



## LOCOMOÇÃO SEMIAUTÔNOMA EM ESCADAS PARA ROBÔS MÓVEIS COM ESTEIRAS

Lucas Carvalho de Lima

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Fernando Cesar Lizarralde

Rio de Janeiro  
Março de 2016

LOCOMOÇÃO SEMIAUTÔNOMA EM ESCADAS PARA ROBÔS MÓVEIS  
COM ESTEIRAS

Lucas Carvalho de Lima

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

---

Prof. Fernando Cesar Lizarralde, D.Sc.

---

Prof. Luciano Cunha de Araújo Pimenta, D.Sc.

---

Prof. Alessandro Jacoud Peixoto, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2016

Lima, Lucas Carvalho de

Locomoção semiautônoma em escadas para robôs móveis com esteiras/Lucas Carvalho de Lima. – Rio de Janeiro: UFRJ/COPPE, 2016.

XXII, 164 p.: il.; 29, 7cm.

Orientador: Fernando Cesar Lizarralde

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2016.

Referências Bibliográficas: p. 155 – 159.

1. Robôs autônomos.      2. Locomoção em escadas.
3. Localização.          I. Lizarralde, Fernando Cesar.  
II. Universidade Federal do Rio de Janeiro, COPPE,  
Programa de Engenharia Elétrica. III. Título.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## LOCOMOÇÃO SEMIAUTÔNOMA EM ESCADAS PARA ROBÔS MÓVEIS COM ESTEIRAS

Lucas Carvalho de Lima

Março/2016

Orientador: Fernando Cesar Lizarralde

Programa: Engenharia Elétrica

A crescente utilização de robôs em aplicações de exploração e resgate exige a capacidade de superar diversos tipos de obstáculos como escombros, escadas, dentre outros. Os dados dos sensores do robô como imagens das câmeras, dados inerciais e odometria são enviados ao operador auxiliando-o durante a teleoperação. Entretanto, os dados sensoriais fornecem ao operador uma visão limitada do ambiente, incentivando a realização de tarefas autônomas pelos robôs. Portanto, esta dissertação apresenta uma metodologia para locomoção semiautônoma em escadas aplicada a robôs móveis com esteiras.

A metodologia é dividida em três etapas principais: detecção e modelagem de escadas, localização em ambiente conhecido e estratégia de controle.

A etapa de detecção e modelagem de escadas permite que os dados dimensionais da escada e sua localização sejam estimados. São apresentadas duas abordagens, sendo a primeira realizada a partir de *lasers scan* e a segunda fazendo uso de sensores 3D como o *Kinect*.

Para a etapa de localização, utilizou-se uma abordagem baseada em mapa conhecido. A posição do robô é estimada através do Filtro de Kalman Estendido (EKF) que realiza a fusão dos dados da odometria com as medidas de posição obtidas a partir de um *laser scan*.

Uma estratégia de controle cinemático é utilizada para realizar o deslocamento em escadas, sendo baseada em rastreamento de trajetórias.

É proposta ainda uma metodologia de reconfiguração de postura do robô para auxiliar a locomoção sobre a escada e suavizar a aterrissagem no piso superior.

A metodologia proposta para locomoção semiautônoma em escadas é avaliada a partir de simulações e através de experimentos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## SEMI-AUTONOMOUS STAIR CLIMBING FOR A TRACKED MOBILE ROBOT

Lucas Carvalho de Lima

March/2016

Advisor: Fernando Cesar Lizarralde

Department: Electrical Engineering

The robots increase in urban search and rescue applications is evident and requires the ability to overcome a range of obstacles as wreckage, stairs and others. Data as camera images, inertial measurements and odometry are acquired by robot sensors and sent to the operator, assisting him during teleoperation. However the mentioned data provide to operator a limited view of the environment, encouraging the implementation of semi-autonomous tasks for robots. Therefore this work presents a methodology for semi-autonomous stair climbing for tracked mobile robots.

The methodology is divided in three main phases: stairs detection and modelling, map-based localization and robot control.

Stair detection and modelling phase provides estimation of stair dimensions and localization. Two detection approaches are studied, the first is performed using laser scan and the second using 3D sensor as Kinect.

The localization approach is based on a known environment map. Robot pose is estimated through the Extended Kalman Filter (EKF) that fuses odometry data and position measures provided by laser scan.

A kinematic control is used to drive the robot during stair climbing. The control approach is based on path following.

It is proposed also a robot posture reconfiguration method to perform stair climbing and soft landing on the upper floor.

The approach for semi-autonomous stair climbing is evaluated through simulations and experiments.

# Sumário

<b>Lista de Figuras</b>	<b>ix</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Lista de Símbolos</b>	<b>xviii</b>
<b>Lista de Abreviaturas</b>	<b>xxii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	4
1.1.1 Desafios da locomoção em escadas . . . . .	5
1.2 Objetivo . . . . .	6
1.3 Revisão da Literatura . . . . .	8
1.4 Estrutura do Texto . . . . .	17
<b>2 Locomoção semiautônoma em escadas durante a subida</b>	<b>18</b>
2.1 Robôs móveis com esteiras do tipo EOD . . . . .	19
2.2 Classificação das escadas . . . . .	22
2.3 Descrição do ambiente de operação . . . . .	25
2.4 Conclusões . . . . .	26
<b>3 Métodos de detecção e modelagem de escadas</b>	<b>27</b>
3.1 Detecção de escadas utilizando sensores <i>laser</i> . . . . .	28
3.1.1 Sensor <i>laser scan</i> . . . . .	28
3.1.2 Descrição do método . . . . .	29
3.1.3 Simulações e Experimentos . . . . .	36
3.2 Detecção e modelagem de escadas utilizando imagem 3D densa . . . . .	40
3.2.1 Sensor 3D <i>Kinect</i> . . . . .	41
3.2.2 Descrição do método . . . . .	42
3.2.3 Simulações e experimentos . . . . .	48
3.2.4 Conclusões . . . . .	53

<b>4</b>	<b>Método de localização em mapa conhecido</b>	<b>55</b>
4.1	Metodologia aplicada . . . . .	57
4.2	Localização durante a aproximação até a escada . . . . .	58
4.2.1	Método de extração de linhas . . . . .	59
4.2.2	Matching de <i>landmarks</i> . . . . .	61
4.2.3	Localização utilizando o Filtro de Kalman Estendido (EKF) . . . . .	69
4.2.4	Modelo da odometria do robô móvel com esteiras . . . . .	73
4.2.5	Filtro EKF aplicado ao robô móvel com esteiras . . . . .	76
4.3	Localização durante a locomoção na escada . . . . .	79
4.4	Simulações e experimentos . . . . .	82
4.4.1	Resultados experimentais . . . . .	85
4.5	Conclusões . . . . .	90
<b>5</b>	<b>Metodologia para locomoção em escadas durante a subida</b>	<b>91</b>
5.1	Cinemática direta do robô móvel com esteiras do tipo EOD . . . . .	93
5.2	Transição do piso inferior para a escada . . . . .	94
5.2.1	Fases da transição . . . . .	99
5.3	Deslocamento na escada durante a subida . . . . .	102
5.4	Aterrissagem suave sobre o piso superior . . . . .	103
5.5	Simulações . . . . .	112
5.5.1	Simulação da primeira etapa: Transição do piso inferior para a escada . . . . .	113
5.5.2	Simulação da terceira etapa: Aterrissagem suave no piso superior	115
5.6	Conclusões . . . . .	116
<b>6</b>	<b>Controle para locomoção em escadas durante a subida</b>	<b>117</b>
6.1	Modelo cinemático do robô móvel . . . . .	118
6.2	Controle cinemático baseado em rastreamento de trajetória . . . . .	122
6.2.1	Controle para deslocamento em escadas durante a subida . . . . .	125
6.2.2	Controle para a aproximação até a escada . . . . .	130
6.3	Simulações . . . . .	132
6.3.1	Simulação: Controle para o deslocamento em escadas durante a subida . . . . .	133
6.3.2	Simulação: Controle para a aproximação até a escada . . . . .	137
6.4	Conclusões . . . . .	139
<b>7</b>	<b>Locomoção semiautônoma em escadas durante a descida</b>	<b>140</b>
7.1	Formulação do problema . . . . .	140
7.2	Solução proposta . . . . .	141
7.2.1	Detecção e modelagem da escada . . . . .	141

7.2.2	Localização baseada em mapa conhecido . . . . .	145
7.2.3	Metodologia para locomoção em escadas durante a descida . .	145
7.3	Conclusões . . . . .	149
<b>8</b>	<b>Conclusões</b>	<b>150</b>
8.1	Trabalhos Futuros . . . . .	153
	<b>Referências Bibliográficas</b>	<b>155</b>
<b>A</b>	<b>Cálculo da matriz de rotação ótima <math>R_l</math></b>	<b>160</b>
<b>B</b>	<b>Lemas e provas: Controle para locomoção em escadas</b>	<b>163</b>
B.1	Lema de Barbalat . . . . .	163
B.2	Prova que a derivada $d(v_c^3 l \frac{\sin(e_\phi)}{e_\phi})/dt$ é limitada . . . . .	163

