da velocidade do robô sobre a evolução dos parâmetros λ pode ser bem modelada através da Matriz de Interação \mathbf{L}_λ dada por (3.39). Com isso, pode-se fazer uma predição da evolução das coordenadas 3D das retas à medida em que o sistema se move. Desde que se garantam as condições de observabilidade dos parâmetros de λ , sequências de imagens podem ser utilizadas como medidas de correção da predição de forma que, com o passar do tempo, chegue-se a uma estimativa razoável dos parâmetros.

Na abordagem aqui apresentada, as retas são consideradas descorrelacionadas uma das outras, podendo ser estimadas independentemente, cada uma com seu EKF [72]. Optou-se por não explorar

a rigidez da porta pois com isso tem-se um método mais abrangente, podendo ser estendido a situações de navegação que envolvam retas pertencentes a diferentes alvos, por exemplo. A separação do processo de estimação em diversos EKF, um para cada reta, também resulta em atualizações mais rápidas da estimativa, graças à inversão de matrizes menores, que são normalmente as tarefas mais demoradas dos filtros de Kalman [73]. Além do mais, no caso de uma das retas não possuir observações para o laço atual devido à oclusão temporária ou por ter saído fora do alcance da câmera, as demais retas ainda podem ser corrigidas usando suas respectivas observações.

O sistema formado pelos parâmetros de reta λ , velocidades da câmera ν e observações visuais y pode ser descrito no espaço de estados por

$$\begin{cases} \lambda_k = f(\lambda_{k-1}, \nu_{k-1}) + \mathbf{q}_k \\ \mathbf{y}_k = g(\lambda_k) + \mathbf{w}_k \end{cases} \quad (3.40)$$

com $\mathbf{q}_k \sim N(\mathbf{0}, \mathbf{Q}_k)$, $\mathbf{w}_k \sim N(\mathbf{0}, \mathbf{R}_k)$ descorrelacionados.

A partir de (3.39), $f(\lambda_{k-1}, \nu_{k-1})$ pode ser aproximada por

$$f(\lambda_{k-1}, \nu_{k-1}) \approx \lambda_{k-1} + T\mathbf{L}_\lambda \nu_{k-1}, \quad (3.41)$$

em que T é o período de amostragem e \mathbf{L}_λ é dado por (3.39).

A matriz de covariâncias \mathbf{R}_k é determinada com propagação de incertezas pela Transformada Unscented [51] a partir dos ruídos de medição de cada pixel da imagem. O vetor de observações $\mathbf{y}_k \in \mathbb{R}^{6 \times 1}$ corresponde às observações β e \mathbf{h} (cf. Seção 3.5) acrescidas das constantes 1 e 0, referentes às restrições de \mathbf{u} unitário e \mathbf{u} e \mathbf{h} perpendiculares:

$$\mathbf{y}_k = \begin{pmatrix} \beta_k \\ \underline{\mathbf{h}}_k \|\mathbf{h}_k\| \\ 1 \\ 0 \end{pmatrix}. \quad (3.42)$$

Assim sendo, tem-se a função $g(\lambda_k)$ dada por

$$g(\lambda_k) = \begin{pmatrix} \arctan(u_{y_k}/u_{x_k}) \\ \mathbf{h}_k \\ \|\mathbf{u}_k\|^2 \\ \mathbf{h}_k^T \mathbf{u}_k \end{pmatrix}.$$

Percebe-se que devido a $f(\lambda_{k-1}, \nu_{k-1})$ e $g(\lambda_k)$, o sistema dado em 3.40 é não-linear. Portanto, para realizar a estimação dos parâmetros das retas, implementou-se um filtro de Kalman estendido [72, 73], que é o resultado de um processo de linearização do sistema por aproximação segundo uma expansão em primeira ordem por série de Taylor. Esse filtro estocástico é dividido em duas etapas: predição e correção.

Na etapa de predição, dada a estimativa inicial $\hat{\lambda}_0$ e correspondente matriz de covariâncias \mathbf{P}_0 para os parâmetros da reta, deve-se determinar a evolução da estimativa de λ_k dados $\hat{\lambda}_{k-1}$ e ν_{k-1} .

Tal evolução é dada por

$$\hat{\boldsymbol{\lambda}}_{k|k-1} = f(\hat{\boldsymbol{\lambda}}_{k-1}, \boldsymbol{\nu}_{k-1}). \quad (3.44)$$

A matriz de covariâncias do erro de predição é dada por

$$\mathbf{P}_{k|k-1} = \mathbf{P}_{k-1} + T^2 \mathbf{L}_\lambda \mathbf{S}_k \mathbf{L}_\lambda^T + \mathbf{Q}_k, \quad (3.45)$$

com \mathbf{Q}_k referente à incerteza de predição do sistema e \mathbf{S}_k referente à incerteza sobre $\boldsymbol{\nu}_k$, proveniente da odometria.

A etapa de correção ocorre sempre que houver novas observações dadas por um vetor de medidas \mathbf{y}_k . A estimativa corrigida de $\boldsymbol{\lambda}$ é obtida utilizando-se a seguinte equação

$$\hat{\boldsymbol{\lambda}}_k = \hat{\boldsymbol{\lambda}}_{k|k-1} + \mathbf{G}_k \left(\mathbf{y}_k - g(\hat{\boldsymbol{\lambda}}_{k|k-1}) \right), \quad (3.46)$$

com \mathbf{G}_k conhecido como ganho de Kalman e dado por

$$\mathbf{G}_k = \mathbf{P}_{k|k-1} \left(\frac{\partial g(\hat{\boldsymbol{\lambda}}_{k-1})}{\partial \hat{\boldsymbol{\lambda}}_{k-1}} \right)^T \left(\frac{\partial g(\hat{\boldsymbol{\lambda}}_{k-1})}{\partial \hat{\boldsymbol{\lambda}}_{k-1}} \mathbf{P}_{k|k-1} \left(\frac{\partial g(\hat{\boldsymbol{\lambda}}_{k-1})}{\partial \hat{\boldsymbol{\lambda}}_{k-1}} \right)^T + \mathbf{R}_k \right)^{-1}.$$

A matriz de covariâncias após a correção é calculada por

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{G}_k \frac{\partial g(\hat{\boldsymbol{\lambda}}_{k-1})}{\partial \hat{\boldsymbol{\lambda}}_{k-1}} \right) \mathbf{P}_{k|k-1} \left(\mathbf{I} - \mathbf{G}_k \frac{\partial g(\hat{\boldsymbol{\lambda}}_{k-1})}{\partial \hat{\boldsymbol{\lambda}}_{k-1}} \right)^T + \mathbf{G}_k \mathbf{R}_k \mathbf{G}_k^T. \quad (3.47)$$

Quando não há nova medida, a etapa de correção não ocorre e a estimativa é dada apenas pelo valor obtido na predição.

3.5 OBTENÇÃO DAS OBSERVAÇÕES VISUAIS

A fim de utilizar o sistema de câmeras como sensores, é importante determinar, dentre as informações fornecidas pela imagem, quais são capazes de contribuir para a estimação do vetor de parâmetros $\boldsymbol{\lambda}$ que descreve \mathcal{L} no espaço. Primeiramente, como mencionado no Capítulo 2, a reta ℓ projetada na imagem da câmera esquerda fornece de forma trivial a componente $\underline{\mathbf{h}}$ das coordenadas de Plücker da reta \mathcal{L} no espaço 3D [48]. Isso pode ser facilmente verificado nos passos seguintes.

Considere \mathcal{P} um ponto qualquer de \mathcal{L} representado no sistema de coordenadas \mathcal{O}^c . Suas coordenadas no espaço são dadas por $\mathbf{p} = (x, y, z)^T$ e as coordenadas homogêneas de sua projeção no sistema \mathcal{O}^i são $\tilde{\mathbf{p}}^i = (x^i, y^i, 1)^T$. De (2.26) chega-se à relação

$$\mathbf{p}_n = (x_n, y_n, 1)^T = \frac{1}{z} \mathbf{p} = \mathbf{K}^{-1} \tilde{\mathbf{p}}^i. \quad (3.48)$$

A equação da reta ℓ é da forma $ax_n + by_n + c = 0$, com as restrições $a > 0$ e $a^2 + b^2 + c^2 = 1$. Sendo $\mathbf{h} = \|\mathbf{h}\| \underline{\mathbf{h}}$ e utilizando (3.48), pode-se re-escrever $\mathbf{h}^T \mathbf{p} = 0$ como $z \|\mathbf{h}\| \underline{\mathbf{h}}^T \mathbf{p}_n = 0$. Sabe-se que $\|\mathbf{h}\|$

e z são diferentes de zero, logo tem-se que $\underline{\mathbf{h}}^T \mathbf{p}_n = 0$ que corresponde exatamente à equação da reta ℓ . Assim sendo, o vetor $\underline{\mathbf{h}}$ é formado pelos coeficientes a, b e c da equação de ℓ

$$\underline{\mathbf{h}} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}. \quad (3.49)$$

Apesar de ser possível calcular $\underline{\mathbf{h}}$ a partir de apenas dois pontos, ruídos de medição tornam essa estratégia bastante susceptível à grande variação destes pontos. Alternativamente, preferiu-se uma abordagem redundante utilizando um sistema sobredeterminado com todos¹ os *pixels* associados à reta. A partir do conjunto de n pontos pertencentes à reta projetada ℓ , é possível determinar a, b e c pela solução de uma equação linear do tipo $\mathbf{A}\mathbf{x} = \mathbf{0}$, em que $\mathbf{x} = (a, b, c)^T$, $\mathbf{0} \in \mathbb{R}^{n \times 1}$ e $\mathbf{A} \in \mathbb{R}^{n \times 3}$ é dado por

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{pmatrix} \quad (3.50)$$

Por tratar-se de um sistema sobredeterminado em que há mais equações do que incógnitas deseja-se obter a melhor estimativa de \mathbf{x} . Em outras palavras, procura-se determinar o vetor \mathbf{x} para o qual $\|\mathbf{A}\mathbf{x}\|$ tenha o menor valor sujeito à restrição $\|\mathbf{x}\| = 1$. Tal solução é conhecida como a estimativa de mínimos-quadrados para o sistema sobredeterminado e pode ser encontrada através da decomposição em valores singulares da matriz \mathbf{A}

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \quad (3.51)$$

em que $\mathbf{U} \in \mathbb{R}^{n \times 3}$ é ortogonal de norma unitária, $\mathbf{D} \in \mathbb{R}^{3 \times 3}$ é diagonal e $\mathbf{V} \in \mathbb{R}^{3 \times 3}$ é ortogonal de norma unitária. A solução de mínimos-quadrados para $\mathbf{A}\mathbf{x} = \mathbf{0}$ é dada pela última coluna da matriz \mathbf{V} (cf. Anexo C.). Obtido o vetor normalizado $\underline{\mathbf{h}}$, ainda se faz necessário extrair $\|\underline{\mathbf{h}}\|$ para recuperar \mathbf{h} completamente. No entanto, a informação de profundidade se perde na projeção e portanto, não pode ser recuperada a partir de uma única imagem. O problema pode ser resolvido utilizando-se o par de imagens adquiridas pelas duas câmeras do sistema binocular. Conhecendo-se a transformação entre os sistemas \mathcal{O}^c e \mathcal{O}^d , é possível determinar $\|\underline{\mathbf{h}}\|$. Para isso, utiliza-se o método descrito a seguir [45].

Sendo \mathcal{L} representada por $\boldsymbol{\lambda} = (\mathbf{u}, \mathbf{h})^T$, e escolhendo \mathbf{p} o ponto de \mathcal{L} mais próximo da origem do sistema de coordenadas \mathcal{O}^c , temos que este mesmo ponto pode ser representado na câmera direita por um \mathbf{p}^d dado por

$$\mathbf{p}^d = \mathbf{R}_c^d \mathbf{p} + \mathbf{t}_c^d. \quad (3.52)$$

De (2.13) e (3.10) tem-se

$$\mathbf{h}^d = \mathbf{R}_c^d \left(\mathbf{h} + (\mathbf{R}_c^d)^{-1} \mathbf{t}_c^d \times \mathbf{u} \right). \quad (3.53)$$

¹Na verdade, o uso de todos os *pixels* é extremamente pesado computacionalmente já que cada reta pode ter mais de 100 pontos associados em cada imagem do sistema binocular. Assim sendo, uma estratégia de *bootstrap* é utilizada, em que escolhe-se aleatoriamente um subconjunto de pontos que seja representativo da reta em questão. O tamanho ideal deste subconjunto é determinado segundo um compromisso entre desempenho computacional e o grau de incerteza nas coordenadas dos pontos. Para o algoritmo de tracking e extração de retas utilizado em nossa aplicação, verificou-se experimentalmente que 20 pontos em cada imagem do par binocular são suficientes para uma estimativa robusta dos parâmetros da reta projetada na imagem.

Como \mathbf{p} é o ponto \mathcal{L} mais próximo à origem, o ângulo entre \mathbf{p} e \mathbf{u} é de 90° . Assim sendo, de (2.13) e de (2.17), chega-se a

$$\begin{aligned}\|\mathbf{h}\| &= \|\mathbf{p} \times \mathbf{u}\| = \|\mathbf{p}\| \|\mathbf{u}\| \sin 90^\circ, \\ &= \|\mathbf{p}\| \|\mathbf{u}\|.\end{aligned}\quad (3.54)$$

Multiplicando ambos os lados de (3.53) por $(\mathbf{R}_c^d)^{-1}$ à esquerda:

$$(\mathbf{R}_c^d)^{-1}\mathbf{h}^d = \mathbf{h} + (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}, \quad (3.55)$$

e então:

$$\mathbf{h} = (\mathbf{R}_c^d)^{-1}\mathbf{h}^d - (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}. \quad (3.56)$$

De (3.55):

$$\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d] = \mathbf{h} \times [\mathbf{h} + (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}] = \mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}]. \quad (3.57)$$

Usando a fórmula de Lagrange (B..8):

$$\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d] = \cancel{[\mathbf{h}^T \mathbf{u}]^0} (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d - \mathbf{h}^T [(\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d] \mathbf{u}, \quad (3.58)$$

$$\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d] = -\mathbf{h}^T [(\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d] \mathbf{u}. \quad (3.59)$$

Do produto interno entre (3.56) e $(\mathbf{R}_c^d)^{-1}\mathbf{h}^d \mathbf{u}$ temos:

$$\begin{aligned}\mathbf{h}^T [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d \mathbf{u}] &= [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d - (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}]^T [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d \mathbf{u}], \\ &= [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d]^T [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d \mathbf{u}] - [(\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \times \mathbf{u}]^T [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d \mathbf{u}], \\ &= (\mathbf{h}^d)^T (\mathbf{R}_c^d)^{-1}\mathbf{t}_c^d \mathbf{u}, \\ &= (\mathbf{h}^d)^T \mathbf{t}_c^d \mathbf{u}.\end{aligned}\quad (3.60)$$

Substituindo (3.60) em (3.59):

$$\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d] = -(\mathbf{h}^d)^T \mathbf{t}_c^d \mathbf{u}, \quad (3.61)$$

$$(\mathbf{h}^d)^T \mathbf{t}_c^d = -(\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d]) \mathbf{u}^{-1}, \quad (3.62)$$

$$\|(\mathbf{h}^d)^T \mathbf{t}_c^d\| = \|\mathbf{u}\|^{-1} \|\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d]\|. \quad (3.63)$$

Usando o resultado de (3.54) para substituir $\|\mathbf{u}\|^{-1}$ em (3.63):

$$\|(\mathbf{h}^d)^T \mathbf{t}_c^d\| = \|\mathbf{p}\| \|\mathbf{h}\|^{-1} \|\mathbf{h} \times [(\mathbf{R}_c^d)^{-1}\mathbf{h}^d]\|. \quad (3.64)$$

Dividindo ambos lados por $\|\mathbf{h}^d\|$:

$$\|(\underline{\mathbf{h}}^d)^T \underline{\mathbf{t}}_c^d\| = \|\mathbf{p}\| \|\underline{\mathbf{h}} \times [(\mathbf{R}_c^d)^{-1}\underline{\mathbf{h}}^d]\|. \quad (3.65)$$

Sabendo que $\|\mathbf{u}\| = 1$, de (3.54) tem-se que $\|\mathbf{p}\| = \|\mathbf{h}\|$. Assim sendo, de (3.65) tem-se finalmente que

$$\|\mathbf{h}\| = \left\| \underline{\mathbf{h}} \times [(\mathbf{R}_c^d)^{-1} \underline{\mathbf{h}}^d] \right\|^{-1} \left\| (\underline{\mathbf{h}}^d)^T \mathbf{t}_c^d \right\|. \quad (3.66)$$

Com isso, a partir de um par de imagens do sistema binocular, tem-se:

$$\mathbf{h} = \underline{\mathbf{h}} \|\mathbf{h}\|, \quad (3.67)$$

com $\underline{\mathbf{h}}$ obtido de (3.50). e $\|\mathbf{h}\|$ dado por (3.66).