# Lista de Figuras

1.1	Exemplos do desafio DRC, (a) apresenta o robô Metal Rebel realizando atuação em uma válvula e (b) apresenta o robô CHIMP deslocando-se sobre a escada. Fonte: <a href="http://www.theroboticschallenge.org/gallery-all">www.theroboticschallenge.org/gallery-all</a> . . . . .	2
1.2	Exemplos de robôs do tipo <i>two legged</i> usados em aplicações de exploração e resgate. (a) Fonte: <a href="http://www.bostondynamics.com/robot_Atlas.html">www.bostondynamics.com/robot_Atlas.html</a> , (b) Fonte: <a href="http://www.theroboticschallenge.org/finalist/team-valor">www.theroboticschallenge.org/finalist/team-valor</a> . . . . .	2
1.3	Exemplos de robôs do tipo <i>Explosive Ordnance Disposal - EOD</i> . . . . .	3
1.4	Exemplo de robô do tipo EOD utilizando pernas robóticas para superar obstáculos. Fonte: <a href="http://www.diginfo.tv/v/12-0186-r-en.php">www.diginfo.tv/v/12-0186-r-en.php</a> . . . . .	5
1.5	Em (a) é apresentado o robô do tipo <i>wheel legged</i> e (b) ilustra os três passos realizados pelo robô durante a locomoção sobre a escada. Fonte: Luo et al. (2013a) . . . . .	9
1.6	Exemplo do robô móvel com esteiras durante a locomoção sobre a escada. Fonte: Helmick et al. (2002) . . . . .	9
1.7	Robô móvel com esteiras e dois sensores <i>laser scan</i> , um posicionado horizontalmente e outro verticalmente. Fonte: Kalantari et al. (2009) . . . . .	10
1.8	A sequência ilustra a identificação da escada após realizar o processamento da imagem. A segunda figura apresenta a detecção das bordas e as últimas duas imagens apresentam a operação morfológica de dilatação e a identificação da escada através dos quadros vermelhos e a ausência nos quadros verdes. Fonte: Carbonara & Guaragnella (2014) . . . . .	11
1.9	(a) ilustra a varredura vertical do <i>laser</i> e a captura dos degraus da escada. Já (b) apresenta a varredura horizontal do ambiente e a captura das linhas para determinar a localização e orientação da escada. Fonte: Zhang et al. (2011) . . . . .	12
1.10	Tabela com tarefas realizadas na locomoção semiautônoma em escadas e os autores que desenvolveram trabalhos relacionados ao tema . . . . .	16

2.1	Exemplo da trajetória percorrida pelo robô do ponto inicial até o final da escada . . . . .	19
2.2	Plataformas robóticas em aplicações EOD . . . . .	20
2.3	Diagrama simplificado do sistema de controle embarcado de um robô móvel com esteiras. . . . .	21
2.4	Exemplo de robô móvel com esteiras considerado no problema de locomoção semiautônoma em escadas. Fonte: <a href="http://www2.warwick.ac.uk/fac/sci/eng/meng/wmr/projects/rescue/">www2.warwick.ac.uk/fac/sci/eng/meng/wmr/projects/rescue/</a> . . . . .	22
2.5	Diagrama com os ângulos das escadas e a respectiva nomenclatura associada. . . . .	23
2.6	Classificação das escadas quanto à geometria . . . . .	23
2.7	Modelo de escada sem espelho (à esquerda) e com espelho (à direita)	24
2.8	Planta do interior de um prédio, ilustrando o tipo de ambiente considerado no problema proposto. . . . .	25
3.1	Ilustrações representativas do sensor <i>laser scan</i> em funcionamento . .	28
3.2	Exemplo do feixe luminoso do <i>laser</i> com ângulo $\alpha_{Li}$ e distância $d_{Li}$ do sensor ao ponto de reflexão . . . . .	29
3.3	(a) apresenta as vistas lateral e superior dos <i>lasers</i> vertical e horizontal realizando o escaneamento do ambiente e (b) apresenta a vista em perspectiva . . . . .	30
3.4	Vista superior do posicionamento dos sistemas de coordenadas $O_{LH}, O_{LV}, O_R$ e $O_B$ em relação à escada . . . . .	30
3.5	Sensor <i>laser</i> realizando varredura vertical da escada. . . . .	31
3.6	Ilustração demonstrando casos com paredes alinhadas e não alinhadas com o primeiro degrau da escada . . . . .	34
3.7	Segmentações aplicadas no método de detecção de escada utilizando sensor <i>laser</i> . . . . .	35
3.8	Vista superior do posicionamento dos sistemas de coordenadas $O_{LH}, O_{LV}, O_R$ e $O_B$ em relação à escada . . . . .	37
3.9	Gráfico das coordenadas 2D dos pontos capturados a partir do <i>laser</i> vertical (sistema de coordenadas $O_{LV}$ ) . . . . .	38
3.10	Ilustração do problema de oclusão dos degraus da escada capturados pelo <i>laser</i> . . . . .	39
3.11	Gráfico das coordenadas 2D dos pontos capturados a partir do <i>laser</i> horizontal e linhas extraídas pelo método <i>Ransac</i> (sistema de coordenadas $O_R$ ) . . . . .	39
3.12	Gráfico das coordenadas 2D dos pontos capturados a partir do <i>laser</i> vertical (sistema de coordenadas $O_{LV}$ ) . . . . .	40

3.13	Câmeras e projetor que compõem o <i>Kinect</i> . . . . .	41
3.14	Exemplo do mapeamento 3D realizado pelo <i>Kinect</i> . . . . .	42
3.15	Descontinuidades observadas na imagem em profundidade ( <i>depth image</i> ) de uma escada . . . . .	43
3.16	As figuras demonstram a aplicação do método de detecção de escadas utilizando sensor 3D. Da esquerda para a direita, a primeira figura demonstra a <i>depth image</i> , a segunda demonstra a detecção de bordas através do método de <i>Canny</i> e a última apresenta em verde as linhas detectadas a partir da imagem das bordas. . . . .	43
3.17	Exemplo do plano <i>pl</i> ajustado aos pontos 3D das bordas dos degraus e a caixa delimitadora $B_l$ , ambos obtidos através do método de detecção e modelagem de escadas utilizando o <i>Kinect</i> . . . . .	46
3.18	As figuras ilustram o plano <i>pt</i> transversal ortogonal ao plano <i>pl</i> e os pontos 3D das bordas dos degraus projetados no plano <i>pt</i> . . . . .	47
3.19	Vista superior do posicionamento dos sistemas de coordenadas inercial $O_B$ e sistema de coordenadas do <i>Kinect</i> $O_K$ em relação à escada .	49
3.20	(a) apresenta a <i>depth image</i> da escada e (b) apresenta a imagem das bordas com as linhas da escada capturadas. Ambas as imagens foram capturadas no experimento com o <i>Kinect</i> posicionado a 3.285 m em profundidade (direção $-y_B$ ) e a 0.62 m lateralmente (direção $x_B$ ) medidos em relação ao ponto inferior esquerdo da escada. . . . .	49
3.21	Modelagem 3D da escada realizada experimentalmente com o <i>Kinect</i> posicionado a 3.285 m em profundidade (direção $-y_B$ ) e a 0.62 m lateralmente (direção $y_B$ ) medidos em relação ao ponto inferior esquerdo da escada. . . . .	50
3.22	(a) apresenta a <i>depth image</i> de escada e (b) apresenta a imagem das bordas com as linhas da escada capturadas. Ambas as imagens foram capturadas no experimento utilizando o <i>Kinect</i> defasado por $120^\circ$ em orientação, posicionado a 2.74 m em profundidade (direção $-y_B$ ) e a 2.27 m lateralmente (direção $x_B$ ) medidos em relação ao ponto inferior mais à esquerda da escada. . . . .	52
3.23	Modelagem 3D da escada realizada experimentalmente com o <i>Kinect</i> defasado por $120^\circ$ graus em orientação, posicionado a 2.74 m em profundidade (direção $-y_B$ ) e a 2.27 m lateralmente (direção $x_B$ ) medidos em relação ao ponto inferior mais à esquerda da escada. . . . .	52
4.1	Vista superior do robô posicionado no ambiente durante a etapa de localização para a aproximação até a escada . . . . .	58

4.2	Segmentações aplicadas no método de extração de linhas durante a localização . . . . .	60
4.3	Parâmetros da linha extraída através do método <i>Ransac</i> . . . . .	60
4.4	A primeira imagem demonstra o robô na posição inicial. Após o robô realizar uma translação e rotação, o <i>laser</i> passa a capturar pontos diferentes do ambiente como é demonstrado na segunda ilustração, já na terceira figura, observam-se os pontos capturados do <i>laser</i> , por fim a última ilustração demonstra a transformação dos dados do sistema de coordenadas do <i>laser</i> , após o deslocamento do robô, para o sistema de coordenadas do mesmo antes do deslocamento. . . . .	62
4.5	A figura ilustra como ocorre a associação entre as linhas extraídas e as linhas mapeadas através da relação de rotação entre seus vetores direcionais. . . . .	63
4.6	Exemplo geométrico do cálculo da distância entre linhas extraídas e as linhas associadas do mapa. . . . .	66
4.7	Exemplo geométrico da minimização da distância entre as linhas associadas para o caso em que existe mais de uma solução . . . . .	68
4.8	Diagrama do Filtro de Kalman . . . . .	72
4.9	Vista lateral (a) e superior (b) do robô realizando o deslocamento sobre a escada e a representação dos sistemas de coordenadas $O_S$ , $O_R$ e $O_{LH}$ . . . . .	80
4.10	Simulação do método de localização. O gráfico (a) apresenta as trajetórias real e estimada no plano xy e os gráficos em (b) apresentam as posições x e y reais e estimadas, para os parâmetros de simulação: $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.1^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.06$ m e $\sigma_{L\phi} = 1^\circ$ . . . . .	82
4.11	Simulação do método de localização. O gráfico (a) apresenta o erro de estimativa em x e y, já o gráfico (b) apresenta o erro de orientação, para os parâmetros de simulação $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.1^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.06$ m e $\sigma_{L\phi} = 1^\circ$ . . . . .	83
4.12	Simulação do método de localização. O gráfico (a) apresenta as trajetórias real e estimada no plano xy e os gráficos em (b) apresentam as posições x e y reais e estimadas, para os parâmetros de simulação: $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.1^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.06$ m, $\sigma_{L\phi} = 1^\circ$ , $i_l = i_r = [0.8, 1]$ . . . . .	84
4.13	Simulação do método de localização. O gráfico (a) apresenta o erro de estimação em x e y, já o gráfico (b) apresenta o erro de orientação, para os parâmetros de simulação: $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.1^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.06$ m, $\sigma_{L\phi} = 1^\circ$ , $i_l = i_r = [0.8, 1]$ . . . . .	85
4.14	Vistas em perspectiva (a) e inferior (b) do robô Roomba . . . . .	86

4.15	Experimento do método de localização utilizando robô móvel. O gráfico (a) apresenta as trajetórias estimada, odométrica e medidas (método de matching - <i>laser</i> ) no plano xy e o gráfico (b) apresenta uma ampliação das mesmas trajetórias, para os parâmetros: $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.5^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.05$ m, $\sigma_{L\phi} = 1^\circ$ . . . . .	86
4.16	Experimento do método de localização utilizando robô móvel. O gráfico (a) apresenta as trajetórias do robô em x e y, já o gráfico (b) apresenta os dados estimados, odométricos e medidos da orientação do robô, para os parâmetros: $\sigma_x = \sigma_y = 0.005$ m, $\sigma_\phi = 0.5^\circ$ , $\sigma_{Lx} = \sigma_{Ly} = 0.05$ m, $\sigma_{L\phi} = 1^\circ$ . . . . .	87
4.17	Vista lateral (a) e superior (b) do robô e sistemas de coordenadas $O_S$ , $O_R$ e $O_{LH}$ adotados na etapa de localização durante a locomoção sobre a escada . . . . .	88
4.18	Experimento do método de localização sobre a escada. Gráfico da estimativa da posição no plano $XY$ (sistema $O_S$ ), utilizando as posições de referência 0.2 m, 0.6 m e 1.0 m . . . . .	89
4.19	Experimento do método de localização sobre a escada. O gráfico (a) apresenta a estimativa da posição $x$ e o gráfico (b) a estimativa da orientação $\phi$ , utilizando as posições de referência 0.2 m, 0.6 m e 1.0 m . . . . .	89
5.1	Vistas superior e lateral do robô móvel com esteiras e quatro pernas articuladas . . . . .	92
5.2	Sistemas de coordenadas e dimensões utilizados na cinemática direta do robô do tipo EOD . . . . .	93
5.3	Representação dos ângulos de inclinação $\alpha_S$ e $\alpha_R$ da escada e robô, respectivamente. . . . .	95
5.4	Representação do posicionamento das pernas robóticas para iniciar o movimento sobre os primeiros degraus . . . . .	95
5.5	Representação da perna robótica em contato com o primeiro degrau da escada e os parâmetros necessários ao cálculo de $\theta_{2,min}$ . . . . .	96
5.6	Representação da perna robótica em contato com o obstáculo definido como um único degrau . . . . .	97
5.7	Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-A. . . . .	99
5.8	Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-B. . . . .	100
5.9	Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-C. . . . .	100

5.10	Diagrama de blocos do controle de orientação do robô em malha fechada, durante a transição do piso inferior para a escada. . . . .	101
5.11	Configuração de posicionamento das pernas robóticas durante o deslocamento sobre a escada. . . . .	103
5.12	Vista lateral e superior dos sistemas de coordenadas em cada um dos três corpos rígidos que compõem o robô. . . . .	104
5.13	Exemplo do robô próximo ao final da escada e indicação do ponto de contato entre o chassi e o último degrau . . . . .	107
5.14	Robô posicionado no ponto limite antes da mudança do plano da escada para o plano horizontal do piso superior . . . . .	108
5.15	Robô com pernas frontais em contato com pavimento superior . . . .	110
5.16	Simulação da primeira etapa da locomoção sobre a escada, fase S1-A.	113
5.17	Simulação da primeira etapa da locomoção sobre a escada, fase S1-B.	114
5.18	Simulação da primeira etapa da locomoção sobre a escada, fase S1-C.	114
5.19	Ângulo de inclinação do robô obtido durante a simulação da primeira etapa de transição do piso inferior para a escada. . . . .	114
5.20	Gráficos referentes a simulação da etapa de aterrissagem suave sobre o piso superior. . . . .	115
6.1	Vista lateral (a) e vista superior (b) representando a locomoção do robô sobre a escada modelada como um plano inclinado . . . . .	118
6.2	Exemplo do uniciclo em movimento . . . . .	119
6.3	Robô móvel com esteiras e direção diferencial . . . . .	121
6.4	Exemplo do caso geral para o rastreamento de trajetórias . . . . .	122
6.5	Vista superior de um robô do tipo EOD realizando rastreamento de trajetória retilíneas em escadas . . . . .	124
6.6	Diagrama de blocos da estratégia de controle aplicada durante a locomoção em escadas . . . . .	126
6.7	Exemplo do robô realizando o rastreamento de trajetória do ponto inicial $p_{d0}$ até o ponto final $p_{df}$ próximo à escada . . . . .	130
6.8	Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, utilizando os parâmetros $\mathbf{q}_0 = [1 \ 0 \ 70^\circ]$ , $\phi_d = 90^\circ$ , $p_{d0} = [0.5 \ 0] \text{ m}$ , $(i_r, i_l) \in [0.7 \ 0.95]$ , $k = 0.38$ e $k_2 = 3.34$ . Os gráficos (c) e (d) apresentam o erro de orientação $e_\phi$ e a distância $l$ entre robô e trajetória desejada para os mesmos parâmetros de simulação. . . . .	134

6.9	Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta a variação da velocidade linear $v_c$ pelo operador e o gráfico (b) apresenta o sinal de controle $\omega_c$ , obtidos utilizando os parâmetros $\mathbf{q}_0 = [1 \ 0 \ 70^\circ]$ , $\phi_d = 90^\circ$ , $p_{d0} = [0.5 \ 0]$ m, $(i_r, i_l) \in [0.7 \ 0.95]$ , $k = 0.38$ e $k_2 = 3.34$ . . . . .	135
6.10	Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, utilizando os parâmetros $\mathbf{q}_0 = [-0.4 \ 0 \ 90^\circ]$ , incertezas paramétricas $r_e = 0.1 + 0.005$ m e $d_R = 0.6 + 0.02$ m, $(i_r, i_l) \in [0.7 \ 0.95]$ , $k = 0.38$ e $k_2 = 3.34$ . Os gráficos (c) e (d) apresentam o erro de orientação $e_\phi$ e a distância $l$ entre robô e trajetória desejada para os mesmos parâmetros de simulação. . . . .	136
6.11	Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta a variação da velocidade linear $v_c$ pelo operador e o gráfico (b) apresenta o sinal de controle $\omega_c$ , obtidos utilizando os parâmetros $q = [-0.4 \ 0 \ 90^\circ]$ , $r_e = 0.1 + 0.005$ , $d_R = 0.6 + 0.02$ , $i_r, i_l \in [0.7 \ 0.95]$ , $k = 0.38$ e $k_2 = 3.34$ . . . . .	137
6.12	Simulação do controle: aproximação até a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, ambos obtidos utilizando os parâmetros $\mathbf{q}_0 = [3 \ 1 \ 60^\circ]$ , $p_{d0} = [2 \ 1]$ , $p_{df} = [1.1 \ 5.3]$ , $(i_r, i_l) \in [0.8 \ 1]$ , $k = 0.20$ e $k_2 = 3.24$ . Os gráficos (c) e (d) apresentam o erro de orientação $e_\phi$ e a distância $l$ entre robô e trajetória desejada para os mesmos parâmetros de simulação. . . . .	138
6.13	Simulação do controle: aproximação até a escada. O gráfico (a) apresenta a variação da velocidade linear $v_c$ e o gráfico (b) apresenta o sinal de controle $\omega_c$ , obtidos utilizando os parâmetros $\mathbf{q}_0 = [3 \ 1 \ 60^\circ]$ , $p_{d0} = [2 \ 1]$ , $p_{df} = [1.1 \ 5.3]$ , $i_r, i_l \in [0.8 \ 1]$ , $k = 0.20$ e $k_2 = 3.24$ . . . . .	139
7.1	Exemplo da varredura do <i>laser</i> vertical aplicado ao problema de descida da escada. . . . .	142
7.2	Pontos 2D da varredura do <i>laser</i> vertical representados no sistema de coordenadas $O_R$ do robô. . . . .	142
7.3	Captura das paredes laterais à escada utilizando o <i>laser</i> horizontal. . . . .	143
7.4	Exemplo do posicionamento do sensor <i>Kinect</i> para capturar a escada no problema de descida, e a representação dos sistemas de coordenadas adotados. . . . .	144
7.5	Exemplo do posicionamento das pernas robóticas durante a fase SD-A. . . . .	146
7.6	Exemplo da transição do piso superior para escada, fase SD-A. . . . .	147