Nota-se que a medida $\|\mathbf{h}\|$ é referente à profundidade da reta e calculada a partir de um par de imagens. Tal informação é muito útil para o caso de estimação e controle realizados simultaneamente pois, sem o uso direto da medida de profundidade, tem-se baixa observabilidade de λ tornando a convergência da estimativa lenta demais para ser utilizada no cálculo da ação de controle [46].

Caso as observações de uma reta apresentem uma variação muito pequena no parâmetro $\underline{\mathbf{h}}$, não é possível recuperar a informação de profundidade à partir de (3.66) e conseqüentemente, a estimativa de \mathbf{h} não pode ser corrigida de forma adequada. Assim sendo, para um sistema de visão monocular, o par de imagens precisa ser obtido à partir de duas medidas realizadas em instantes diferentes e de posições suficientemente distantes de forma a provocar variação substancial dos parâmetros $\underline{\mathbf{h}}$ extraídos. Com isso, tem-se que a capacidade do par de imagens fornecer $\|\mathbf{h}\|$ num sistema monocular depende do deslocamento realizado pelo robô entre as duas medições.

Já para o caso binocular, o par de imagens é adquirido simultaneamente à partir de posições distintas e à uma distância fixa definida pela *baseline* do arranjo de câmeras. Em problemas como o proposto neste trabalho, em que a tarefa de navegação é executada em ambientes internos, tem-se que os objetos da cena encontram-se no máximo há algumas dezenas de metros do robô. Desta forma, um sistema binocular com *baseline* da ordem de $10^{-1} m$ já é suficiente para fornecer boas medidas de $\|\mathbf{h}\|$.

Assim sendo, enquanto a extração de profundidade é garantida para o caso binocular independentemente do deslocamento realizado pelo robô, para um sistema de visão monocular é necessário manipular a trajetória para se garantir um par de imagens capaz de fornecer $\|\mathbf{h}\|$ com alguma precisão. Isso representa uma grande vantagem da abordagem binocular sobre a monocular pois tem-se convergência rápida e garantida das estimativas dos parâmetros da reta, além de não ser necessário deslocar o robô arbitrariamente, muitas vezes operando em malha aberta. Tais fatores compensam o custo de se instalar uma câmera extra de forma a se ter uma configuração binocular, abordagem adotada para este trabalho.

A componente \mathbf{u} das coordenadas de Plücker poderia ser determinada a partir de um par de imagens. No entanto, a extração de tal informação exigiria cálculos complexos e tão ou mais penosos computacionalmente que a determinação de \mathbf{h} . Contudo, a inclinação de ℓ fornece de forma simples e direta a relação entre as coordenadas de \mathbf{u} nos eixos X^c e Y^c do sistema \mathcal{O}^c da câmera esquerda. O ângulo formado entre ℓ e o eixo positivo x é igual a

$$\beta = \arctan \left(\frac{u_y}{u_x} \right) = \arctan \left(\frac{-a}{b} \right). \quad (3.68)$$

A combinação da relação (3.68) com as restrições entre \mathbf{u} e \mathbf{h} definidas em (2.14) é explorada pelo FKE na estimação seqüencial de \mathbf{u} , tornando possível corrigi-lo sem que este parâmetro seja extraído explicitamente de cada par de imagens.

Assim sendo, para cada reta a ser estimada, tem-se duas observações visuais associadas: \mathbf{h} e β . Devido à própria natureza do processo de captura de imagens, existe uma incerteza associada às coordenadas em pixels dos pontos extraídos das imagens. O uso destes pontos para determinar \mathbf{h} e β propaga o erro para a solução obtida. A incerteza associada a cada observação visual é estimada utilizando o método de propagação de incertezas através da Transformada *Unscented* [51].

3.6 EXTRAÇÃO E RASTREAMENTO DAS RETAS NA IMAGEM

O processo descrito na Seção 3.5 calcula as observações visuais das retas à partir das coordenadas dos pontos que as constituem. Entretanto, é necessário que se tenha identificado as retas na imagem e extraído as coordenadas de seus pontos em um processo anterior. Para tanto, utiliza-se um procedimento de duas etapas proposto e desenvolvido por G. A. Borges e descrito brevemente à seguir.

A primeira etapa corresponde à extração das retas presentes nas imagens das câmeras direta e esquerda (*cf.* Figura 3.3(a)). Para cada imagem, aplica-se um detector de bordas de Canny, seguido de um processo de clusterização do tipo *split-and-merge*. Ao final desta etapa, tem-se todos os segmentos de reta presentes em cada uma das imagens.

A segunda etapa, chamada de rastreamento, é relativamente simples e é a responsável por associar uma reta extraída da imagem atual à uma das retas rastreadas na imagem anterior (*cf.* 3.3(b)). Para a primeira imagem, o usuário seleciona quais retas serão rastreadas para a aplicação em questão. O rastreamento é realizado de forma independente para cada reta e para cada câmera, de forma que uma dada reta na câmera esquerda é rastreada na seqüência de imagens provida por aquela câmera somente. Assim sendo, todas as retas extraídas na primeira etapa que estejam dentro de uma região de

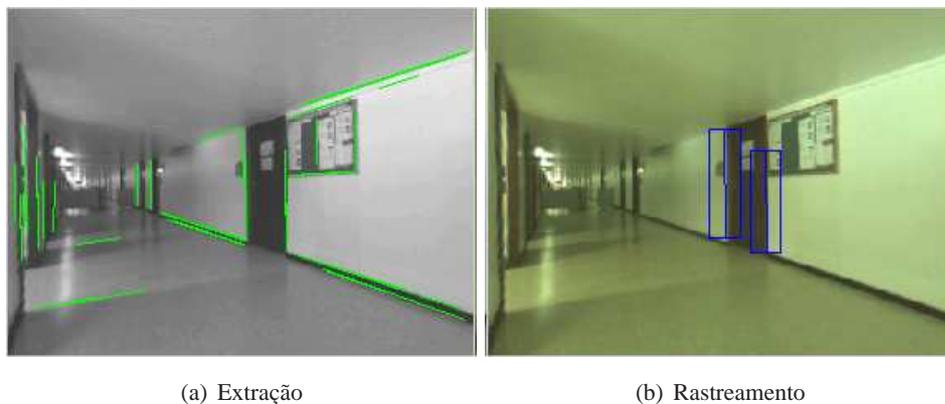


Figura 3.3: Retas extraídas de imagem da câmera esquerda e resultado correspondente na etapa de rastreamento. (a) As retas em verde representam todas as retas extraídas da imagem. (b) Os retângulos azuis representam as regiões de busca e as duas retas azuis são o resultado do rastreamento.

busca são ditas como retas candidatas. O procedimento de associação calcula medidas de similaridade entre a reta rastreada na imagem anterior e todas as suas candidatas. Tais medidas de similaridade são conferidas pela distância entre segmentos de reta, distância de orientação e distância lateral. A linha candidata que apresentar distância entre segmentos menor que 15, distância de orientação menor que 15° e tiver a menor distância lateral é associada à reta rastreada e as coordenadas de seus *pixels* são armazenadas e utilizadas na obtenção das observações visuais conforme descrito na Seção 3.5.

4 RESULTADOS EXPERIMENTAIS

A theory must be tempered with reality.

Jawaharlal Nehru (1889 - 1964)

4.1 INTRODUÇÃO

O método desenvolvido neste trabalho foi avaliado em duas fases. Inicialmente, utilizou-se uma simulação computacional desenvolvida em *Matlab*TM para verificar o desempenho do estimador FKE bem como dos controladores servo-visuais *IBVS* e *PBVS* propostos. Em um segundo passo, avaliou-se a solução proposta sobre a plataforma Omni em situações reais de *docking*. Os resultados obtidos, combinados com as simulações computacionais, comprovam a viabilidade do controle servo-visual com estimação ânea para resolução de problemas de docking com retas.

4.2 SIMULAÇÃO

Uma simulação do robô Omni e de seu sistema de câmeras foi construída em *Matlab*TM para realização de experimentos antes da implementação sobre a plataforma real. Tal simulação apresenta diferentes configurações possíveis de execução, permitindo analisar as etapas de estimação e controle separadamente ou em conjunto, alternar entre controle baseado em imagem ou posição, determinar o número de câmeras utilizadas, alterar os parâmetros dos controladores, inicialização do estimador, posição inicial do sistema, posição alvo para o *docking* ou até mesmo a largura e posição da porta. Com isso, alguns experimentos foram realizados, buscando identificar as principais características da abordagem desenvolvida e sintonizar os parâmetros de forma segura. As configurações de teste utilizadas estão especificadas em detalhes no Anexo D..

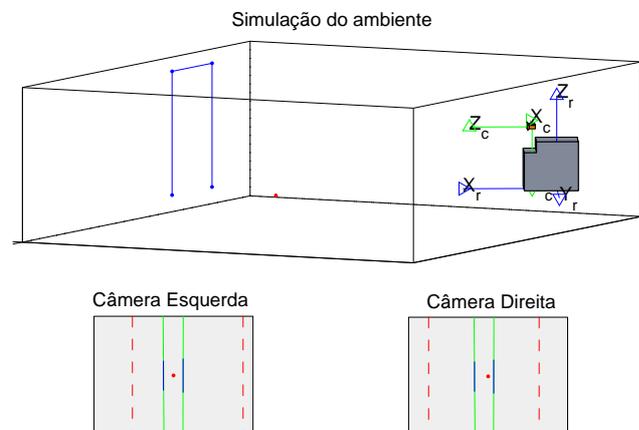


Figura 4.1: Ambiente de simulação em Matlab.

A Figura 4.1 traz a tela do simulador, onde a parte superior corresponde a uma representação tridimensional das portas, robô e sistema de câmeras, enquanto a área inferior é constituída das duas imagens simuladas para as câmeras. As retas azuis correspondem à imagem da porta no instante atual, as retas verdes são a projeção da última estimativa das laterais da porta e as retas tracejadas em vermelho correspondem à projeção da porta na imagem quando o robô estiver na posição final desejada.

4.2.1 Estimação dos parâmetros da porta

Os primeiros experimentos realizados em simulação nos permitem avaliar o desempenho do estimador estocástico do tipo FKE proposto e implementado segundo a Seção 3.4. Nas simulações realizadas, a plataforma é controlada por joystick e move-se em direção à porta em uma trajetória suave com o sistema de odometria fornecendo medidas a cada 5 ms enquanto o sistema de câmeras embarcado na frente do robô captura imagens de $320 \times 240\text{ pixels}$ a uma taxa de 10 frames por segundo. Tais medições são utilizadas nas etapas de correção e predição do EKF, respectivamente, o qual é executado a cada 100 ms .

Primeiramente, comparou-se o desempenho do estimador proposto em duas configurações de câmera: binocular e monocular. Enquanto na primeira tem-se um par de imagens obtidas simultaneamente por duas câmeras em posições diferentes, na segunda abordagem o par é formado por duas imagens obtidas em instantes diferentes pela mesma câmera. Conforme discutido na Seção 3.5, para o caso de uma aplicação *indoor*, a configuração monocular apresenta o inconveniente de ser necessário mover o robô apenas por predição, muitas vezes sem uma estimativa inicial adequada da posição da porta, até que finalmente se obtenha duas imagens distintas o suficiente para realizar a primeira correção das estimativas. Com a abordagem monocular, também há a necessidade de forçar trajetórias que resultem em seqüências de imagens razoavelmente diferentes da anterior, sob risco de não ser possível extrair satisfatoriamente a informação de profundidade.

Para avaliar ambas estratégias, partiu-se de uma estimativa inicial $\hat{\mathbf{u}}(0) = (0 \ -1 \ 0)^T$ e $\hat{\mathbf{h}}(0) = (3 \ 0 \ 0)^T$ para ambas as retas, com desvio padrão de $0,5\text{ pixel}$ para as coordenadas dos pontos na imagem e desvio padrão de $0,05\text{ m/s}$ e $1^\circ/\text{s}$ para as medidas de odometria. Simulou-se uma situação em que o robô move-se em trajetória retilínea em direção à porta e está posicionado de tal forma que uma das retas passa exatamente sobre o eixo óptico da câmera esquerda (no caso monocular, a única câmera presente) durante toda a simulação. A Figura 4.2 apresenta as imagens construídas pelo simulador para os instantes inicial e final do experimento, onde nota-se que a projeção da lateral esquerda da porta não muda de posição quando vista pela câmera esquerda.

Essa situação é crítica no sistema monocular pois não excita o EKF de forma satisfatória, tornando impossível recuperar a informação de profundidade para o parâmetro \mathbf{h} de uma das retas. A Figura 4.3 apresenta o erro de estimação para cada lateral da porta ao longo do experimento monocular. Observa-se que a estimação para a reta esquerda (Figura 4.3(a)) diverge devido a uma excitação pobre dos parâmetros, enquanto a reta direita (Figura 4.3(b)) não diverge, mas converge muito lentamente devido à necessidade de se obter pares de imagens suficientemente distantes para extrair a profundidade.

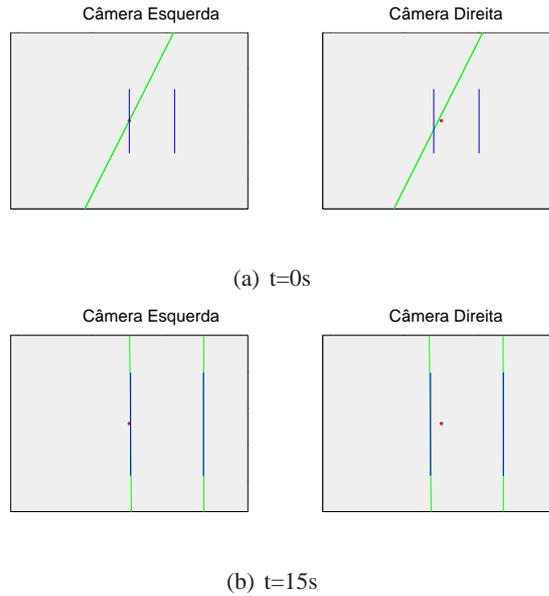
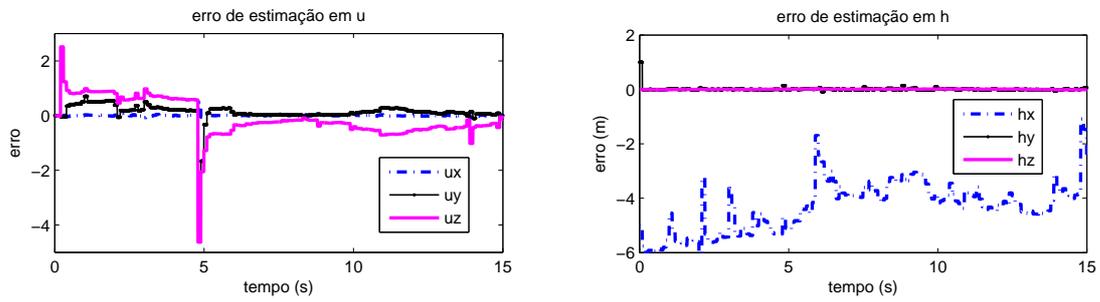


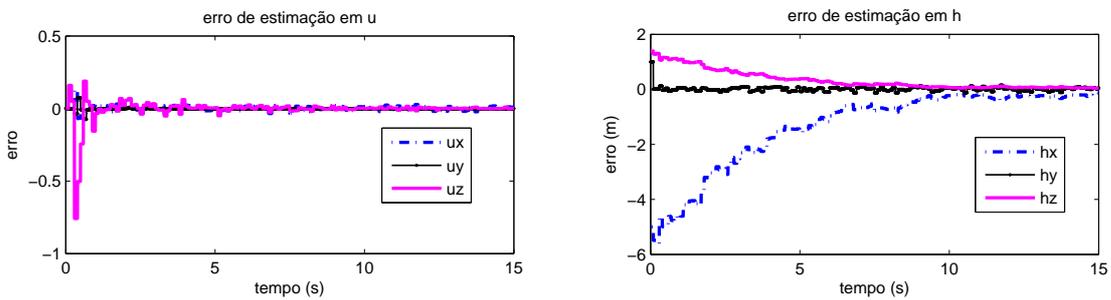
Figura 4.2: Imagens simuladas para as câmeras durante trajetória retilínea com uma das retas posicionada sempre ao centro da imagem da câmera esquerda. As linhas azuis correspondem às projeções das laterais da porta enquanto as linhas verdes são projeções da estimação.

No método binocular esse tipo de problema é completamente contornado já que, para objetos em ambientes fechados, o par de câmeras sempre fornece dois \underline{h} razoavelmente diferentes independentemente da trajetória seguida pelo robô. Com isso, a informação de profundidade é recuperada, possibilitando a correção das estimativas de \underline{h} . A Figura 4.4 apresenta o erro de estimação de cada lateral da porta ao longo do experimento binocular. Observa-se que as estimativas tanto de \underline{u} quanto de \underline{h} convergem rapidamente para as duas retas devido à disponibilidade de informação de profundidade logo no início do experimento.