7.7	Exemplo da transição do piso superior para escada, fase SD-B. . . . .	147
7.8	Exemplo da transição suave da escada para o piso inferior. . . . .	148

# Lista de Tabelas

3.1	Dados do primeiro experimento de detecção de escadas utilizando <i>lasers scan</i> . . . . .	38
3.2	Resultados experimentais para detecção e modelagem de escadas utilizando sensor 3D . . . . .	51
3.3	Estimativa da posição e orientação da escada através do método de detecção e modelagem de escadas utilizando sensor 3D . . . . .	51
3.4	Resultados do segundo experimento do método de detecção e modelagem de escadas utilizando sensor 3D . . . . .	53
3.5	Estimativa da posição e orientação da escada através do método de detecção e modelagem de escadas utilizando sensor 3D . . . . .	53
5.1	Ângulos das juntas utilizados em cada fase S1-A, S1-B e S1-C. . . . .	113

# Lista de Símbolos

$(p_{1s})_R$	Posição do primeiro degrau em relação ao sistema de coordenadas $O_R$ , p. 39
$(p_{2s})_R$	Posição do segundo degrau em relação ao sistema de coordenadas $O_R$ , p. 39
$(p_{LH})_R$	Posição do laser horizontal representado no sistema de coordenadas $O_R$ , p. 37
$(p_{LV})_R$	Posição do laser vertical representado no sistema de coordenadas $O_R$ , p. 37
$B_l$	Caixa delimitadora do modelo da escada, p. 45
$B_{l,cx}, B_{l,cy}, B_{l,cz}$	Coordenadas $x, y, z$ do centroide da caixa limitadora $B_l$ , p. 45
$E_T(T_l)$	Função custo do erro de translação entre as linhas, p. 66
$E_\phi(R_l)$	Função custo do erro de orientação entre as linhas, p. 64
$H_S, W_S, D_S$	Altura, Largura e Profundidade da escada, p. 37
$H_{B_l}, W_{B_l}, D_{B_l}$	Dimensões de altura, largura e profundidade da caixa delimitadora $B_l$ , respectivamente, p. 46
$H_\alpha$	Histograma dos ângulos das linhas em $L$ , p. 44
$H_{col}$	Histograma do número de linhas em cada coluna da imagem das bordas ( $Im_E$ ), p. 44
$Im_D$	Imagem em profundidade da escada obtida pelo sensor <i>Kinect</i> , p. 44
$Im_E$	Imagem das bordas detectadas através do método de <i>Canny</i> aplicado à imagem ( $Im_D$ ), p. 44
$L$	Conjunto de linhas detectadas da imagem das bordas $Im_E$ , p. 44

$L'$	Linhas extraídas do histograma ( $H_\alpha$ ) com maior frequência e direção horizontal, p. 44
$O_B$	Origem do sistema de coordenadas inercial da base, p. 37
$O_L$	Origem do sistema de coordenadas do laser, p. 28
$O_R$	Origem do sistema de coordenadas fixado ao robô, p. 37
$O_{LH}$	Origem do sistema de coordenadas do laser posicionado horizontalmente, p. 37
$O_{LV}$	Origem do sistema de coordenadas do laser posicionado verticalmente, p. 37
$P_{cloud}$	Conjunto de pontos 3D ( <i>point cloud</i> ) correspondente à imagem em profundidade $Im_D$ , p. 44
$P'_{cloud}$	Pontos 3D correspondentes às linhas em $L'$ no interior de $b_{rect}$ , p. 44
$R_l$	Matriz de rotação ótima que minimiza a função custo $E_\phi(R_l)$ , p. 64
$T_l$	Translação ótima que minimiza a função custo $E_T(T_l)$ , p. 66
$\Delta\theta_h$	Resolução do parâmetro de <i>Hough</i> $\theta_h$ em radianos, p. 50
$\Delta r_h$	Resolução do parâmetro de <i>Hough</i> $r_h$ em pixel, p. 50
$\alpha_{Li}$	Ângulo entre o feixe $i$ do <i>laser</i> e a direção $x$ do sistema de coordenadas do <i>laser</i> , p. 28
$\alpha_S$	Ângulo de inclinação da escada, p. 46
$\alpha_{li}$	Ângulo da reta ortogonal à linha $i$ com o eixo $\vec{x}$ do sistema de coordenadas do laser, p. 60
$\alpha_{temp}$	Ângulo diedral entre o plano $pl_{temp}$ e o plano do solo, p. 44
$\eta_{rot}$	Constante de penalização do erro de orientação entre as linhas, p. 62
$\eta_{trans}$	Constante de penalização do erro da distância entre as linhas, p. 66
$\phi$	Ângulo de orientação do robô em relação ao sistema de coordenadas $O_B$ , p. 55

$\phi_l$	Ângulo de rotação ótima correspondente à matriz $R_l$ , p. 65
$\rho_{li}$	Distância da origem do sistema de coordenadas até a linha $i$ , p. 60
$b_{rect}$	Caixa representada na imagem $Im_E$ que delimita o conjunto com maior número de linhas em $L'$ sobrepostas verticalmente na imagem, p. 44
$d_L$	Distância do ponto de reflexão do feixe de luz até o sensor <i>laser</i> , p. 28
$d_R$	Distância entre o centro das duas rodas ou esteiras, p. 75
$d_S$	Profundidade do degrau da escada, p. 37
$d_{lS}$	Profundidade entre segmentos de linha verticais consecutivos, p. 32
$e_{rij}$	Erro da distância e rotação entre linhas extraídas $i$ e linhas do mapa $j$ , p. 62
$e_{tij}$	Erro da distância e rotação entre linhas extraídas $i$ e linhas do mapa $j$ , p. 66
$h_S$	Altura do degrau da escada, p. 37
$h_{lS}$	Altura do segmento de linha atual do degrau, p. 32
$l_i$	Comprimento da linha $i$ , p. 60
$p_{ai}$	Coordenadas 2D do ponto inicial da linha extraída $i$ , p. 62
$p_{bi}$	Coordenadas 2D do ponto final da linha extraída $i$ , p. 62
$pl$	Plano da escada formado pelas linhas das bordas dos degraus, p. 46
$pl_{temp}$	Plano temporário ajustado ao conjunto de pontos $P'_{cloud}$ pelo método de mínimos quadrados, p. 44
$q_{aj}$	Coordenadas 2D do ponto inicial da linha do mapa $j$ , p. 62
$q_{bj}$	Coordenadas 2D do ponto final da linha do mapa $j$ , p. 62
$r_e$	Raio da roda ou o raio da engrenagem principal que traciona as esteiras do robô, p. 75

$th_h$	Número mínimo de pontos para detectar a linha, p. 50
$u_{li}$	Vetor unitário que representa a direção da linha $i$ , p. 60
$u_{li}^d$	Vetor da direção da linha extraída $i$ , p. 62
$u_{lj}^m$	Vetor da direção da linha do mapa $j$ , p. 62
$v_i^d$	Vetor ortogonal à direção da linha extraída $i$ , p. 66
$v_j^m$	Vetor ortogonal à direção da linha do mapa $j$ , p. 66
$x_R, y_R, z_R$	Eixos do sistema de coordenadas $O_R$ , p. 37
$(x_{Li}, y_{Li})$	Coordenadas do ponto de reflexão do feixe luminoso em relação ao sistema de coordenadas do <i>laser</i> , p. 29

# Lista de Abreviaturas

3D	três dimensões, p. 8
CAN	<i>Controller Area Network</i> , p. 21
DH	<i>Denavit-Hartenberg</i> , p. 93
DRC	<i>DARPA Robotics Challenge</i> , p. 1
EKF	<i>Extended Kalman Filter</i> , p. 13
EOD	<i>Explosive Ordnance Disposal</i> , p. 3
GPS	<i>Global Positioning System</i> , p. 20
IMU	<i>Inertial Measurement Unit</i> , p. 20
MCL	<i>Monte Carlo Localization</i> , p. 13
RGB-D	<i>Red, Green, Blue and Depth</i> , p. 11
RGB	<i>Red, Green and Blue</i> , p. 41
ROS	<i>Robot Operating System</i> , p. 36
SLAM	<i>Simultaneous Localization and Mapping</i> , p. 14
USAR	<i>Urban Search and Rescue</i> , p. 1

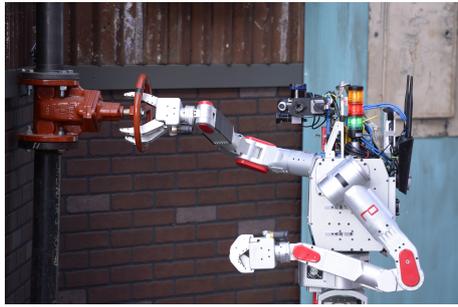
# Capítulo 1

## Introdução

São inúmeros os motivos pelos quais o ambiente pode se tornar hostil ao ser humano, seja pela presença de irradiação nuclear, destroços, risco de desmoronamentos ou possibilidade de explosões. Uma das causas mais comuns das alterações no ambiente mencionadas é a ocorrência de desastres naturais de grandes proporções. Um exemplo recente deste fato foi o terremoto seguido de tsunami e o conseqüente acidente nuclear que devastaram a cidade de Fukushima no Japão (2011), bem como o furacão Katrina que deixou um rastro de destruição no litoral sul dos Estados Unidos em 2005.

Em virtude da hostilidade dos ambientes após os acidentes, vem crescendo o interesse da comunidade científica em desenvolver e utilizar sistemas robóticos para exploração e reconhecimento em desastres, identificação e resgate de pessoas, além da atuação no ambiente. Os robôs utilizados neste tipo de aplicação são denominados *Urban Search and Rescue (USAR) robots* (Marques et al. 2006, Murphy 2012).

Uma das manifestações mais evidentes do interesse neste tipo de aplicação foi a criação da competição: *DARPA Robotics Challenge (DRC)*, que desafia equipes de robótica e seus robôs a realizarem ações relevantes em situações semelhantes a desastres, ações como: navegação autônoma, caminhar sobre pneus, chavear disjuntores, atuação em válvulas e locomoção em escadas. A figura 1.1 apresenta imagens de robôs realizando as tarefas mencionadas em uma das competições DRC.



(a)



(b)

Figura 1.1: Exemplos do desafio DRC, (a) apresenta o robô Metal Rebel realizando atuação em uma válvula e (b) apresenta o robô CHIMP deslocando-se sobre a escada. Fonte: [www.theroboticschallenge.org/gallery-all](http://www.theroboticschallenge.org/gallery-all)

A enorme variedade de obstáculos e terrenos que os robôs podem enfrentar durante a exploração de um ambiente urbano incentivou o estudo e desenvolvimento de sistemas robóticos com diversas formas de locomoção. Robôs bípedes (figura 1.2) são alvos de estudo neste sentido (Atmeh et al. 2014, Fallon et al. 2014), por sua complexidade em termos de controle e adaptabilidade aos diversos tipos de terrenos. Destacam-se ainda os robôs reconfiguráveis como é o caso do Robô CHIMP (Stentz et al. 2015) que se locomove sob duas pernas, entretanto para deslocamentos em terrenos acidentados e com obstáculos, o robô modifica sua configuração para quatro pernas sob esteiras.



(a)



(b)

Figura 1.2: Exemplos de robôs do tipo *two legged* usados em aplicações de exploração e resgate. (a) Fonte: [www.bostondynamics.com/robot\\_Atlas.html](http://www.bostondynamics.com/robot_Atlas.html), (b) Fonte: [www.theroboticschallenge.org/finalist/team-valor](http://www.theroboticschallenge.org/finalist/team-valor)

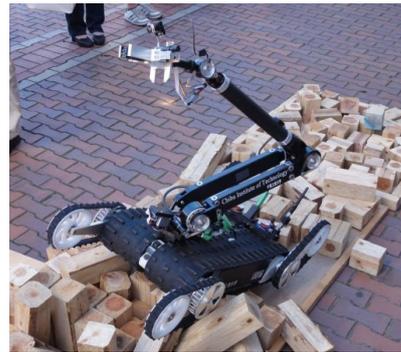
Devido a flexibilidade de deslocamento em ambientes urbanos, grande parte dos estudos são direcionados aos robôs móveis sob esteiras, como é apresentado em (Maggid et al. 2010, Marques et al. 2006). Alguns dos robôs com esteiras são amplamente

utilizados em aplicações militares, sendo denominados: *Bomb Disposal/Explosive Ordnance Disposal (EOD)*. Robôs cuja função consiste basicamente em explorar, identificar e neutralizar bombas e explosivos quando possível.

A maior parte dos robôs da classe *EOD* apresenta esteiras como principal meio de deslocamento, pernas articuladas frontais e traseiras que permitem transpor diversos tipos de obstáculos como escadas, para exploração de novos ambientes. Os robôs apresentam ainda, em sua parte superior, um manipulador robótico com graus de liberdade que permite manipular e carregar objetos ou até mesmo abrir uma porta. Exemplos de robôs *EOD* são apresentados na figura 1.3.



(a) Robô iniciando remoção de artefato explosivo. Fonte: [www.army-technology.com](http://www.army-technology.com)



(b) Robô se locomovendo sob terreno com obstáculos. Fonte: [thetechjournal.com/science/robots-help-search-for-survivors-in-japan.xhtml](http://thetechjournal.com/science/robots-help-search-for-survivors-in-japan.xhtml)

Figura 1.3: Exemplos de robôs do tipo *Explosive Ordnance Disposal - EOD*

Tanto para aplicações de exploração e resgate (*USAR - Urban Search and Rescue*) como em aplicações militares (*EOD*), um dos grandes desafios dos robôs é a superação de obstáculos como: escombros, entulhos, escadas, meios-fios, dentre outros. Neste sentido, diversos estudos e metodologias (Colas et al. 2013, Li et al. 2012, Wang et al. 2012) vêm sendo desenvolvidos e direcionados para superar escadas como obstáculos.

Não só para os robôs com esteiras como também para os robôs com rodas e pernas, a locomoção em escadas pode ser realizada tanto de maneira teleoperada, como de forma autônoma ou semiautônoma.

A teleoperação pode ser caracterizada pela operação remota de um sistema robótico. Desta maneira, a partir de um ambiente diferente do qual o robô está inserido, o operador envia comandos de controle para o sistema e recebe dados de sensores pertencentes ao robô como câmeras, *lasers*, odometria, dentre outros. Os dados sensoriais são a principal fonte de informações do ambiente e robô, sendo enviados ao operador para realizar as tarefas pretendidas durante a operação.

Já a operação autônoma é caracterizada pela realização de todas as tarefas automatizadas, diferenciando-se da semiautônoma que executa tanto ações autônomas

como ações oriundas do operador. Apesar do elevado grau de inteligência computacional que a operação autônoma ou semiautônoma carrega consigo, ela aumenta a segurança do sistema robótico, pois utiliza os diversos sensores e os algoritmos de controle para realizar as tarefas com maior grau de precisão, além de reduzir o tempo de operação e treinamento por parte do operador.

Dentre as tarefas necessárias para execução da locomoção sobre escadas em uma operação de exploração, destacam-se: localização no ambiente, planejamento de trajetória, identificação de escadas, controle, posicionamento e estabilidade dos movimentos durante o deslocamento.

Tendo em vista a variedade e complexidade das subtarefas necessárias à resolução do problema, esta dissertação é centrada na proposta de uma metodologia para locomoção semiautônoma em escadas, utilizando robôs móveis com esteiras da classe *EOD*.

A metodologia desenvolvida apresenta inicialmente uma solução para o problema de subida da escada, englobando principalmente as subtarefas de localização em ambiente conhecido, detecção e identificação de escadas, estratégia de controle e posicionamento para locomoção durante a subida, além da aterrissagem suave no piso superior. Após apresentar e avaliar as ferramentas para a locomoção durante a subida da escada, é discutido o problema de descida da mesma sendo proposta uma solução baseada nas ferramentas mencionadas.

## 1.1 Motivação

Durante a operação de robôs móveis em aplicações de busca e resgate em ambientes urbanos, é comum deparar-se com a presença de escadas como obstáculos, principalmente no interior de prédios, fábricas e casas. Desta maneira, a locomoção sobre escadas se torna um ponto crítico para a exploração de novos pavimentos e ambientes.

Ao longo dos anos, os robôs do tipo *EOD* se tornaram uma solução eficaz não só para o problema de locomoção em escadas, como também para superar diversos outros obstáculos que um ambiente urbano pode apresentar. Como já mencionado, os robôs *EOD* utilizam esteiras laterais para se deslocar, permitindo uma maior superfície de contato com o terreno e, conseqüentemente, maior estabilidade e tração durante os movimentos. Eles apresentam ainda pernas robóticas articuladas dianteiras e traseiras que os permitem reconfigurar sua postura para superar os diferentes tipos de obstruções no caminho (figura 1.4).