Por fim, verificou-se a robustez do estimador em configuração binocular através de testes em que variou-se a inicialização dos estados do filtro, o desvio padrão de medição e o desvio padrão de odometria. Observa-se que quanto piores as estimativas iniciais ou quanto maiores as incertezas utilizadas, maior é o tempo necessário para se atingir a convergência. Entretanto, em todas as combinações testadas, o estimador sempre atinge a convergência após um período de tempo suficiente. As Tabelas 4.1, 4.2 e 4.3 apresentam a média da norma do erro de estimação para os parâmetros \underline{u} e \underline{h} em 50 execuções do processo de estimação. A Tabela 4.1 mostra os resultados obtidos após 15 segundos de simulação, em que os desvios padrões foram os mesmos nas três situações testadas e variou-se apenas a estimativa inicial $\hat{\underline{u}}(0)$ e $\hat{\underline{h}}(0)$. Nota-se que mesmo utilizando estimativas iniciais muito erradas como no caso da terceira linha, após um período de tempo suficiente, os parâmetros 3D de ambas retas chegaram a valores pequenos de erro, indicando uma situação de convergência. A Tabela 4.2 mostra os resultados obtidos após 15 segundos de simulação, em que variou-se apenas o desvio padrão de medição, mantendo-se todo o resto constante. É possível observar que, mesmo com grande incerteza nas coordenadas dos *pixels* da imagem, os parâmetros das duas retas convergiram com o tempo. De forma similar, a Tabela 4.3 mostra os resultados após 15 segundos de simulação para o caso em que varia-se apenas o desvio padrão de odometria. Assim como nos testes anteriores, após tempo suficiente, as estimativas convergem apesar dos erros nas medidas de velocidade do robô.

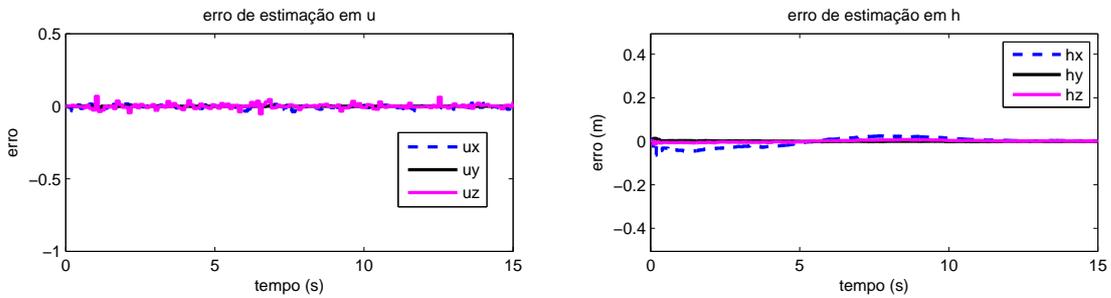


(a) Lateral esquerda da porta

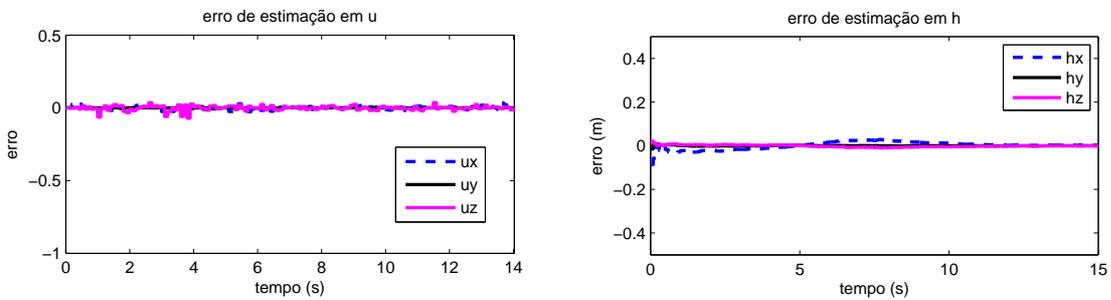


(b) Lateral direita da porta

Figura 4.3: Erros de estimação das laterais esquerda (a) e direita (b) da porta em simulação do método monocular.



(a) Lateral esquerda da porta



(b) Lateral direita da porta

Figura 4.4: Erros de estimação das laterais esquerda (a) e direita (b) da porta em simulação do método binocular.

Tabela 4.1: Estados Iniciais

$\hat{\mathbf{u}}(\mathbf{0})$ e $\hat{\mathbf{h}}(\mathbf{0})$	$\ u_1 - \hat{u}_1\ $	$\ h_1 - \hat{h}_1\ $	$\ u_2 - \hat{u}_2\ $	$\ h_2 - \hat{h}_2\ $
$\begin{pmatrix} 0 & -1 & 0 \end{pmatrix}^T$	0,0005	0,0027 m	0,0005	0,0024 m
$\begin{pmatrix} 0,45 & -0,89 & 0 \end{pmatrix}^T$	0,0004	0,0027 m	0,0005	0,0025 m
$\begin{pmatrix} 0,58 & 0,58 & 0,58 \end{pmatrix}^T$	0,0005	0,0023 m	0,0005	0,0026 m

Tabela 4.2: Desvio padrão de medição

\mathbf{r}_k	$\ u_1 - \hat{u}_1\ $	$\ h_1 - \hat{h}_1\ $	$\ u_2 - \hat{u}_2\ $	$\ h_2 - \hat{h}_2\ $
0,5 pixel	0,0005	0,0031 m	0,0005	0,0027 m
1 pixel	0,0011	0,0055 m	0,0015	0,0057 m
5 pixels	0,0022	0,0063 m	0,0025	0,0074 m

Tabela 4.3: Desvio padrão de odometria

\mathbf{s}_k	$\ u_1 - \hat{u}_1\ $	$\ h_1 - \hat{h}_1\ $	$\ u_2 - \hat{u}_2\ $	$\ h_2 - \hat{h}_2\ $
0,05 m/s, 1 grau/s	0,0005	0,0025 m	0,0004	0,0023 m
0,1 m/s, 2 grau/s	0,0005	0,0029 m	0,0005	0,0034 m
0,25 m/s, 5 grau/s	0,0005	0,0044 m	0,0004	0,0041 m

4.2.2 Controle servo-visual

Supondo conhecimento prévio dos parâmetros 3D da porta, foram realizados testes em simulação para avaliar o desempenho dos controladores servo-visuais propostos na Seção 3.3. O objetivo destes ensaios é verificar as estratégias de controle servo-visual propostas de forma independente do processo de estimação. Assim sendo, o cálculo dos sinais de controle não usa uma estimação $\hat{\lambda}$, mas os valores reais de λ em cada instante de tempo. Tanto na abordagem *IBVS* quanto na *PBVS* variou-se a posição inicial do robô para verificar o desempenho dos controladores e a qualidade das trajetórias geradas.

O controlador *PBVS* foi avaliado com pólos em $p = -1$ e fazendo o vetor $\mathbf{b} = \mathbf{0}_{3 \times 12}$, em posições iniciais distintas, como é mostrado na Figura 4.5, em que tem-se uma vista superior do ambiente de trabalho. Os triângulos representam a pose do robô num dado instante de tempo, os dois pontos azuis representam as laterais da porta e o ponto vermelho corresponde à posição desejada para o centro do robô ao fim da ação de controle. Analisando as trajetórias geradas em cada experimento, percebe-se que elas foram suaves e razoavelmente diretas em todas as situações testadas e os sinais de controle gerados mantiveram-se em valores aceitáveis e suavemente decrescentes.

Nos ensaios com o controlador *IBVS*, primeiramente variou-se o tipo de aproximação escolhido para a matriz $\hat{\mathbf{L}}_s^\dagger$ de acordo com as possibilidades discutidas na Seção 2.1.3, de forma a avaliar o efeito de tal escolha no desempenho do controlador. É de se esperar que a escolha de $\hat{\mathbf{L}}_s^\dagger = \mathbf{L}_s^\dagger$ resulte em trajetórias quase retilíneas no espaço da imagem, mas no entanto nada é garantido quanto ao movimento do robô no plano cartesiano [74]. De fato, nota-se a partir da Figura 4.6(a) que essa aproximação

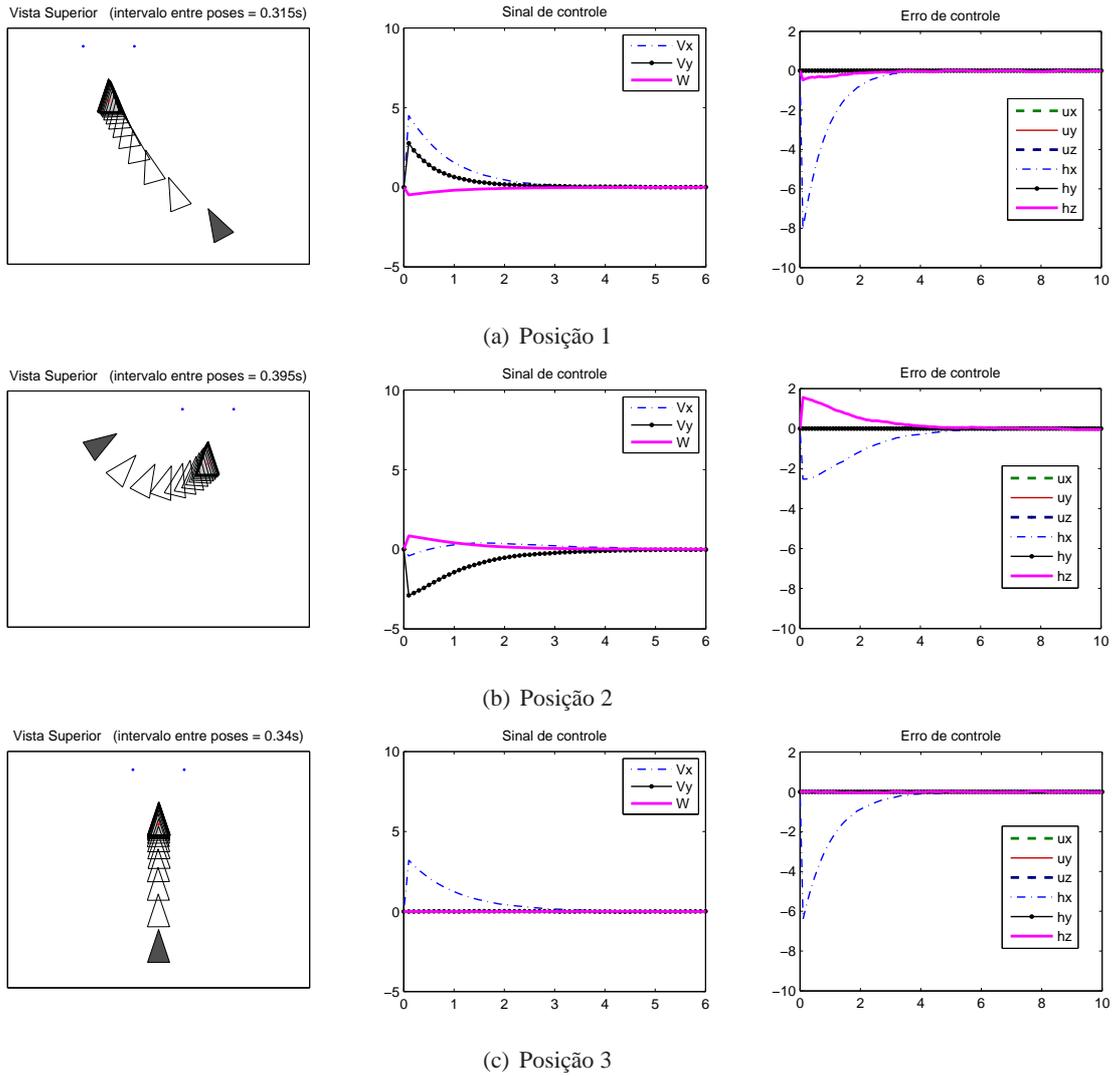


Figura 4.5: Trajetórias e sinais de controle gerados pelo controlador *PBVS* partindo respectivamente das posições 1, 2 e 3 descritas na Figura D..1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.

para $\hat{\mathbf{L}}_s^\dagger$ não necessariamente resulta em uma trajetória retilínea no espaço de trabalho do robô, além de provocar sinais de controle muito elevados no início da atuação e ter uma convergência bastante lenta quando se aproxima da posição final desejada, o que pode ser observado devido ao grande acúmulo de triângulos na vizinhança da posição alvo do *docking*. Esse comportamento foi justificado em [61], em que o problema de controle é analisado como um problema de minimização não-linear por mínimos-quadrados e mostra-se que o método de controle pela pseudo-inversa da Jacobiana é equivalente a método de Minimização de Gauss-Newton (GNM), de convergência linear para o caso de sistemas não-lineares. Uma alternativa seria adotar pólos em malha fechada mais rápidos com o intuito de acelerar a convergência. No entanto tal medida causaria sinais de controle iniciais ainda maiores e portanto esta opção não é viável.

A escolha de $\hat{\mathbf{L}}_s^\dagger = \mathbf{L}_{s^*}^\dagger$ faz com que a lei de controle comporte-se como se o erro na imagem fosse tão pequeno quanto possível [57]. Com isso, os sinais de controle mostram-se bem mais razoáveis, como pode-se perceber na Figura 4.6(b). Entretanto, assim como a primeira escolha analisada, fazer

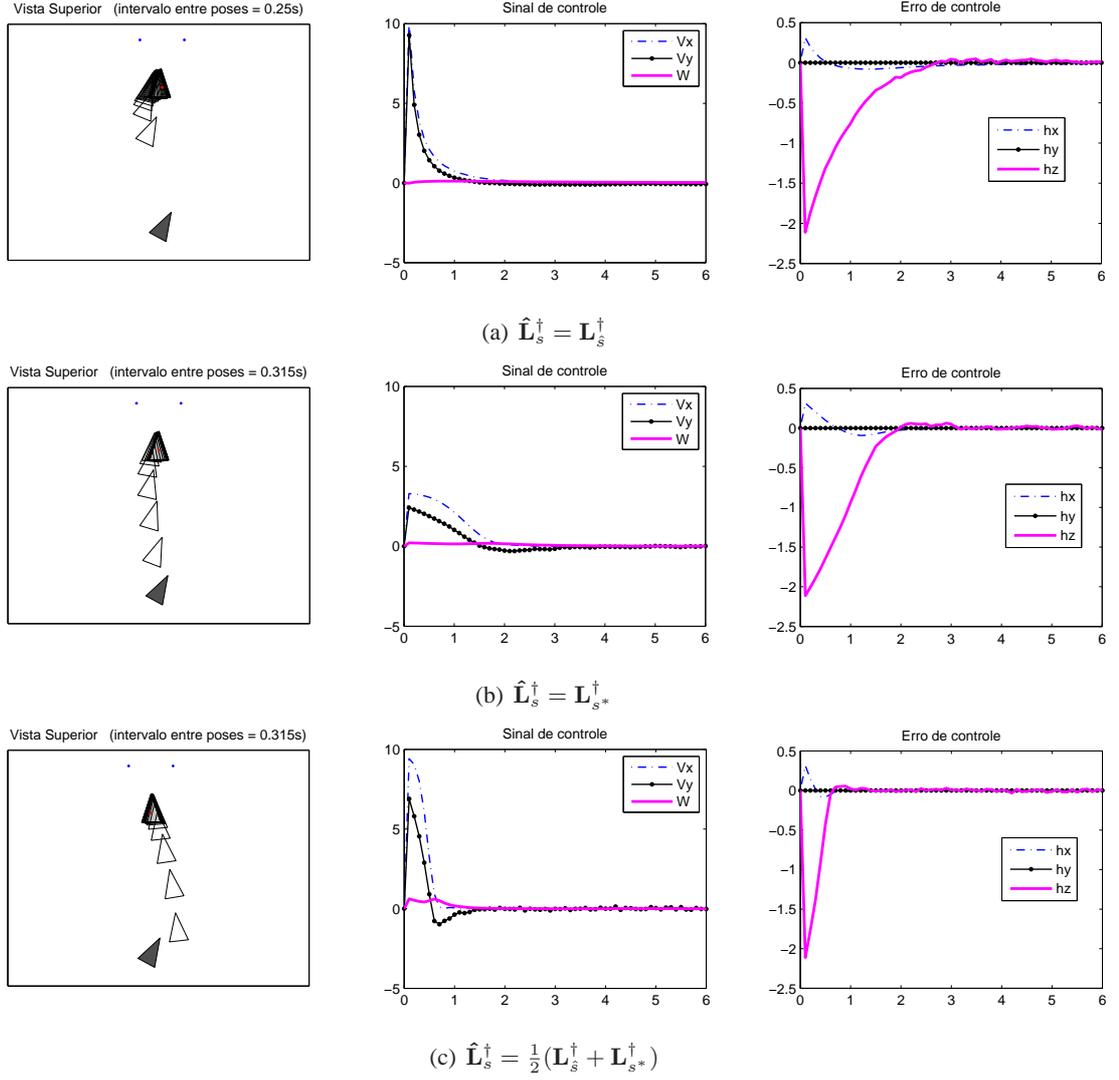


Figura 4.6: Trajetórias e sinais de controle gerados pelo controlador *IBVS* partindo da posição 4 descritas na Figura D.1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.

$\hat{\mathbf{L}}_s^\dagger = \mathbf{L}_s^\dagger$ também nos leva a um problema de minimização do tipo GNM, e conseqüentemente, a taxa de convergência é lenta, como pode-se perceber pela grande quantidade de triângulos na vizinhança do alvo.