Figura 1.4: Exemplo de robô do tipo EOD utilizando pernas robóticas para superar obstáculos. Fonte: [www.diginfo.tv/v/12-0186-r-en.php](http://www.diginfo.tv/v/12-0186-r-en.php)

### 1.1.1 Desafios da locomoção em escadas

Apesar das características favoráveis dos robôs do tipo EOD para superar alguns tipos de obstáculos, realizar a locomoção em escadas através da teleoperação ainda é um grande desafio.

Para se deslocar em uma escada durante a subida, o robô realiza basicamente três etapas: a transição do piso inferior para a escada, a locomoção propriamente dita sobre a mesma e a aterrissagem no pavimento superior.

A transição do piso inferior para a escada é o primeiro desafio durante a teleoperação. Como as dimensões dos robôs do tipo EOD e dos degraus das escadas são próximas em geral, o operador deve alterar a configuração do robô, utilizando as pernas articuladas, para iniciar o deslocamento sobre os primeiros degraus.

O operador dispõe de uma visão limitada do ambiente através dos dados de sensores como câmeras, *lasers*, odometria, dentre outros, o que torna complexa a realização da primeira etapa, pois comandos incorretos e imprecisos podem gerar escorregamentos do robô e até mesmo tombamentos.

A segunda etapa, a locomoção propriamente dita sobre a escada, também é desafiadora. As variações de orientação do robô ao redor da direção de subida podem reduzir o atrito entre esteiras e degraus, ocasionando escorregamentos entre os mesmos e, possivelmente, o tombamento do robô.

Os desvios de orientação podem resultar ainda em colisões com balaustres e paredes laterais à escada, causando danos ao sistema robótico ou até mesmo o fracasso da missão. Portanto, o operador deve controlar precisamente as velocidades das esteiras para manter a orientação adequada do robô e conduzi-lo até o final da escada, utilizando apenas os dados sensoriais como informação do ambiente.

O último desafio durante a subida da escada é a sub tarefa de aterrissagem sobre o piso superior. Na transição da escada para o pavimento superior, caso o operador não altere a orientação do robô através das pernas articuladas, o robô perde a estabilidade proporcionada pelos degraus e realiza uma queda brusca sobre o piso

superior. Assim, ainda que dispondo de uma visão limitada do ambiente através dos sensores, os comandos do operador devem ser precisos para evitar escorregamentos e choques entre robô e o pavimento.

A descida da escada também é desafiadora quando realizada através da teleoperação. A tarefa principal de descida pode ser também dividida em três subtarefas: transição do piso superior para a escada, deslocamento na escada durante a descida e transição da escada para o piso inferior.

O ponto mais crítico durante a descida da escada é a primeira subtarefa, a transição do piso superior para a escada. Nesta etapa, comandos incorretos do operador podem causar uma queda brusca do robô sobre a escada e o consequente tombamento do mesmo. Já as outras subtarefas executadas na descida apresentam desafios semelhantes às subtarefas realizadas durante a subida da escada.

Como pôde ser observado, a locomoção sobre a escada por meio da teleoperação é uma tarefa complexa, pois exige precisão, habilidade e destreza do operador que dispõe de uma visão do ambiente restrita aos dados sensoriais do robô.

Portanto, as dificuldades inerentes à teleoperação dos robôs durante a locomoção em escadas, por exemplo, tornam evidente e necessária a automação de tarefas desempenhadas pelos robôs, para aumentar a segurança e precisão durante o deslocamento e reduzir o tempo necessário para execução das ações.

## 1.2 Objetivo

O objetivo desta dissertação é propor uma metodologia para locomoção semi-autônoma em escadas aplicada a robôs móveis com esteiras.

Inicialmente é formulado o problema de locomoção em escadas durante a subida, sendo dividido em quatro tarefas principais: detecção e modelagem de escadas; localização em ambiente conhecido; estratégia de controle; configurações de posicionamento das pernas robóticas e aterrissagem suave no piso superior.

Após formular o problema de locomoção durante a subida, além de apresentar e analisar as ferramentas para sua solução, é discutida a solução para o problema de locomoção em escadas durante a descida, sendo baseada nas ferramentas avaliadas.

A etapa de detecção e modelagem de escadas para o problema de subida foi baseada em duas abordagens distintas. A primeira foi proposta em (Zhang et al. 2011), que utiliza dois sensores *laser scan* para realizar a detecção e localização da escada. Um dos sensores é posicionado para realizar escaneamentos verticais, de modo que o algoritmo permite identificar a geometria característica e repetitiva dos degraus da escada, utilizando as coordenadas 2D do sensor. Desta maneira, além de detectar a escada, é possível estimar a altura e profundidade dos degraus. O segundo *laser scan* é posicionado para realizar escaneamentos horizontais do ambiente, de

modo que o algoritmo realiza a extração de linhas a partir dos dados do sensor, permitindo localizar o primeiro degrau da escada e estimar sua orientação.

A segunda metodologia para detecção e modelagem de escadas é baseada no estudo desenvolvido em (Delmerico et al. 2013). Através de sensores 3D como o *Kinect*, as descontinuidades características dos degraus da escada são capturadas na *depth image* (imagem em profundidade) e processadas, obtendo-se as linhas dos degraus e os correspondentes pontos em 3D. Em seguida, um modelo de escadas é ajustado aos dados obtidos, modelo que consiste em um plano inclinado limitado por um prisma. Desta maneira, o método permite estimar a inclinação da escada, altura e profundidade dos degraus, além da posição e orientação da escada.

Para a etapa de localização no ambiente, utilizou-se uma abordagem probabilística baseada em mapa conhecido a priori. A posição e orientação do robô são estimadas através do Filtro de Kalman Estendido (EKF) que realiza a fusão dos dados da odometria do robô com as medidas de posição obtidas através do sensor *laser scan*.

A posição do robô medida através do *laser* é obtida aplicando-se a metodologia de associação de dados proposta neste trabalho. Através do método *Ransac* (Fischler & Bolles 1981), as linhas do ambiente são extraídas. Logo em seguida, as linhas são comparadas ao mapa através do método de associação de dados, resultando na medida de rotação e translação realizadas pelo robô.

Durante a locomoção sobre a escada, foi considerado que as esteiras são tracionadas em contato com as bordas dos degraus modelados como um plano inclinado, de forma que as forças normais de contato se mantêm uniformes (Freitas 2008) e as forças de atrito apresentam magnitude suficiente para impulsionar o robô. Desta maneira, é possível desconsiderar efeitos de escorregamento entre esteiras e degraus, para pequenos desvios de orientação do robô em relação à direção de subida da escada.

Ao considerar o movimento do robô em baixas velocidades em conjunto com a hipótese da escada representada por um plano com forças de contato uniformes, foi possível desprezar os efeitos da dinâmica do sistema e assim propor uma solução cinemática para o controle.

O principal objetivo da estratégia de controle cinemático é permitir que o robô suba a escada, mantendo sua posição equidistante em relação às laterais da mesma e com a orientação na direção de subida. Desta maneira, evita-se uma possível colisão com paredes ou balaustres da escada, além de reduzir a possibilidade de tombamentos causados pela variação da orientação do robô.

A estratégia de controle utilizada é baseada em rastreamento de trajetórias. É definida uma trajetória em linha reta equidistante das laterais da escada e na direção de subida da mesma, de modo que o controle permite que o robô siga a

trajetória desejada, minimizando os erros de posição e orientação. Ressalta-se que o controle proposto permite que o operador altere a velocidade linear do robô durante a locomoção sobre a escada.

Para executar a locomoção sobre a escada durante a subida, é proposto ainda um conjunto de configurações de posicionamento das pernas robóticas com as seguintes funcionalidades: permitem a transição do solo para a escada, aumentam a superfície de contato das esteiras com os degraus e, por fim, realizam a aterrissagem suave no piso superior.

Por fim, as ferramentas utilizadas na solução do problema de subida da escada são adaptadas e discutidas para solucionar o problema de locomoção durante a descida da mesma.

As diversas etapas que envolvem a metodologia para locomoção semiautônoma sobre escadas são simuladas, sendo algumas delas validadas também através de experimentos.

### 1.3 Revisão da Literatura

As soluções propostas para o problema de locomoção semiautônoma em escadas não se limitam apenas aos robôs com esteiras. Em (Luo et al. 2014) é proposto um planejamento e controle autônomo para locomoção sobre escadas, utilizando um robô bípede. No estudo em questão, o sistema de percepção do robô identifica e modela a escada, em seguida os dados do modelo da escada e posicionamento do robô são entradas para um algoritmo de planejamento de trajetória. Por fim, uma estratégia de controle é responsável por executar o movimento do robô de acordo com a trajetória desejada.

Em (Luo et al. 2013a) um robô do tipo *wheel legged* (figura 1.5) é desenvolvido para aplicações no interior de ambientes urbanos. O robô consiste em dois conjuntos de três rodas que além de girar em torno de seus próprios eixos também executam rotação em torno do eixo transversal do robô, permitindo que seja executada a tarefa de locomoção em escadas. Uma estratégia de controle é desenvolvida baseada na estrutura do robô.