Uma terceira alternativa é utilizar a abordagem proposta em [61] em que utiliza-se $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_s^{*})$ e resultando em convergência pelo menos quadrática. Como pode-se observar na Figura 4.6(c), este método resulta em sinais de controle bem comportados e uma trajetória que converge logo no início do movimento, o que pode ser notado pela menor quantidade de triângulos na vizinhança do ponto vermelho.

Assim sendo, os mesmos ensaios realizados com o controlador *PBVS* foram repetidos para o controlador *IBVS* usando $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_s^{*})$, pólos em $p = -10$ e vetor $\mathbf{b} = \mathbf{0}_{3 \times 12}$. Os resultados são apresentados na Figura 4.7, onde percebe-se que o controlador convergiu para a referência em todas as situações testadas com trajetórias suaves e sinais de controle decrescentes no tempo.

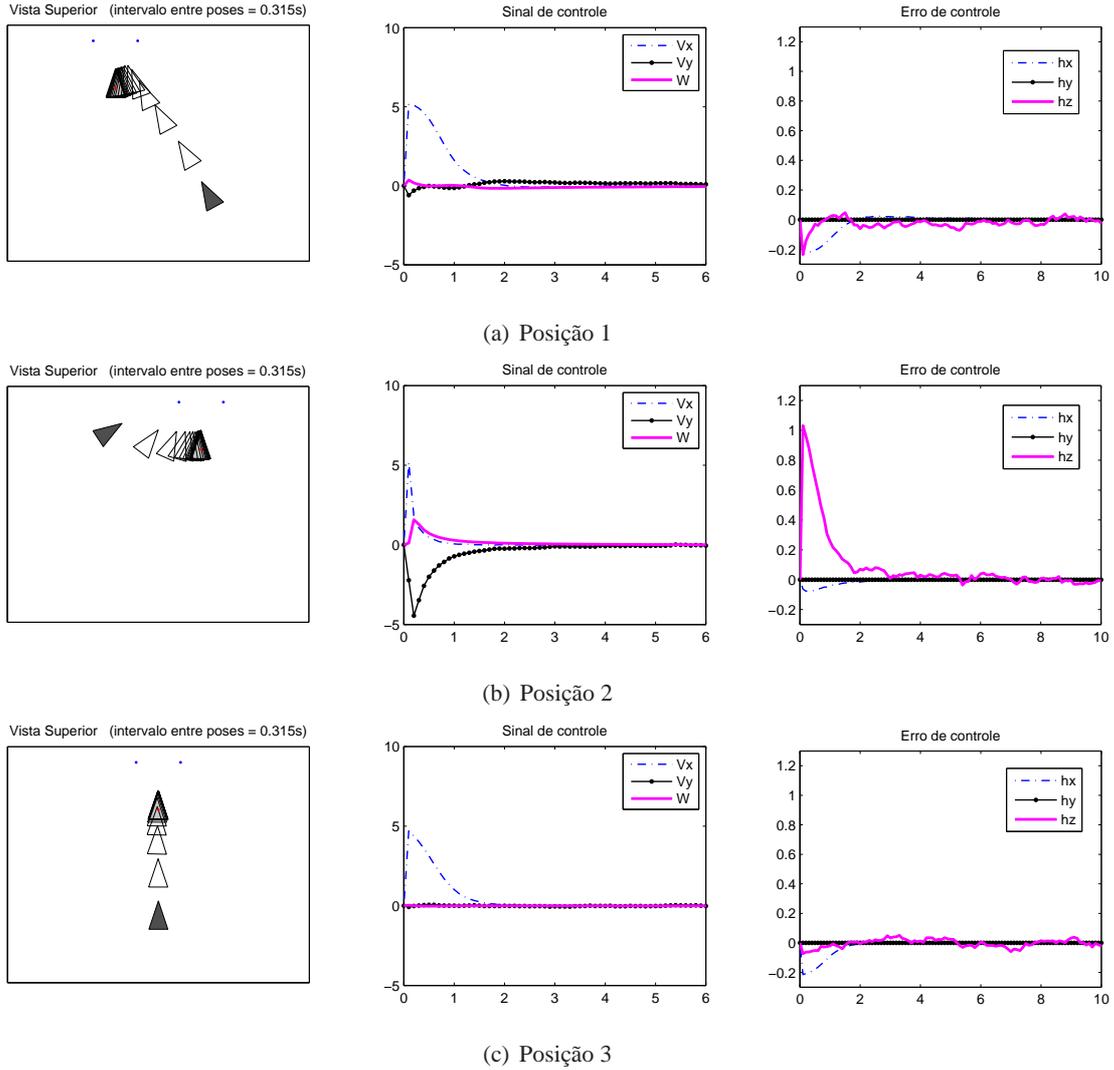


Figura 4.7: Trajetórias e sinais de controle gerados pelo controlador $IBVS$ com $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_{s^*}^\dagger)$ partindo respectivamente das posições 1, 2 e 3 descritas na Figura D.1. O triângulo escuro representa a pose inicial, enquanto os demais triângulos representam poses em instantes de tempo igualmente espaçados.

4.2.3 Controle servo-visual e estimação simultâneos

A seguir, uma testes em simulação foram feitos com o intuito de avaliar o desempenho do método proposto quando ambos, estimador e controle, atuam simultaneamente. Assim sendo, os mesmos testes realizados sobre o controlador $PBVS$ e $IBVS$ com $\hat{\mathbf{L}}_s^\dagger = \frac{1}{2}(\mathbf{L}_s^\dagger + \mathbf{L}_{s^*}^\dagger)$ foram repetidos utilizando desta vez, a estimativa $\hat{\lambda}$ provida pelo EKF para o cálculo do sinal de controle. Nestes testes, o EKF foi inicializado com $\hat{\mathbf{u}}(0) = \begin{pmatrix} 0 & -1 & 0 \end{pmatrix}^T$, $\hat{\mathbf{h}}(0) = \begin{pmatrix} 3 & 0 & 0 \end{pmatrix}^T$ para ambas as retas, com desvio padrão de medição de $0,5 \text{ pixel}$ tanto na horizontal quanto na vertical e desvio padrão de odometria de $0,05 \text{ m/s}$ para velocidade de translação e $1^\circ/\text{s}$ para velocidade de rotação.

Como consequência da rápida convergência do EKF implementado, as trajetórias resultantes e os sinais de controle gerados foram aproximadamente os mesmos dos experimentos anteriores, com pequenas flutuações devido aos erros de estimação que, apesar de presentes, não provocaram variação

significativa no resultado na tarefa de controle.

Por fim, verificou-se em simulação a influência de uma estimação ruim no desempenho do sistema. Para isso, erros de calibração foram adicionados ao modelo do simulador. Observou-se claramente que quanto maiores os erros introduzidos, pior era o desempenho dos controladores, sobretudo do controlador *PBVS*. Como sabe-se, tanto a abordagem *IBVS* quanto a *PBVS* utilizam valores provenientes do estimador no cálculo dos sinais de controle. Assim sendo, os dois métodos são afetados em caso de uma estimação grosseira. No entanto, enquanto no controlador *IBVS* tais parâmetros aparecem apenas no cálculo da Matriz de Interação L_s , no caso *PBVS* eles estão presentes não só em L_s , mas também no sinal de erro e , tornando o visivelmente mais sensível a imprecisões na estimação e a erros de calibração, como mencionado na Seção 2.1.1.3.

4.3 VALIDAÇÃO EXPERIMENTAL NA PLATAFORMA OMNI

Em uma segunda etapa, o método proposto foi avaliado sobre a plataforma robótica Omni, descrita brevemente na Seção 1.2.1 do Capítulo de Introdução. Em todos experimentos utilizou-se uma configuração binocular que consiste de um par de câmeras *FireWire* da Videre Design, modelo STH-MDCS3, fixada sobre uma base de 23 cm, resolução de 320 x 240 pixels e 15 fps. Os parâmetros de calibração do par de câmeras são:

$$\mathbf{K} = \begin{pmatrix} 175 & 0 & 150,3648 \\ 0 & 175 & 120,6602 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{R}_e^d = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \text{e} \quad \mathbf{t}_e^d = \begin{pmatrix} -0,2334 \\ 0 \\ 0 \end{pmatrix}. \quad (4.1)$$

com \mathbf{K} igual para ambas as câmeras.

Nas avaliações experimentais realizadas, o EKF foi inicializado com $\hat{\mathbf{u}}(0) = \begin{pmatrix} 0 & -1 & 0 \end{pmatrix}^T$ e $\hat{\mathbf{h}}(0) = \begin{pmatrix} 3 & 0 & 0 \end{pmatrix}^T$ para ambas as retas, o desvio padrão de medição adotado é de 0,5 pixel e o desvio padrão de odometria corresponde a 0,05 m/s para velocidade de translação e 1 °/s para velocidade de rotação. Devido às dimensões e ao peso do robô, por motivos de segurança restringiu-se a velocidade máxima do sistema a 0,20 m/s para velocidades de translação e 10 °/s para velocidade de rotação e a aceleração máxima do sistema a 0,04 m/s² para translação e 2 °/s² para rotação. As poses iniciais utilizadas em todos os experimentos encontram-se especificadas em detalhes no Anexo D..

4.3.1 Somente Estimação dos parâmetros da porta

Primeiramente, uma seqüência de pares de imagens foi colhida com o robô Omni em movimento, operado por um *joystick*, enquanto seu sistema de odometria armazenava os dados referentes ao deslocamento do robô. A partir dessas informações, avaliou-se o desempenho do estimador quando sob operação em condições reais. Na Figura 4.8 tem-se uma seqüência de imagens da câmera esquerda durante o processo descrito combinado com as projeções das estimativas do EKF nos respectivos instantes de tempo. A Figura 4.9 mostra a evolução da estimativa dos parâmetros 3D de ambas retas, enquanto a Tabela 4.4 apresenta as estimativas finais para os parâmetros quando o robô estava posici-

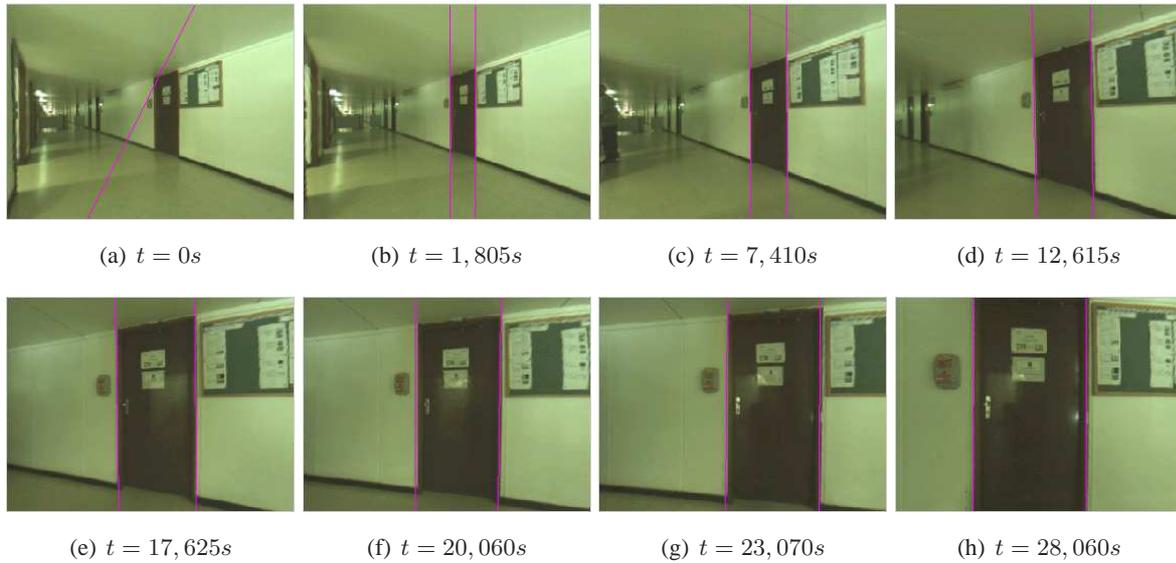


Figura 4.8: Sequência de imagens da câmera esquerda. As linhas em magenta correspondem à projeção da estimativa atual do EKF.

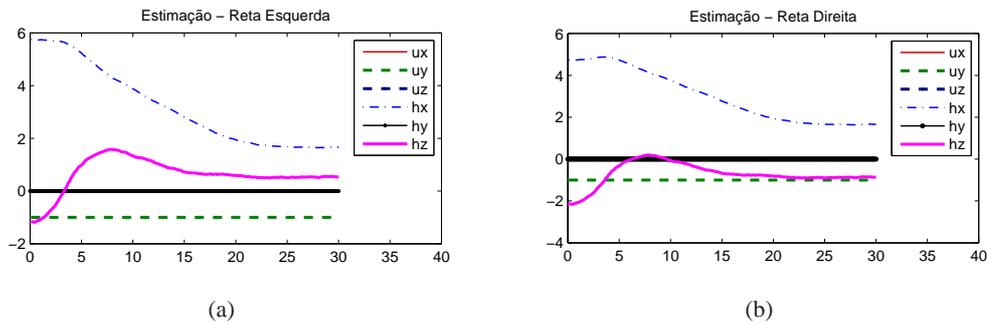


Figura 4.9: Resultado do experimento de estimação com robô guiado por *joystick*. (a) Estimação dos parâmetros 3D da lateral esquerda da porta. (b) Estimação dos parâmetros 3D da lateral direita da porta.

Tabela 4.4: Resultados Finais da Estimação

Lateral	u_x	u_y	u_z	h_x [m]	h_y [m]	h_z [m]
Esquerda	-0,0005	-0,9985	0,0011	1,9974	-0,0098	0,5141
Direita	0,0025	-0,9990	0,0009	1,9536	0,0021	-0,3958

onado a cerca de 2 m da porta com orientação aproximadamente perpendicular. Para ambas as retas, \mathbf{u} é quase vertical ($u_y \approx 1$) e \mathbf{h} é aproximadamente perpendicular a ele com norma $\|\mathbf{h}\| \approx 2$ m. O tempo médio de processamento para cada laço de estimação é de 274,45 ms calculado após 100 ciclos e considerado compatível para uso simultâneo com um controlador servo-visual.

4.3.2 Controle servo-visual IBVS

A estratégia de controle *IBVS* foi avaliada experimentalmente em duas situações. Na primeira, o robô encontra-se perpendicular e em frente à porta, bastando aproximar-se em linha reta para atingir

o objetivo de *docking*, enquanto na segunda situação, o robô está numa pose em diagonal, sendo necessário corrigir não apenas sua posição, mas também sua orientação para alcançar a pose alvo (cf. Figura D..2). A tarefa de *docking* foi concluída com sucesso em ambos experimentos e as Figuras 4.10, 4.11, 4.12 e 4.13 mostram os dados de estimação e controle colhidos durante as duas execuções.

Pode-se perceber que em ambas situações, a estimativa dos parâmetros 3D das retas estabiliza-se rapidamente já com a primeira medição e mantém-se praticamente constante até o instante t_1 em que o controlador é iniciado. A partir daí, o robô entra em movimento e a estimativa é atualizada pelo EKF utilizando informações de odometria e imagem fornecidos a cada 500 *ms*. O cálculo dos sinais de controle é realizado à mesma taxa que o filtro e utiliza as informações das estimativas para determinar a Jacobiana da imagem e o erro de controle definido pela diferença entre a projeção das retas na imagem e a projeção de referência definida previamente à partir das especificações do usuário. Observa-se pelos gráficos que o erro de controle apresenta oscilações, mesmo após convergência. Tal fato deve-se aos ruídos de medição propagados para as observações visuais extraídas das imagens. Nos experimentos realizados também observa-se que tais ruídos fazem com que exista não apenas uma pose alvo para o *docking*, mas sim uma pequena região em torno da qual o erro de controle tende a zero. Com isso, dependendo do ponto de partida do robô, pode-se chegar a posições finais em que as retas assemelham-se às referências em ambas câmeras, porém o robô encontra-se com a orientação levemente errada (cf. Figura 4.14).

4.3.3 Controle servo-visual *PBVS*

Primeiramente, investigou-se o comportamento do controle *PBVS* nas mesmas condições utilizadas nos experimentos para o controle *IBVS*. Foi observado que um período de amostragem de 500 *ms* apresentou problemas de estabilização, oscilando em torno da posição final de controle. Tal comportamento pode ser influenciado por uma série de fatores como um estimador com dinâmica lenta quando comparada ao controlador, atrasos devido a cálculos e processamento das imagens e períodos de amostragem muito grandes. No entanto, uma análise mais aprofundada de tais fenômenos é bastante complexa e está fora do escopo deste trabalho

Assim sendo, procurou-se apenas identificar um período de amostragem suficiente para avaliar a estratégia *PBVS* proposta. Fazendo uso do simulador em *Matlab*, chegou-se ao valor de 300 *ms* para o período de amostragem do sistema. Esse valor de período foi utilizado para avaliar o controle baseado em posição nas mesmas situações propostas na Seção 4.3.2. A tarefa de *docking* foi concluída com sucesso em ambos experimentos e as Figuras 4.15, 4.16, 4.17 e 4.18 apresentam os dados de estimação e controle colhidos durante as duas execuções.

Observa-se nos gráficos que a estimação comportou-se de forma similar aos experimentos com *IBVS*, fato já esperado pois o uso de uma configuração binocular torna a excitação do filtro eficiente para cenas *indoor*, independente da trajetória seguida pelo robô. Já os sinais de controle apresentando foram mais suaves que os obtidos na estratégia anterior, graças ao uso de estimativas filtradas pelo FKE, que são mais robustas aos ruídos de medição se comparadas às observações visuais extraídas diretamente das imagens. Como o alvo de controle *PBVS* está definido no espaço 3D de parâmetros das retas e a incerteza de estimação para tais parâmetros é bastante pequena, a região de *docking* na qual o erro de controle tende a zero é limitada, ao contrário do controle *IBVS*, resultando em uma

postura final bastante próxima da desejada, com orientação aproximadamente perpendicular à porta.

Em um outro experimento (*cf.* Figura 4.20), procurou-se verificar a tolerância do sistema a falhas, forçando uma situação em que uma das retas sai momentaneamente do campo de visão do sistema de câmeras (instante t_2). Com isso, o algoritmo de rastreamento responsável por indicar quais segmentos de retas extraídos das imagens correspondem à porta entra em falha, obrigando o estimador a prosseguir por predição até a reta voltar ao campo de visão do robô e ser detectada pelo algoritmo de rastreamento (instante t_3). Numa abordagem com características visuais extraídas diretamente da imagem, a perda do rastreamento seja por obstrução, falha do rastreador ou qualquer outro motivo, implica em suspensão da ação de controle por falta de sinal de entrada para ser comparado à referência.

Os resultados experimentais dos testes com o robô Omni mostram que o uso de características visuais obtidas de um filtro estocástico possui vantagens quando comparado ao uso de parâmetros extraídos diretamente das imagens pois além de apresentar maior robustez à ruídos de medição, essa escolha permite que o sistema continue operando por predição em casos de falha momentânea do algoritmo de rastreamento ou obstrução do alvo, como foi observado nos experimentos com o controlador *PBVS*. Associado à isso, a abordagem de controle *PBVS* tem uma convergência mais rápida e suave, como já havia sido observado nas simulações computacionais. Por fim, a abordagem *IBVS* ainda apresenta o inconveniente de possuir uma região no espaço cartesiano na qual o erro no espaço da imagem é nulo, levando o controlador à estabilização fora da posição final desejada. No entanto, o controle *PBVS* apresenta comportamento instável a baixas taxas de amostragem, o que representa uma possível restrição de uso em plataformas com limitação de velocidade de processamento.

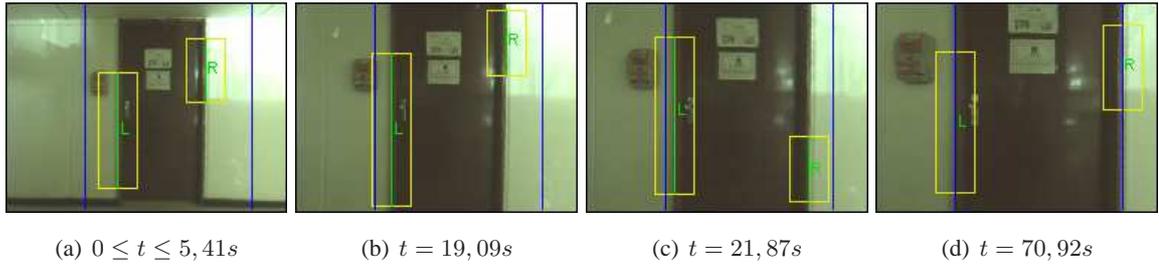


Figura 4.10: Seqüência de imagens da câmara esquerda durante o experimento de estimação e controle *IBVS* simultâneos partindo da posição 1 descrita na Figura D..2. As linhas verdes são as laterais esquerda (L) e direita (R) da porta e as azuis são a referência de controle.

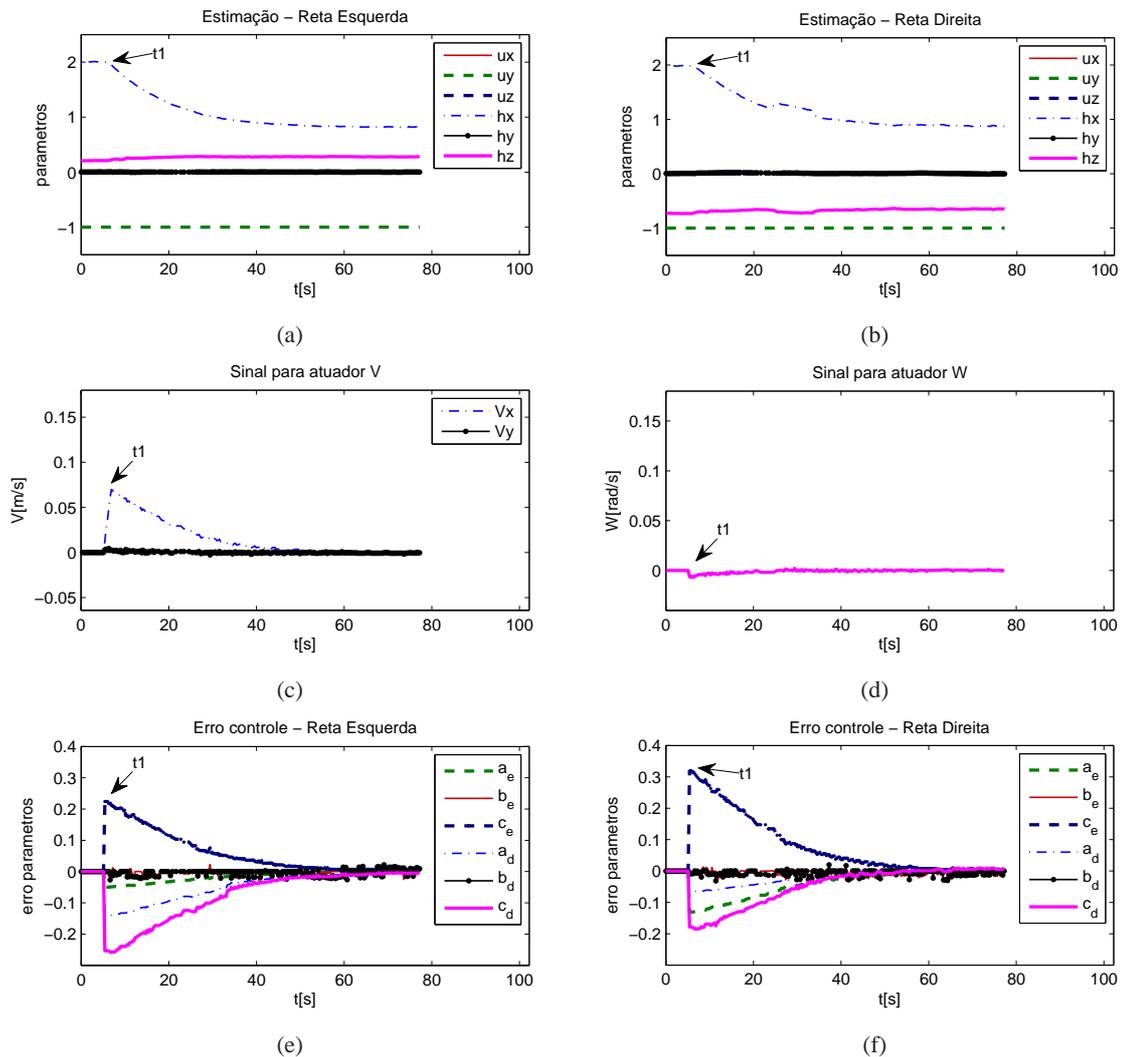


Figura 4.11: Resultados do experimento de estimação e controle *IBVS* simultâneos partindo da posição 1 descrita da Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) Erro de controle referente aos parâmetros a, b e c da projeção da reta esquerda em ambas as câmeras. (f) Erro de controle referente aos parâmetros a, b e c da projeção da reta direita em ambas as câmeras. $t_1 = 5,41s$ corresponde ao instante de início da ação de controle.

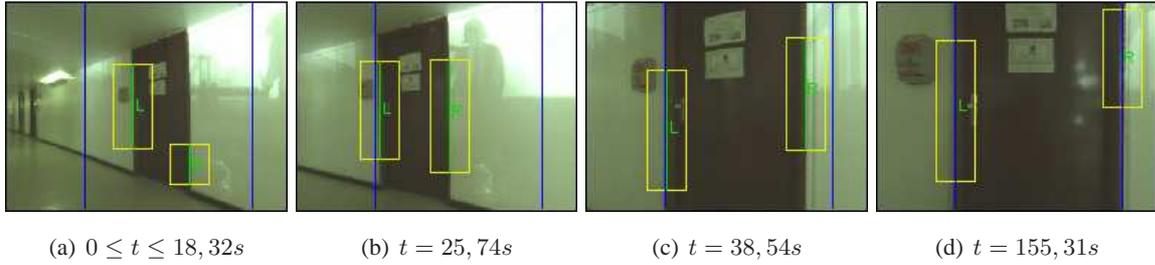


Figura 4.12: Seqüência de imagens da câmara esquerda durante o experimento de estimação e controle *IBVS* simultâneos partindo da posição 2 descrita na Figura D..2. As linhas verdes são as laterais esquerda (L) e direita (R) da porta e as azuis são a referência de controle.

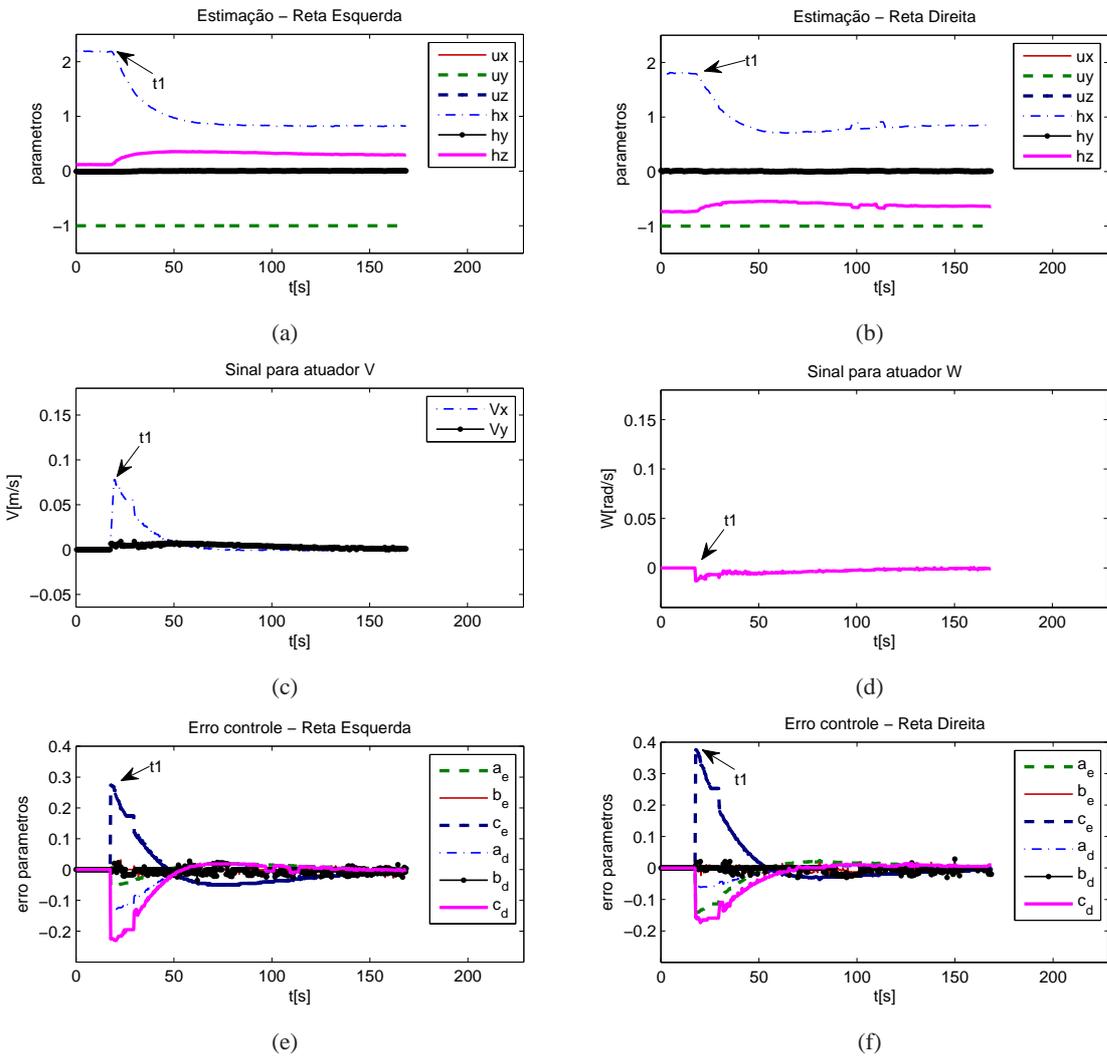


Figura 4.13: Resultados do experimento de estimação e controle *IBVS* simultâneos partindo da posição 2 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) Erro de controle referente aos parâmetros a, b e c da projeção da reta esquerda em ambas as câmeras. (f) Erro de controle referente aos parâmetros a, b e c da projeção da reta direita em ambas as câmeras. $t_1 = 18,32$ corresponde ao instante de início da ação de controle.

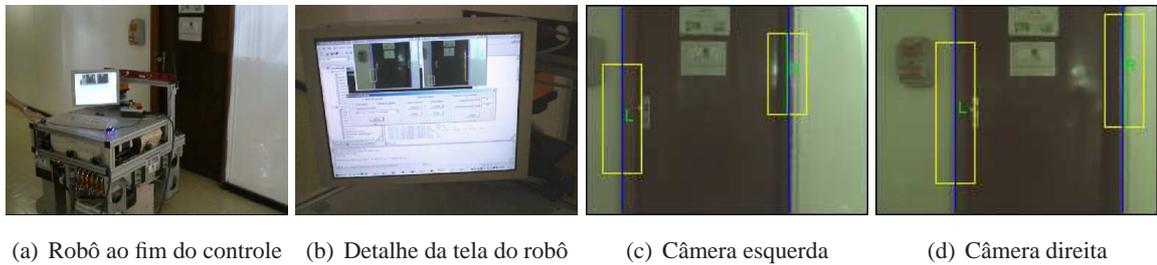


Figura 4.14: Situação em que o controle *IBVS* converge para a referência no espaço da imagem com posição diferente da desejada para o *docking*. (a) Robô com pequeno erro final de orientação. (c) e (d) Imagens das câmeras. As retas plotadas correspondem às referências de controle. No caso, ambas coincidem com as laterais da porta apesar de o robô estar ligeiramente fora do alvo de *docking*.

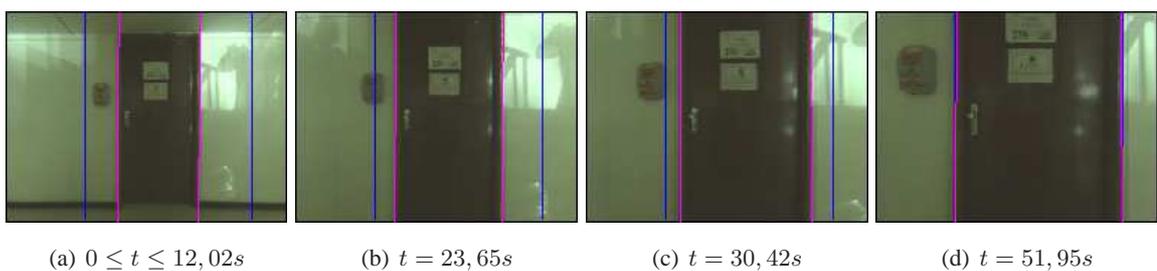


Figura 4.15: Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle *PBVS* simultâneos partindo da posição 1 descrita na Figura D.2. As linhas em magenta são as projeções da estimativa e as linhas azuis são a projeção da referência de controle.