O sistema utiliza o sensor 3D conhecido como *Kinect* para capturar os dados da escada, processá-los e através de algoritmos específicos obter os dados de posição e dimensão dos degraus. As informações obtidas são utilizadas na estratégia de controle para definir a sequência de movimentos do robô.

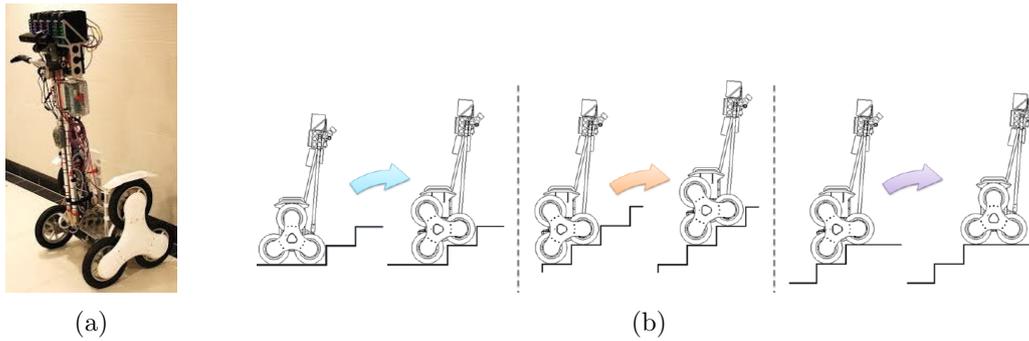


Figura 1.5: Em (a) é apresentado o robô do tipo *wheel legged* e (b) ilustra os três passos realizados pelo robô durante a locomoção sobre a escada. Fonte: Luo et al. (2013a)

Diversos trabalhos foram desenvolvidos no sentido de solução do problema para os robôs móveis com esteiras.

Em (Helmick et al. 2002) é proposta uma metodologia para estimação da posição e orientação do robô, além de controle para aumentar a velocidade e efetividade do deslocamento sobre as escadas. Uma câmera é utilizada para detecção das bordas dos degraus e um *laser scan* é responsável por detectar as paredes e balaustres laterais. Contudo, a utilização de câmeras monoculares para detecção pode tornar o método sensível às condições de luminosidade do ambiente. O filtro de Kalman é utilizado para realizar a fusão dos dados do *laser*, câmera e de sensores inerciais, de maneira a obter a melhor estimativa da posição em relação às laterais da escada e orientação do robô. Dois controles independentes são propostos e implementados, um controle dinâmico para minimizar o erro orientação e outro para minimizar o erro de posição.



Figura 1.6: Exemplo do robô móvel com esteiras durante a locomoção sobre a escada. Fonte: Helmick et al. (2002)

Já em (Zhang et al. 2011) a detecção e localização de escadas é realizada utilizando dois sensores *laser scan*. Um dos *lasers* é posicionado verticalmente para detecção, e para localização são efetuados escaneamentos horizontais de um segundo *laser*. Durante a locomoção sobre a escada, um giroscópio é responsável por obter os dados de orientação do robô e, através de um controle baseado em lógica *fuzzy*, o erro de orientação é minimizado. O final da escada é detectado através da fusão de dados de sensor sonar e infravermelho. Uma sequência de aterrissagem é pro-

posta para reduzir o impacto do robô com o solo após a saída da escada. Deve ser ressaltado que todas as etapas são realizadas de forma autônoma.

Em (Kalantari et al. 2009) é proposto algoritmo autônomo para locomoção sobre escadas baseado em controle cinemático. Novamente são utilizados dois sensores *laser*, um vertical e outro horizontal (figura 1.7). O *laser* vertical é responsável por identificar o final da escada e o *laser* horizontal detecta as paredes laterais, fornecendo dados para o cálculo da orientação e posição do robô em relação às laterais da escada. Um controle cinemático baseado em modelo de referência é responsável por manter o robô com a posição e orientação desejadas. Durante o deslocamento sobre a escada, o autor propõe ainda três configurações diferentes para as pernas traseiras e dianteiras do robô, sendo uma para a etapa de saída do solo para a escada, outra para deslocamento sobre a escada e a última configuração para suavizar a aterrissagem no piso superior.

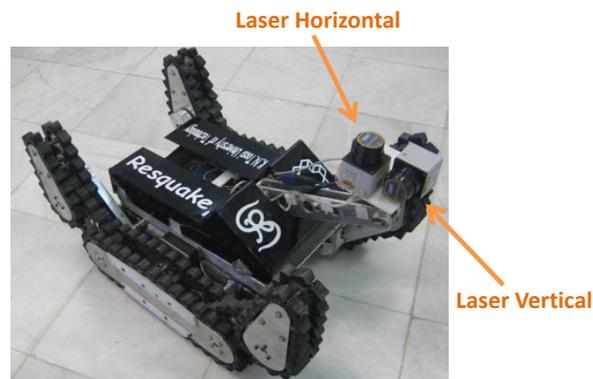


Figura 1.7: Robô móvel com esteiras e dois sensores *laser scan*, um posicionado horizontalmente e outro verticalmente. Fonte: Kalantari et al. (2009)

Como pôde ser observado, a locomoção semiautônoma ou autônoma sobre escadas engloba uma série de tarefas que vão desde a detecção, identificação e modelagem da escada, até o posicionamento dos braços robóticos e aterrissagem durante a saída para o piso superior. Entretanto, os estudos mencionados não contemplam todas as etapas necessárias em virtude do foco que cada autor pretende ressaltar em seus estudos.

Desta maneira, a seguir são mencionados os trabalhos relacionados a cada etapa aplicada nesta dissertação para a solução do problema, de modo que as referidas etapas são: detecção, identificação e modelagem de escadas, localização no ambiente e métodos de controle para a locomoção sobre escadas.

## Detecção e modelagem de escadas

A etapa de percepção da escada consiste em uma tarefa chave para locomoção semi-autônoma, pois é necessário que o robô detecte a escada e determine a sua orientação

e localização para deslocar-se até seu objetivo. Ressalta-se ainda que a modelagem da escada é de suma importância, tanto para determinar se o robô será capaz de transpor o obstáculo, quanto para fornecer dados geométricos, objetivando o planejamento do controle e posicionamento do sistema robótico.

Os principais métodos de percepção utilizam três tipos de sensores: câmeras monoculares, sensores *laser scan* e sensores 3D como as câmeras RGB-D ou o conhecido sensor *Kinect* desenvolvido pela empresa *Microsoft*.

Em (Carbonara & Guaragnella 2014) um algoritmo de detecção de escadas utilizando câmera monocular é proposto. A partir da imagem em escala de cinza, o método detecta as bordas da imagem através de processamento e captura as linhas correspondentes às bordas dos degraus da escada.

Entretanto, os algoritmos de detecção baseados em processamento de imagem de câmeras monoculares são sensíveis às variações de luminosidade, textura e cor do ambiente. A figura 1.8 ilustra o método em questão.

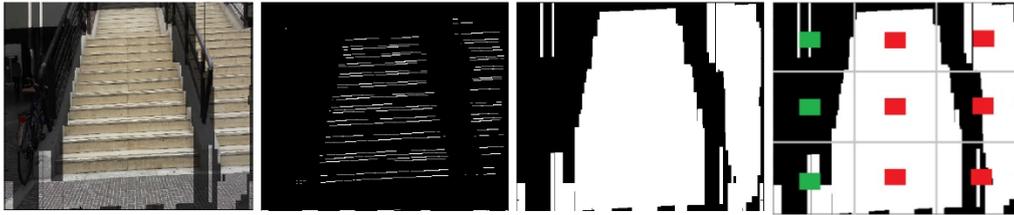
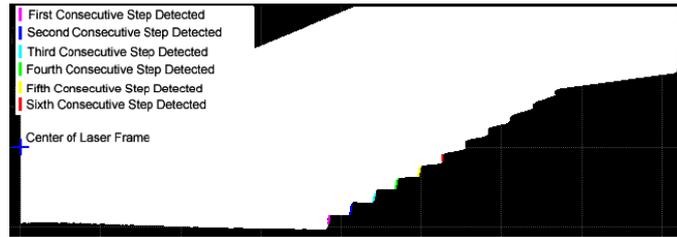


Figura 1.8: A sequência ilustra a identificação da escada após realizar o processamento da imagem. A segunda figura apresenta a detecção das bordas e as últimas duas imagens apresentam a operação morfológica de dilatação e a identificação da escada através dos quadros vermelhos e a ausência nos quadros verdes. Fonte: Carbonara & Guaragnella (2014)

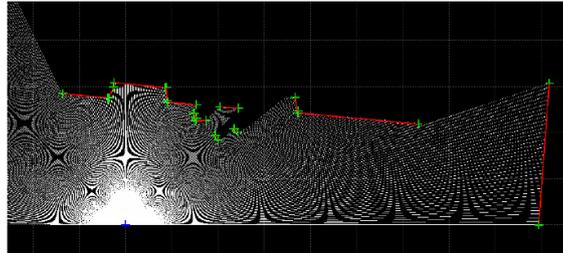
Uma das alternativas à utilização das câmeras monoculares é a adoção de *lasers scan*, como ocorre em Mihankhah et al. (2009), Zhang et al. (2011).