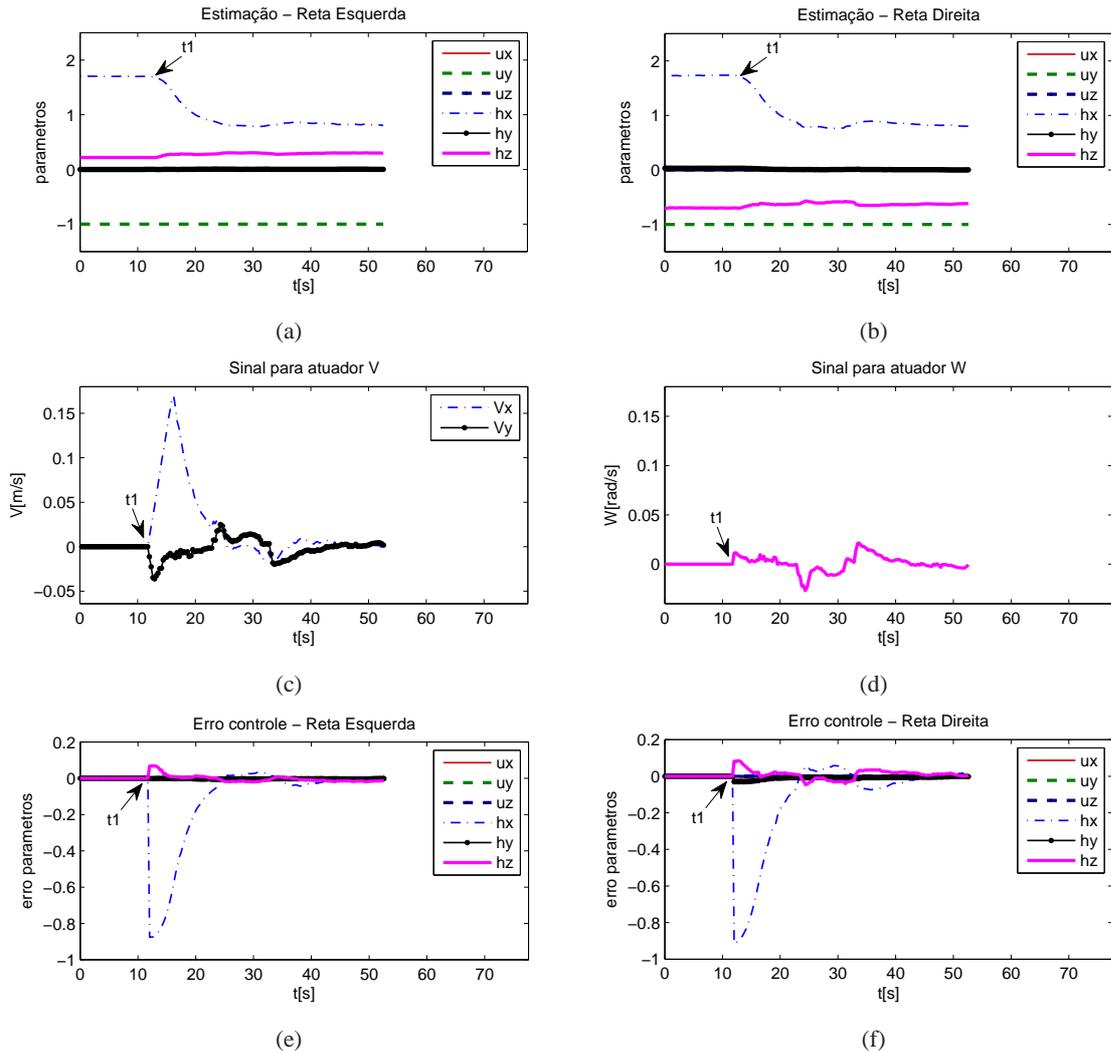


Figura 4.16: Resultados do experimento de estimação e controle *PBVS* simultâneos partindo da posição 1 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 12,02$ corresponde ao instante de início da ação de controle.

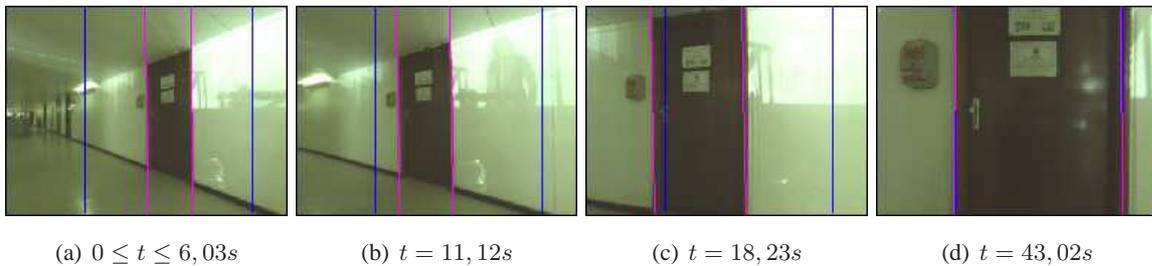


Figura 4.17: Seqüência de imagens da câmera esquerda durante o experimento de estimação e controle *PBVS* simultâneos partindo da posição 2 descrita na Figura D..2. As linhas em magenta são as projeções da estimativa e as linhas azuis são a projeção da referência de controle.

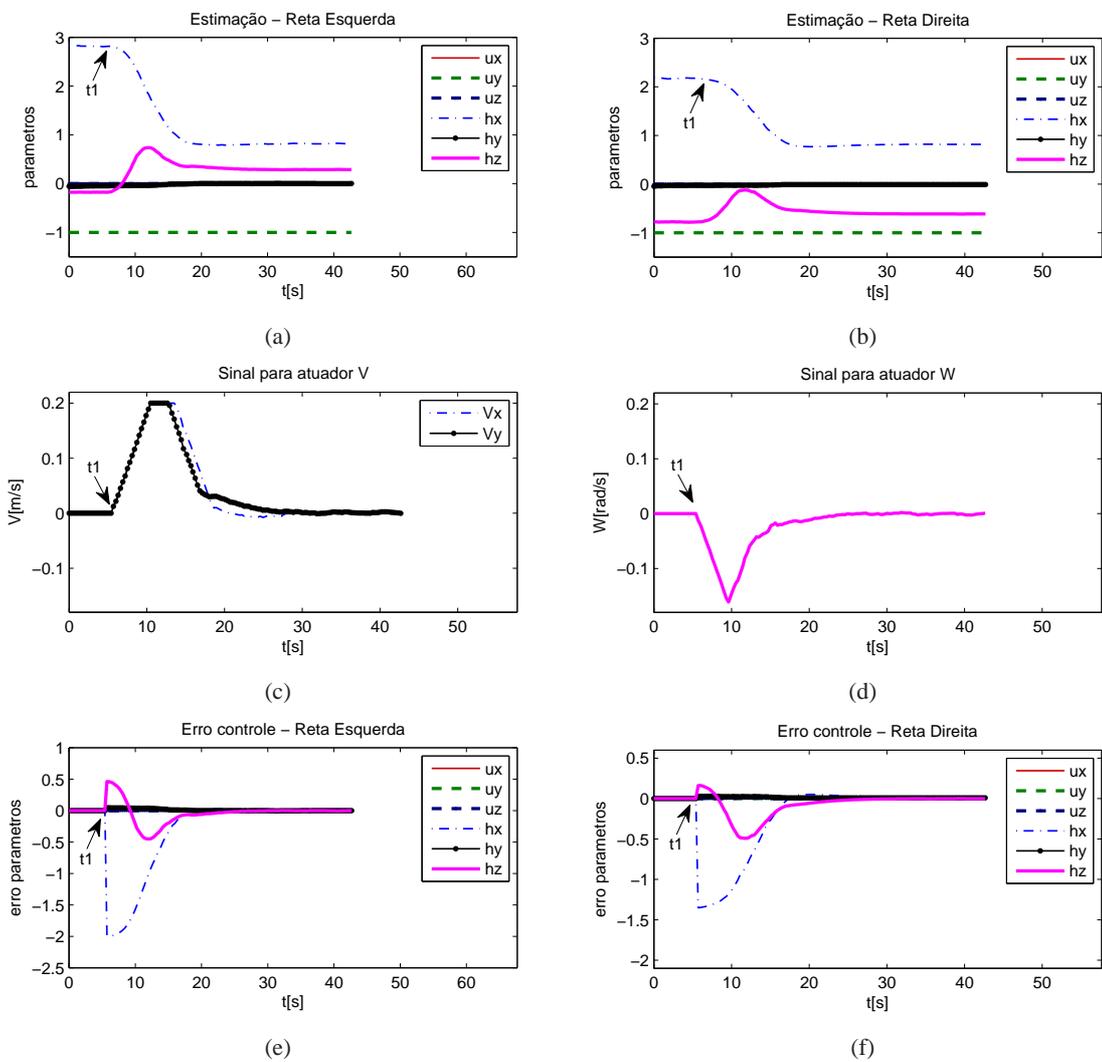


Figura 4.18: Resultados do experimento de estimação e controle *PBVS* simultâneos partindo da posição 2 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 6,03$ corresponde ao instante de início da ação de controle.

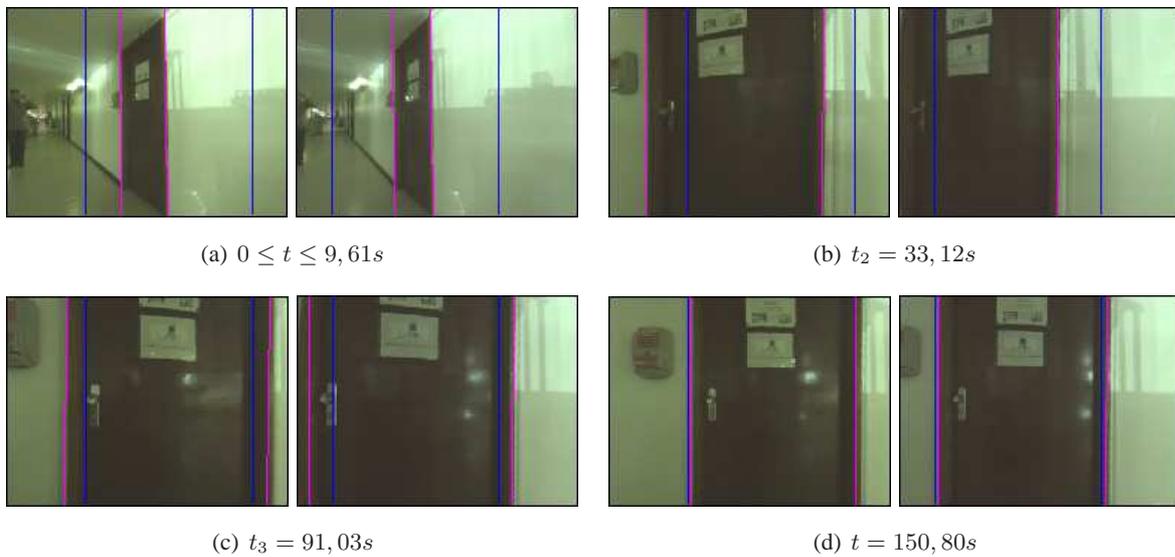


Figura 4.19: Seqüência de pares de imagens durante o experimento de estimação e controle *PBVS* simultâneos partindo da posição 3 descrita na Figura D..2. As linhas em magenta são as projeções das estimativas e as linhas azuis são a projeção da referência de controle. (a) Início da ação de controle. (b) Instante t_2 em que uma das laterais sai do campo de visão da câmara direita. (c) Instante t_3 em que a reta perdida volta a ser rastreada. (d) Imagens ao fim do controle.

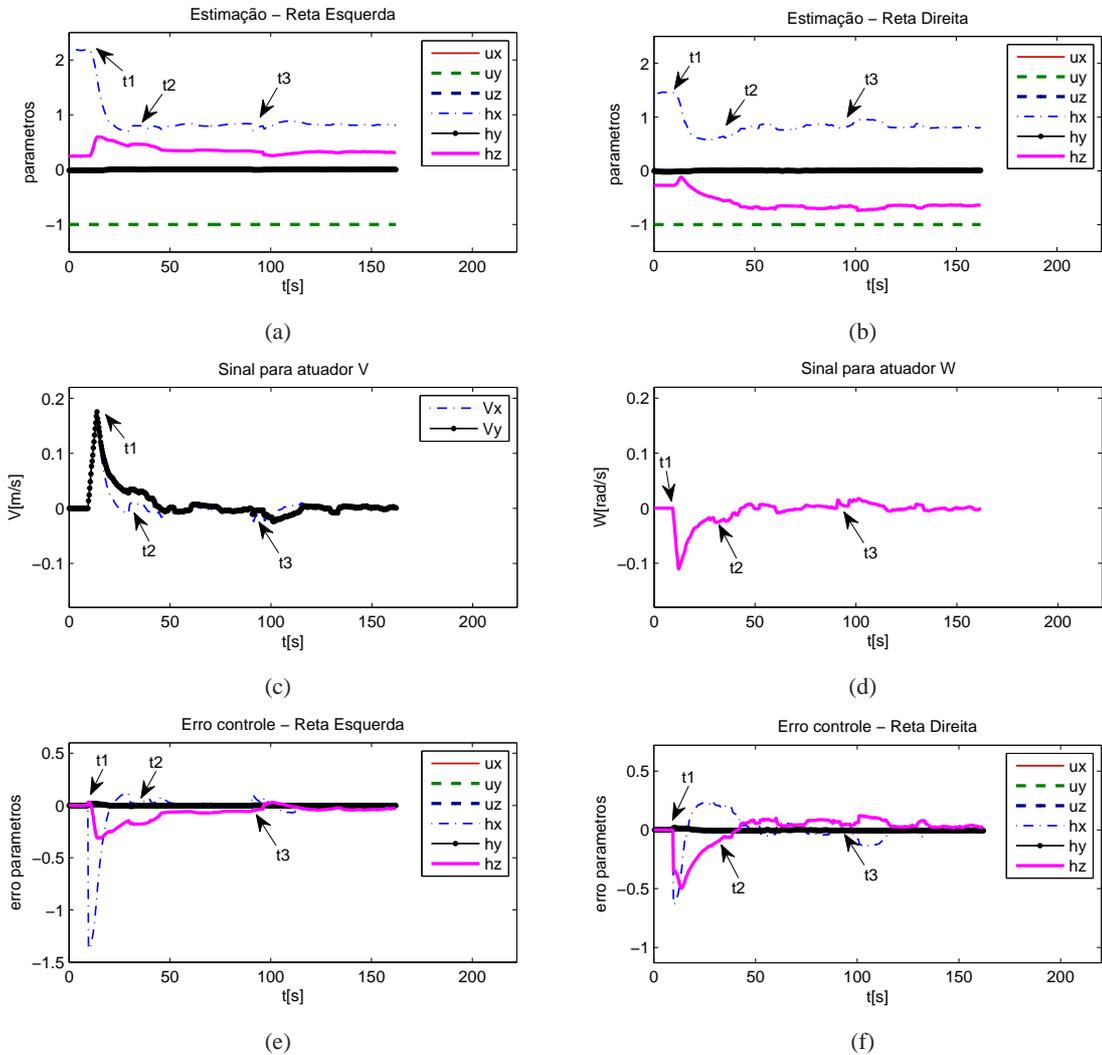


Figura 4.20: Resultados do experimento de estimação e controle *PBVS* simultâneos partindo da posição 3 descrita na Figura D..2. (a) e (b) Estimação dos parâmetros da reta esquerda e direita, respectivamente. (c) Velocidades de translação enviadas ao atuador. (d) Velocidade de rotação enviada ao atuador. (e) e (f) Erro de controle referente aos parâmetros 3D da reta esquerda e direita, respectivamente. $t_1 = 9,61$ corresponde ao instante de início da ação de controle. $t_2 = 33,12$ corresponde ao instante em que o *tracking* perdeu a reta direita. $t_3 = 91,03$ corresponde ao instante em que a reta direita volta a ser rastreada.

5 CONCLUSÕES

No one can be a great thinker who does not recognize that as a thinker it is his first duty to follow his intellect to whatever conclusions it may lead.

John Stuart Mill (1806 - 1873)

Este trabalho propôs uma estratégia de reconstrução e controle de trajetória para robôs móveis utilizando retas extraídas do ambiente como características visuais. De forma simplificada, o método desenvolvido pode ser dividido em duas etapas consecutivas: uma etapa de estimação dos parâmetros tridimensionais das retas seguida de outra etapa em que realiza-se o cálculo do sinal de controle a partir das informações visuais fornecidas pela planta.

Foi feita uma investigação dos vários modelos matemáticos para representação tridimensional de retas e selecionada a representação por coordenadas de Plücker para utilização neste trabalho. O problema de estimação dos parâmetros 3D das retas foi resolvido com o uso de métodos de filtragem estocástica, aplicando-se um Filtro de Kalman Estendido com restrições não-lineares.

Com relação ao controlador servo-visual propriamente dito, duas diferentes abordagens foram investigadas neste trabalho. Uma delas, conhecida classicamente na literatura por controle servo-visual baseado na imagem, também chamado 2D, utiliza-se de uma medida de erro no espaço da imagem para cálculo do sinal de controle. Na outra, chamada controle servo-visual baseado em posição, ou 3D, tal medida de erro ocorre em um espaço tridimensional. Usualmente, controladores servo visuais de abordagem 3D são tomados nas coordenadas cartesianas do espaço de trabalho do robô. No entanto, a abordagem aqui proposta não utiliza explicitamente a posição do robô, mas sim os parâmetros 3D das retas para determinar o valor do sinal de controle enviado aos atuadores.

O método desenvolvido neste trabalho foi avaliado em uma tarefa de *docking* em frente à uma porta, sem conhecimento *a priori* de sua posição inicial com relação ao robô. Tanto nos experimentos em simulação e quando nos realizados sobre uma plataforma robótica real, foi verificado o funcionamento satisfatório do controlador servo visual proposto, em que o robô atingiu com sucesso a posição alvo de *docking*.

Ambas abordagens 2D e 3D foram avaliadas experimentalmente, confirmando diversas vantagens e desvantagens já atribuídas na literatura a cada uma das estratégias. Enquanto a abordagem *PBVS* é mais susceptível a erros de calibração e requer maiores taxas de amostragem para evitar problemas de instabilidade, o controle *IBVS* apresenta convergência mais lenta na vizinhança da posição final, sinais de controle muito grandes quando longe da referência, menor robustez a ruídos nas imagens e presença de regiões de erro nulo de controle fora configuração final desejada.

Por fim, verificou-se que a escolha de características visuais provenientes de um filtro estocástico ao invés de se utilizar parâmetros extraídos diretamente das imagens possibilita que o sistema continue atuando por predição mesmo quando há perda do *tracking*, tornando a abordagem tolerante a situações de falha e menos susceptível a ruídos de medição.

5.1 PROPOSTAS PARA TRABALHOS FUTUROS

Trabalhos futuros inicialmente se concentrarão na extensão da abordagem proposta a outras situações similares de navegação de robôs móveis, como por exemplo, o percorrimto de corredores utilizando as duas retas horizontais formadas pela junção das paredes com o piso. Outra possibilidade consiste em investigar mais profundamente a possibilidade de utilizar o termo adicional da solução generalizada do ganho de controle para gerar trajetórias que excitem melhor o estimador estocástico. Uma outra perspectiva de pesquisa envolve explorar abordagens de sistemas híbridos para combinar os métodos de controle 2D e 3D. Com isso, busca-se tirar proveito das vantagens oferecidas por cada tipo de controle servo-visual e ao mesmo tempo evitar seus pontos fracos.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Y. Ma, S. Soatto, J. Košecák, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Interdisciplinary Applied Mathematics. Springer, 2004.
- [2] Y. Shirai and H. Inoue. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition*, 5:99–108, 1973.
- [3] J. Hill and W. T. Park. Real time control of a robot with a mobile camera. *Proceedings of 9th ISIR*, pages 233 – 246, March 1979.
- [4] L. E. Weiss, A. C. Sanderson, and C. P. Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation*, RA-3(5):404–417, Outubro 1987.
- [5] J. T. Feddema and O. R. Mitchell. Vision-guided servoing with feature-based trajectory generation. *IEEE Transactions on Robotics and Automation*, 5(5):691–700, Outubro 1989.
- [6] F. Chaumette, P. Rives, and B. Espiau. Positioning of a robot with respect to an object, tracking it and estimating its velocity by visual servoing. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:2248–2253, Abril 1991.
- [7] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, Jun 1992.
- [8] J. Wang and W. J. Wilson. 3D relative position and orientation estimation using Kalman filter for robot control. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:2638–2645, Maio 1992.
- [9] N. P. Papanikolopoulos and P. K. Khosla. Adaptive robotic visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38:429–445, 1993.
- [10] N. Maru, H. Kase, S. Yamada, and A. Nishikawa. Manipulator control by visual servoing with the stereo vision. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:1866–1870, 1993.
- [11] G. D. Hager. A modular system for robust positioning using feedback from stereo vision. *IEEE Transactions on Robotics and Automation*, 3(4):582–595, Agosto 1997.
- [12] E. Malis, F. Chaumette, and S. Boudet. 2-1/2-D visual servoing. *IEEE Transactions on Robotics and Automation*, 15(2):238–250, Abril 1999.
- [13] D. B. Zhang, L. Van Gool, and A. Oosterlinck. Stochastic predictive control of robot tracking systems with dynamic visual feedback. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:610–615, 1990.
- [14] V. Lippiello, B. Siciliano, and L. Villani. Position-based visual servoing in industrial multi-robot cells using a hybrid camera configuration. *IEEE Transactions on Robotics*, 23(1):73–86, Fevereiro 2007.

- [15] R. C. Harrell, D. C. Slaughter, and P. D. Adsit. A fruit-tracking system for robotic harvesting. *Machine Vision and Applications*, 2(2):69–80, Março 1989.
- [16] E. D. Dickemanns and V. Graefe. Application of dynamic monocular machine vision. *Machine Vision and Applications*, 1(4):241–261, 1988.
- [17] E. D. Dickmanns and F. R. Schell. Autonomous landing of airplanes by dynamic machine vision. *Applications of Computer Vision, Proceedings, 1992., IEEE Workshop on*, 1:172–179, Novembro 1992.
- [18] A. Krupa, J. Gangloff, C. Doignon, M. F. Mathelin, G. Moreland J. Leroy, L. Soler, and J. Marescaux. Autonomous 3-D positioning of surgical instruments in robotized laparoscopiesurgery using visual servoing. *IEEE Transactions on Robotics and Automation*, 19(5):842–853, Outubro 2003.
- [19] R. L. Andersson. Dynamic sensing in a ping-pong playing robot. *IEEE Transactions on Robotics and Automation*, 5(6):728–739, Dezembro 1989.
- [20] E. S. Espinoza-Quesada and L. E. Ramos-Velasco. Visual servoing for an inverted pendulum using a digital signal processor. *IEEE International Symposium on Signal Processing and Information Technology*, 1:76–80, Agosto 2006.
- [21] P. I. Corke. Visual control of robot manipulators - a review. In *Visual Servoing*, pages 1–31, 1993.
- [22] P. Rives and J. Borrelly. Visual servoing techniques applied to an underwater vehicle. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:1851–1856, 1997.
- [23] H. Zhang and J. P. Ostrowski. Visual servoing with dynamics: Control of an unmanned blimp. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:618–623, Maio 1999.
- [24] G. L. Mariottini, D. Prattichizzo, and G. Oriolo. Epipole-based visual servoing for nonholonomic mobile robots. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:497–503, Abril 2004.
- [25] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki. Visual servoing for non-holonomic mobile robots. *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 2:1133–1140, 1994.
- [26] D. P. Tsakiris, P. Rives, and C. Samson. Extending visual servoing techniques to nonholonomic mobile robots. In *The Confluence of Vision and Control, Lecture Notes in Control and Information System*, volume 237. Springer-Verlag, 1998.
- [27] Y. Ma, J. Košecák, and S. S. Sastry. Vision guided navigation for a nonholonomic mobile robot. *Proceedings of Conference on Decision & Control*, 1:3069–3074, 1997.
- [28] J. Chen, W. E. Dixon, D. M. Dawson, and V. K. Chitrakaran. Visual servo tracking control of a wheeled mobile robot with a monocularfixed camera. *Proceedings of IEEE International Conference on Control Applications*, 2:1061–1066, Setembro 2004.

- [29] J. Carvalho, P. Rives, A. Santa-Bárbara, and S. Bueno. Visual servo control for a class of mobile robot. *Proceedings of IEEE International Conference on Control and Applications*, 1:431–436, 2000.
- [30] G. Silveira. Controle servo visual de veículos robóticos aéreos. Master’s thesis, Universidade Estadual de Campinas, 2002.
- [31] G. Silveira, J. Carvalho, P. Rives, J. Azinheira, and S. Bueno and M. Madrid. Optimal visual servoed guidance of outdoor autonomous robotic airships. *Proceedings of American Control Conference*, 1:779–784, Maio 2002.
- [32] C. Castro, L. Dinnouti, S. Bueno, and G. Silveira. Uma abordagem integrada para controle servo visual e monitoramento em temporeal. *Congresso Brasileiro de Automática*, 1:1–6, 2004.
- [33] V. Grassi. Sistema de visão omnidirecional aplicado no controle de robôs móveis. Master’s thesis, Escola Politécnica da Universidade de São Paulo, 2002.
- [34] V. Grassi and J. Okamoto. Development of an omnidirecional vision system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28(1):58–68, 2006.
- [35] C. Weber, S. Wermter, and A. Zochios. Robot docking with neural vision and reinforcement. *Knowledge-Based Systems*, 17:165–172, 2004.
- [36] T. Martinez-Marin and T. Duckett. Learning visual docking for non-holonomic autonomous vehicles. *IEEE Intelligent Vehicles Symposium*, 1:1015–1020, Junho 2008.
- [37] T. Martinez-Marin and T. Duckett. Robot docking by reinforcement learning in a visual servoing framework. *Proceedings of IEEE Conference on Robotics, Automation and Mechatronics*, 1:159–165, 2004.
- [38] S. Zwaan, A. Bernardino, and J. Santos-Victor. Vision based station keeping and docking for an aerial blimp. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:614–620, 2000.
- [39] Z. Zhang, Z. Qu, and Q. Li. An autonomous docking simulation system based on monocular vision. *International Symposium on Systems and Control in Aerospace and Astronautics*, 1:1312–1315, Janeiro 2006.
- [40] R. C. Luo, C. T. Liao, K. L. Su, and K. C. Lin. Automatic docking and recharging system for autonomous security robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2953–2958, Agosto 2005.
- [41] L. Pan-Mook, J. Bong-Hwan, and K. Sea-Moon. Visual servoing for underwater docking of an autonomous underwater vehicle with one camera. *Proceedings of MTS/IEEE OCEANS*, 2:677–682, Setembro 2003.
- [42] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor. Visual servoing and appearance for navigation. *Robotics and Autonomous Systems*, 31:87–97, Abril 2000.

- [43] P. Rives and B. Espiau. Estimation récurrente de primitives 3d au moyen d'une caméra mobile. Technical Report 652, INRIA, 35042 Rennes Cedex, France, Março 1987.
- [44] J. Weng, T. Huang, and N. Ahuja. Estimating motion and structure from line matches: Performance obtained and beyond. *IEEE International Conference on Pattern Recognition*, 1:168–172, Junho 1990.
- [45] J. Weng, T. S. Huang, and N. Ahuja. Motion and structure from line correspondences: closed-form solution, uniqueness and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):318–336, Março 1992.
- [46] T. Lemaire and S. Lacroix. Monocular-vision based SLAM using line segments. *IEEE International Conference on Robotics and Automation*, pages 2791–2796, Abril 2007.
- [47] N. Andreff. *Asservissement visuel à partir de droites et auto-étalonnage pince-caméra*. PhD thesis, Institut National Polytechnique de Grenoble, 1999.
- [48] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. *Proceedings of the IEEE International Conference on Robotics and Automation*, 3:2070–2075, 2000.
- [49] Y. le Corre. *Conception et Commande d'un Robot Omnidirectionnel*. PhD thesis, Université de Montpellier II, Dezembro 1998.
- [50] C. S. R. Aguiar. Estudo da plataforma omni e calibração de câmara para localização 3-D. Technical report, Universidade de Brasília, 2006.
- [51] G. A. Borges. *Cartographie de l'environnement et localisation robuste pour la navigation de robots mobiles*. PhD thesis, Université de Montpellier II, 2002.
- [52] G. A. Borges and M. Aldon. Line extraction in 2D range images for mobile robotics. *Journal of Intelligent and Robotic Systems*, 40(3):267–297, Julho 2004.
- [53] G. A. Borges. Optimal mobile robot pose estimation using geometrical maps. *IEEE Transactions on Robotics and Automation*, 18:87–94, Fevereiro 2002.
- [54] B. V. Adorno, C. S. R. Aguiar, and G. Borges. Planejamento de trajetória para o robô omni utilizando o algoritmo mapade rotas probabilístico. *VIII Simpósio Brasileiro de Automação Inteligente*, 1:1–6, 2007.
- [55] J. Santos-Victor, A. Bernardino, and C. Silva. On the design of visual behaviors for autonomous systems. *Proceedings of IEEE International Symposium on Industrial Electronics*, 1:SS53–SS59, Julho 1997.
- [56] S. Hutchinson, G. D. Hager, and P. I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, Outubro 1996.
- [57] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control, Lecture Notes in Control and Information System*, volume 237. Springer-Verlag, 1998.

- [58] P. M. Shiroma. Controle por visão de veículos robóticos. Master's thesis, Universidade Estadual de Campinas, 2004.
- [59] P. I. Corke and S. A. Hutchinson. Real-time vision, tracking and control. *Proceedings of IEEE International Conference on Robotics and Automation*, 1:622–629, 2000.
- [60] C. R. Rao and S. K. Mitra. *Generalized Inverse of Matrices and its Applications*. John Wiley & Sons, 1 edition, 1971.
- [61] E. Malis. Improving vision-based control using efficient second-order minimization techniques. *Proceedings of IEEE International Conference on Robotics and Automation*, 2:1843–1848, Maio 2004.
- [62] K. S. Roberts. A new representation for a line. *Proceedings of Computer Society Conference on Computer Vision and Pattern Recognition*, 1:635–640, Junho 1988.
- [63] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Dover Publications, 1996.
- [64] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.
- [65] J. Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155:725–791, 1865.
- [66] D. A. Forsyth and J. Ponce. *Computer Vision, A Modern Approach*. Prentice-Hall of India, 2004.
- [67] R. Horaud and O. Monga. *Vision par ordinateur: outils fondamentaux*. Traité Des Nouvelles Technologies. Editions Hermès, 10 edition, 1995.
- [68] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis and Machine Vision*. Brooks/Cole Publishing Company, 2 edition, 1998.
- [69] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Monocular vision based SLAM for mobile robots. *18th International Conference on Pattern Recognition*, 3:1027–1031, 2006.
- [70] M. Spong, S. Hutchinson, and M. Vioyasagar. *Robot Modeling and Control*. John Wiley & Sons, 2006.
- [71] N. Navab, O. D. Faugeras, and T. Vieville. The critical sets of lines for camera displacement estimation: A mixed euclidean-projective and constructive approach. *Proceedings of the Fourth International Conference on Computer Vision*, 1:713–723, Maio 1993.
- [72] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1970.
- [73] C. K. Chui and G. Chen. *Kalman filtering with real-time applications, Series in Information Sciences*. Springer, 1991.
- [74] P. Corke and S. A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, Abril 2001.

- [75] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard and L. E. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms and Implementation*. The MIT Press, 2005.

A. CONFIGURAÇÃO DE CORPOS RÍGIDOS E TRANSFORMAÇÕES DE COORDENADAS

A configuração de um robô é uma representação matemática que permite tratá-lo como um ponto num espaço apropriado. Este modelamento permite que problemas que a princípio pareçam diferentes em termos de geometria e cinemática sejam resolvidos pela mesma abordagem. Nesta seção, são apresentados conceitos básicos sobre configurações de corpos rígidos no espaço tridimensional e o uso desta representação para descrever transformações entre diferentes sistemas coordenados. O desenvolvimento e os conceitos aqui apresentados podem ser encontrados em maior profundidade em [70, 75].

A Figura A..1 representa um corpo rígido atrelado a um sistema coordenado local \mathcal{O}^o com determinada posição e orientação em relação ao sistema coordenado global \mathcal{O}^w . Descreve-se a posição de um corpo rígido num espaço n -dimensional pelo grupo \mathbb{R}^n e sua orientação pelo grupo Grupo Especial Ortogonal SO^n . A composição de posição e orientação formam o vetor de postura do corpo rígido, descrito pelo grupo Grupo Especial Euclidiano SE^n .

Uma posição no espaço 3D é representada por um vetor $\mathbf{t} = (t_1 \ t_2 \ t_3)^T \in \mathbb{R}^3$. As coordenadas de \mathbf{p} representam a origem de \mathcal{O}^o expressa no sistema coordenado \mathcal{O}^w .

Já a representação de orientação se dá por uma matriz $\mathbf{R}_{3 \times 3}$ tal que

$$\mathbf{R} = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{pmatrix} = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \in SO^3, \quad (\text{A..1})$$

em que $\mathbf{x} = (x_1 \ x_2 \ x_3)^T$ é um vetor unitário na direção X^w do corpo rígido expressa no sistema

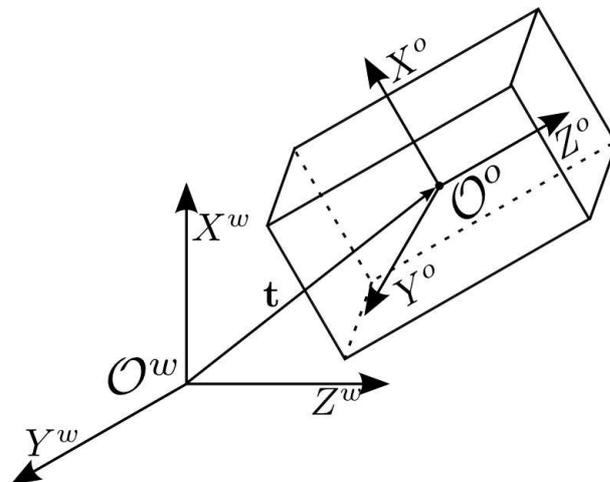


Figura A..1: Representação de um sistema coordenado global \mathcal{O}^w e um sistema coordenado local \mathcal{O}^o , atrelado a um corpo rígido.

coordenado global \mathcal{O}^w . Os vetores \mathbf{y} e \mathbf{z} são definidos similarmente.

A matriz \mathbf{R} é comumente conhecida como matriz de rotação. A representação de orientação através da matriz de rotação utiliza nove valores para representar os três graus de liberdade angulares, com as seguintes restrições:

$$\begin{aligned} \|\mathbf{x}\| = \|\mathbf{y}\| = \|\mathbf{z}\| = 1, \\ \mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{z} = \mathbf{z}^T \mathbf{x} = 0, \end{aligned} \quad (\text{A..2})$$

de forma que os vetores \mathbf{x} , \mathbf{y} e \mathbf{z} sejam todos unitários e ortogonais entre si. Com isso o determinante da matriz \mathbf{R} é sempre igual a 1. Matrizes que satisfaçam essas restrições pertencem ao SO^3 .

A representação da postura do corpo rígido é dada pela matriz homogênea $\mathbf{H}_{4 \times 4}$, formada por uma combinação das representações de posição e orientação:

$$\mathbf{H} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_{3 \times 3} & 1 \end{pmatrix} \in SE^3. \quad (\text{A..3})$$

Como ambos \mathbf{R} e \mathbf{t} possuem três graus de liberdade cada, a configuração de um corpo rígido no espaço tridimensional possui seis graus de liberdade.

Além de representar configurações de corpos rígidos, as matrizes dos grupos SO^n e SE^n podem ser usadas para representar mudanças de postura ou transformações de coordenadas entre sistemas (cf. Figura A..2). Nestes casos, um ponto representado num sistema coordenado \mathcal{O}^A pelo vetor \mathbf{p}^A , terá uma representação do sistema \mathcal{O}^B dada por \mathbf{p}^B , de forma que estes vetores se relacionam pela transformação

$$\mathbf{p}^B = \mathbf{R}_A^B \mathbf{p}^A + \mathbf{t}_A^B, \quad (\text{A..4})$$

em que \mathbf{R}_A^B é uma matriz de rotação que representa a orientação do sistema \mathcal{O}^A com relação ao sistema \mathcal{O}^B e \mathbf{t}_A^B é um vetor que representa a posição do sistema \mathcal{O}^A em relação ao sistema \mathcal{O}^B .

A transformação definida em A..4 pode ser representada por uma única multiplicação matricial utilizando a notação em coordenadas homogêneas

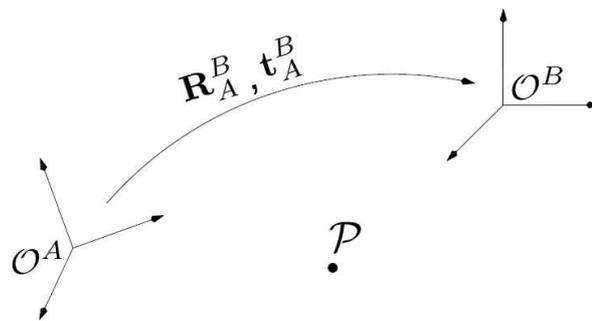


Figura A..2: Representação de uma transformação de sistemas coordenados.

$$\tilde{\mathbf{p}}^B = \mathbf{H}_A^B \tilde{\mathbf{p}}^A,$$

com

$$\tilde{\mathbf{p}}^A = \begin{pmatrix} \mathbf{p}^A \\ 1 \end{pmatrix}, \quad \tilde{\mathbf{p}}^B = \begin{pmatrix} \mathbf{p}^B \\ 1 \end{pmatrix} \quad e \quad \mathbf{H}_A^B = \begin{pmatrix} \mathbf{R}_A^B & \mathbf{t}_A^B \\ \mathbf{0}_{3 \times 3} & 1 \end{pmatrix}. \quad (\text{A..5})$$

B. PRODUTO VETORIAL E SUAS PROPRIEDADES

Aqui são apresentados conceitos básicos de produto vetorial e suas principais propriedades. Mais detalhes sobre o tema podem ser encontrados em [63], que foi a principal referência para esta seção.

No espaço tridimensional, o produto vetorial de dois vetores $\mathbf{a} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}^T$ e $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix}^T$, também chamado de produto externo, é representado geometricamente pela Figura B..1(a) é definido por

$$\mathbf{a} \times \mathbf{b} = \begin{vmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \left(\sum_{i,j,k=1}^3 \epsilon_{ijk} \mathbf{e}_i a_j b_k \right), \quad (\text{B..1})$$

ou mais simplificadamente

$$\mathbf{a} \times \mathbf{b} = \mathbf{c}, \quad c_i = \sum_{j,k=1}^3 \epsilon_{ijk} a_j b_k, \quad (\text{B..2})$$

$$\epsilon_{ijk} = \begin{cases} 1, & \text{se } (ijk) \text{ é } (123), (231) \text{ ou } (312), \\ -1, & \text{se } (ijk) \text{ é } (321), (213) \text{ ou } (132), \\ 0, & \text{caso contrário: } i = j \text{ ou } j = k \text{ ou } k = i. \end{cases} \quad (\text{B..3})$$

com ϵ_{ijk} , conhecido como Símbolo de Levi-Civita e definido por

Evidentemente,

$$\begin{aligned} \mathbf{a} \times \mathbf{b} &= -\mathbf{b} \times \mathbf{a}, & \mathbf{a} \times \mathbf{a} &= \mathbf{0}, \\ \mathbf{b}^T (\mathbf{a} \times \mathbf{b}) &= \mathbf{a}^T (\mathbf{a} \times \mathbf{b}) = \mathbf{0}. \end{aligned} \quad (\text{B..4})$$

A função Delta de Kronecker é definida por

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases}. \quad (\text{B..5})$$

O Símbolo de Levi-Civita está relacionado com o Delta de Kronecker. Em três dimensões, essa relação é dada por

$$\begin{aligned} \epsilon_{ijk} \epsilon_{lmn} &= \det \begin{pmatrix} \delta_{il} & \delta_{im} & \delta_{in} \\ \delta_{jl} & \delta_{jm} & \delta_{jn} \\ \delta_{kl} & \delta_{km} & \delta_{kn} \end{pmatrix} \\ &= \delta_{il} (\delta_{jm} \delta_{kn} - \delta_{jn} \delta_{km}) - \delta_{im} (\delta_{jl} \delta_{kn} - \delta_{jn} \delta_{kl}) + \delta_{in} (\delta_{jl} \delta_{km} - \delta_{jm} \delta_{kl}). \end{aligned} \quad (\text{B..6})$$

De (B..6) tem-se a identidade simplificada de épsilon:

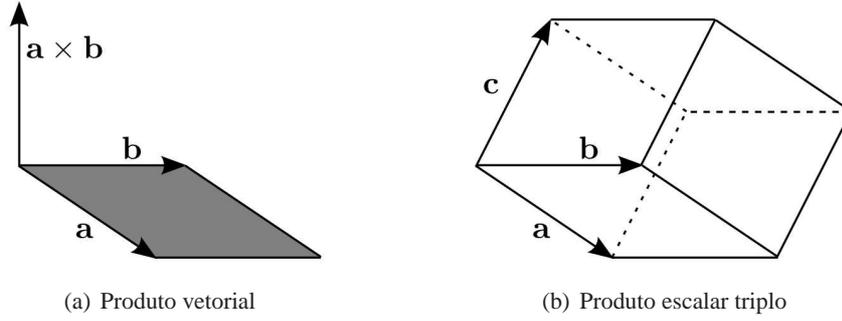


Figura B..1: Representação geométrica do produto vetorial e do produto escalar triplo.

$$\sum_{i,j=1}^3 \epsilon_{ijk} \epsilon_{imn} = \delta_{jm} \delta_{kn} - \delta_{jn} \delta_{km}. \quad (\text{B..7})$$

As identidades conhecidas como Fórmulas de Lagrange são conseqüências diretas da identidade simplificada de épsilon:

$$\begin{aligned} \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) &= (\mathbf{a}^T \mathbf{c}) \mathbf{b} - (\mathbf{a}^T \mathbf{b}) \mathbf{c}, \\ (\mathbf{a} \times \mathbf{b}) \times \mathbf{c} &= (\mathbf{a}^T \mathbf{c}) \mathbf{b} - (\mathbf{b}^T \mathbf{c}) \mathbf{a}, \end{aligned} \quad (\text{B..8})$$

em que as expressões $\mathbf{a} \times (\mathbf{b} \times \mathbf{c})$ e $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c}$ são chamadas produtos triplos vetoriais.

Das Fórmulas de Lagrange extrai-se outra importante propriedade dos produtos vetoriais, as chamadas Identidades de Jacobi:

$$\begin{aligned} \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) + \mathbf{b} \times (\mathbf{c} \times \mathbf{a}) + \mathbf{c} \times (\mathbf{a} \times \mathbf{b}) &= \mathbf{0}, \\ (\mathbf{a} \times \mathbf{b}) \times \mathbf{c} + (\mathbf{b} \times \mathbf{c}) \times \mathbf{a} + (\mathbf{c} \times \mathbf{a}) \times \mathbf{b} &= \mathbf{0}. \end{aligned} \quad (\text{B..9})$$

A norma do produto vetorial é definida como

$$\|\mathbf{a} \times \mathbf{b}\| = \|\mathbf{a}\| \|\mathbf{b}\| \sin \theta, \quad (\text{B..10})$$

com θ igual ao ângulo formado pelos vetores \mathbf{a} e \mathbf{b} .

De B..10 tem-se outra identidade bastante útil:

$$\|\mathbf{a} \times \mathbf{b}\|^2 + \|\mathbf{a}^T \mathbf{b}\|^2 = \|\mathbf{a}\|^2 \|\mathbf{b}\|^2. \quad (\text{B..11})$$

O produto escalar triplo $\mathbf{a}^T (\mathbf{b} \times \mathbf{c})$ pode ser interpretado geometricamente como o volume do paralelepípedo definido pelos vetores \mathbf{a} , \mathbf{b} e \mathbf{c} (cf. Figura B..1(b)). Assim sendo as seguintes caracterizações são equivalentes:

$$\mathbf{a}^T (\mathbf{b} \times \mathbf{c}) = \mathbf{b}^T (\mathbf{c} \times \mathbf{a}) = \mathbf{c}^T (\mathbf{a} \times \mathbf{b}). \quad (\text{B..12})$$

Se \mathbf{a} , \mathbf{b} e \mathbf{c} forem coplanares, o volume do paralelepípedo é igual a zero e conseqüentemente o produto escalar triplo por eles definido será nulo.

Um produto vetorial entre dois vetores pode ser escrito em termos de multiplicação de matrizes como sendo o produto de uma matriz anti-simétrica e um vetor:

$$\mathbf{a} \times \mathbf{b} = As(\mathbf{a})\mathbf{b}, \quad (\text{B..13})$$

em que

$$As(\mathbf{a}) \triangleq \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}. \quad (\text{B..14})$$

C. DECOMPOSIÇÃO EM VALORES SINGULARES PARA RESOLUÇÃO DE EQUAÇÕES LINEARES HOMOGÊNEAS

A decomposição em valores singulares (*Singular Value Decomposition* ou simplesmente *SVD*) é uma importante fatoração de matrizes retangulares, com diversas aplicações em estatística e processamento de sinais. Dentre as várias aplicações para a decomposição *SVD*, tem-se o cálculo de matrizes pseudo-inversas, resolução de sistemas de equações sobre-determinados e minimização por mínimos quadrados. Esta seção apresenta sucintamente a *SVD* e sua aplicação na resolução de um problema de equações lineares homogêneas. Uma abordagem mais aprofundada do tema pode ser encontrada em [64], principal referência desta seção.

Dada uma matriz \mathbf{A} de dimensões $n \times m$, sua decomposição *SVD* é

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T, \quad (\text{C..1})$$

em que $\mathbf{U} \in \mathbb{R}^{n \times n}$ é uma matriz ortogonal e unitária, $\mathbf{D} \in \mathbb{R}^{n \times m}$ é uma matriz diagonal com valores reais não-negativos em ordem decrescente e $\mathbf{V} \in \mathbb{R}^{m \times m}$ é uma matriz ortogonal e unitária.

Considere o sistema de equações na forma

$$\mathbf{A}\mathbf{x} = \mathbf{0}, \quad (\text{C..2})$$

em que \mathbf{A} é uma matriz $n \times m$ com $n > m$, ou seja, há mais equações do que incógnitas.

Procura-se encontrar o vetor $\mathbf{x} \neq \mathbf{0}_{n \times 1}$ que satisfaça a todas as equações do sistema C..2. Observa-se que se \mathbf{x} é solução deste conjunto de equações, então $k\mathbf{x}$ também o é para qualquer k escalar. Assim sendo, uma restrição razoável é procurar uma solução tal que $\|\mathbf{x}\| = 1$.

Em geral, sistemas sobre-determinados não possuem uma solução exata, a não ser que $\text{rank}(\mathbf{A}) < m$, ou seja, a matriz \mathbf{A} não possua posto coluna completo. Na ausência de uma solução exata para o sistema, normalmente procura-se a solução de mínimos quadrados. O problema é então definido como encontrar \mathbf{x} que minimize a norma $\|\mathbf{A}\mathbf{x}\|$, tal que $\|\mathbf{x}\| = 1$, o que pode ser resolvido da seguinte maneira.

Seja a matriz \mathbf{A} decomposta em valores singulares dada por C..1. A norma do enunciado pode então ser reescrita como $\|\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{x}\|$. Entretanto, $\|\mathbf{U}\mathbf{D}\mathbf{V}^T\mathbf{x}\| = \|\mathbf{D}\mathbf{V}^T\mathbf{x}\|$ e $\|\mathbf{x}\| = \|\mathbf{V}^T\mathbf{x}\|$, pois \mathbf{U} e \mathbf{V} são ambas ortogonais de norma unitária. Assim sendo, procura-se minimizar $\|\mathbf{D}\mathbf{V}^T\mathbf{x}\|$ tal que $\|\mathbf{V}^T\mathbf{x}\| = 1$.

Reescrevendo $\mathbf{y} = \mathbf{V}^T\mathbf{x}$, tem-se um problema de minimização da norma $\|\mathbf{D}\mathbf{y}\|$ com a restrição $\|\mathbf{y}\| = 1$. Como \mathbf{D} é uma matriz diagonal de elementos em ordem decrescente, a solução para este problema é $\mathbf{y} = (0, 0, \dots, 0, 1)^T$, com apenas uma entrada diferente de zero na última posição. Finalmente, $\mathbf{x} = \mathbf{V}\mathbf{y}$, que é simplesmente a última coluna de \mathbf{V} .

D. CONFIGURAÇÕES DE TESTES UTILIZADOS NOS EXPERIMENTOS

A Figura D..1 apresenta as configurações iniciais e final utilizadas nas avaliações em simulação dos controladores propostos. A Tabela D..1 mostra as coordenadas e orientação do robô em cada postura de teste com relação ao centro da porta. Nas simulações, a porta possui 2 *m* de altura e 1,40 *m* de largura.

Tabela D..1: Coordenadas e orientação das posições de testes com relação ao centro da porta

Posição	x [<i>m</i>]	y [<i>m</i>]	θ [°]
1	-3	-5	120°
2	-3	-1	32°
3	0	-5	90°
4	0	-6	85°
Final	0	-1,5	90°

A Figura D..2 apresenta as configurações iniciais e final utilizadas nas avaliações experimentais dos controladores propostos. A Tabela D..2 mostra as coordenadas e orientação aproximadas do robô em cada postura de experimento com relação ao centro da porta. Os testes foram realizados utilizando uma porta de 2 *m* de altura e 90 *cm* de largura.

Tabela D..2: Coordenadas e orientação das posições dos experimentos com relação ao centro da porta

Posição	x [<i>m</i>]	y [<i>m</i>]	θ [°]
1	0	-1,8	90°
2	2,7	-2,1	135°
3	3,1	-1,7	150°
Final	0	-1,2	90°

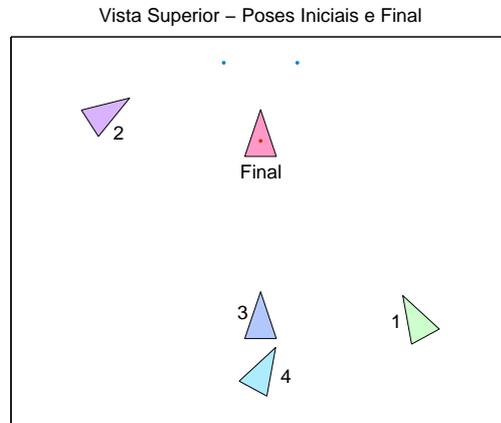


Figura D..1: Configurações iniciais e final para a postura do robô nas avaliações em simulação.

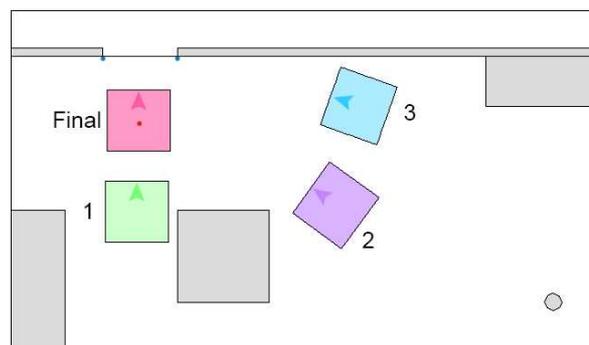


Figura D..2: Configurações iniciais e final para a postura do robô nas avaliações experimentais.