Ambos os autores propõem métodos para detecção de escadas utilizando dois *lasers*, um vertical e outro horizontal. Ao realizar os escaneamentos verticais com o sensor, os algoritmos identificam a geometria padronizada e repetitiva característica dos degraus, seja pela relação entre pontos subsequentes dos dados ou pela extração de linhas dos mesmos.

Após a identificação da escada, ambos os algoritmos utilizam o *laser* horizontal para extrair a linha correspondente ao primeiro degrau da escada detectada, determinando assim seu posicionamento e orientação.



(a)



(b)

Figura 1.9: (a) ilustra a varredura vertical do *laser* e a captura dos degraus da escada. Já (b) apresenta a varredura horizontal do ambiente e a captura das linhas para determinar a localização e orientação da escada. Fonte: Zhang et al. (2011)

Diversos estudos foram direcionados à utilização de sensores 3D como câmeras RGB-D e *Kinect* para detecção de escadas. Em (Luo et al. 2013b), um método para detecção e modelagem de escadas é desenvolvido, baseando-se em conceitos de análise estatística de dados em *point cloud* 3D e modelos geométricos ajustados.

Utilizando os métodos estatísticos, os planos existentes no ambiente são identificados e os pontos correspondentes aos planos são agrupados em *clusters* (grupos de pontos). Em seguida, a relação geométrica definida pelos espaçamentos iguais entre os *clusters* é utilizada para identificar os planos pertencentes aos degraus. Através da *depth image* (imagem em profundidade), os pontos das bordas de cada degrau são obtidos. Por fim, os dados obtidos são ajustados ao modelo para escadas proposto pelo autor, permitindo determinar o número de degraus, altura, largura e profundidade dos mesmos, além da orientação da escada.

O método em questão é capaz de identificar diferentes tipos de escadas com bons resultados, todavia não pode ser aplicado aos robôs móveis com esteiras, pois é necessário que o sensor 3D seja posicionado há uma altura mínima do solo para detectar os planos horizontais do ambiente, fato que não pode ser satisfeito por esta classe de robôs por sua altura reduzida. Ressalta-se ainda que este método só permite identificar e modelar as escadas localizadas a uma distância próxima do robô.

Já em (Delmerico et al. 2013) é desenvolvida metodologia para detecção e modelagem de escadas que utiliza sensor 3D, explorando as discontinuidades dos degraus evidenciadas na *depth image*. O algoritmo em questão captura a *depth image* e, ao aplicar o detector de bordas de *Canny* (Canny 1986), evidencia todas as bordas

presentes na imagem.

Os degraus da escada originam descontinuidades na *depth image* pelas abruptas mudanças de profundidade em relação ao sensor. Desta maneira, ao aplicar o detector de bordas na *depth image*, um conjunto de linhas paralelas correspondentes aos degraus é evidenciado. Em seguida, aplica-se a transformada de *Hough* (Kiryati et al. 1991) para obter as retas da imagem. Utilizando técnicas de filtragem e agrupamento, as retas correspondentes às bordas dos degraus são capturadas. Os respectivos pontos em 3D das linhas extraídas são armazenados e após sucessivas capturas da imagem e processamento, forma-se um grupo de pontos necessário para ajustar o modelo aos dados.

O modelo da escada proposto consiste em um plano inclinado limitado por um prisma, de maneira que após o ajuste dos pontos 3D é possível estimar a inclinação da escada, a altura e profundidade dos degraus, além da posição e orientação da escada.

### **Localização no ambiente**

O problema de localização consiste basicamente em estimar a posição do robô em relação a um sistema de coordenadas fixo. Em virtude das condições ambientais consideradas e dos sensores disponíveis para as aplicações robóticas, as abordagens utilizadas para localização são amplas e diversificadas.

A grande maioria dos estudos utiliza os dados da odometria em conjunto com as informações obtidas pela identificação de *landmarks* do ambiente, para estimar a posição do robô (Corke 2011). Os *landmarks* são pontos de referência com posição conhecida em relação a um sistema de coordenadas inercial, de maneira que eles podem ser identificados pelos mais variados tipos de sensores como: *lasers scan*, sensores 3D (RGB-D), sonares, câmeras monoculares, dentre outros.

Dois dos principais métodos utilizados para realizar a estimativa da posição são: Filtro de Kalman Estendido (*Extended Kalman Filter - EKF*) (Thrun et al. 2005) e *Monte Carlo Localization (MCL)* (Dellaert et al. 1999). O *Monte Carlo Localization* também é chamado de maneira mais genérica como filtro de partículas.

Em (Armesto & Tornero 2006) é proposto método para obter a estimativa da localização de robôs de resgate, utilizando *laser scan* e mapeamentos geométricos. Através dos métodos apresentados para ajustes de reta e ajuste de círculos aos dados do *laser*, os grupos de dados são mapeados em linhas e pontos (*landmarks*), de modo que as linhas representam as paredes, e os pontos representam as colunas e quinas do ambiente.

Em seguida, o autor propõe dois tipos de medidas de posição obtidas através dos dados geométricos, uma comparando as colunas e quinas com o mapa predefinido, e outra comparando apenas as linhas com o mapa. Desta maneira, a metodologia

estima a medida de rotação e translação realizadas pelo robô, utilizando apenas o *laser*.

Por fim, são apresentadas duas abordagens diferentes, Filtro de Kalman Estendido (EKF) e *Monte Carlo Localization (MCL)*, para realizar a fusão dos dados da odometria com a medida de posição obtida do *laser*, resultando na estimativa da posição do robô.

Para ambientes onde não há o conhecimento prévio do mapa, utilizam-se as abordagens conhecidas como *Simultaneous Localization and Mapping (SLAM)*. A abordagem SLAM consiste em estimar a localização do robô a partir dos *landmarks* observados do ambiente, enquanto realiza o mapeamento. De modo semelhante às abordagens de localização em mapas conhecidos, os métodos SLAM utilizam principalmente o EKF e o MCL para estimar a posição do robô.

Em Lionis & Kyriakopoulos (2002) é proposta uma abordagem SLAM, utilizando sensor *laser* 2D para identificação dos *landmarks*. O estudo em questão utiliza segmentos de linha para representar os *landmarks*, que são atualizados a cada instante com os novos pontos medidos do *laser*. A atualização das linhas é realizada através do algoritmo *Weighted Least Square Fitting*. Para estimar a posição do robô a partir dos dados dos *landmarks* e odometria, o autor utiliza dois filtros de Kalman. O primeiro deles estima as orientações do robô e segmentos de linha extraídos, orientações estas transmitidas ao segundo filtro de Kalman para obter as posições de ambos.

## **Controle para locomoção em escadas durante a subida**

A definição da estratégia de controle é extremamente importante para que o robô possa realizar a locomoção sobre a escada durante a subida de maneira segura e eficaz.

Em virtude da geometria dos robôs com esteiras do tipo *EOD* e das características do contato entre o robô e a escada, a grande maioria das abordagens de controle são focadas em manter o robô com uma orientação constante durante o deslocamento, bem como reduzir ao máximo os desvios de posição em relação à linha equidistante das laterais da escada.

Os elevados desvios de orientação podem acarretar em um aumento do escorregamento do robô e possivelmente tombamento, já os desvios de posição podem ocasionar colisões do robô com as laterais da escada.

Os principais estudos dividem-se em estratégias de controle cinemático e controle dinâmico.

Em Helmick et al. (2002), Mourikis et al. (2007), estratégias de controle dinâmico são propostas para solução do problema de locomoção autônoma sobre escadas. Em ambos os trabalhos, a orientação do robô é estimada através da fusão de informações de *laser* ou câmera monocular e giroscópio, utilizando o EKF.

Os referidos autores propõem controles separados, sendo um controle para minimizar o erro de posição e outro para minimizar o erro de orientação. No instante em que o robô se desvia do centro da escada, o controle de posição produz um sinal de referência de orientação que difere da orientação desejada original, fazendo com que o robô seja direcionado para o centro da escada. Ao aproximar-se do centro da escada, a nova referência gradativamente retorna ao valor original de orientação desejada.

Em sua grande maioria, as estratégias de controle cinemático utilizam o modelo do robô móvel com direção diferencial para representar a cinemática dos robôs com esteiras do tipo *EOD*.

Em (Kalantari et al. 2009), como mencionado anteriormente, é proposto controle cinemático para minimizar os erros de orientação e posição do robô em relação às laterais da escada. O controle é baseado em modelo de referência que produz a trajetória desejada a partir das velocidades angular e linear desejadas. O erro de posição e orientação em relação à trajetória desejada é definido e sua dinâmica é obtida. A partir da linearização da dinâmica do erro, o autor obtém a lei de controle e estabiliza o sistema em torno da origem.

O controle produz bons resultados, entretanto em virtude da linearização da dinâmica do erro em torno do ponto de equilíbrio, não se pode inferir a estabilidade assintótica global do sistema em malha fechada.

Existem estudos mais recentes que sugerem a utilização de controles cinemáticos baseados em lógica *Fuzzy* para executar a locomoção autônoma sobre escadas.

Na abordagem de controle proposta em Zhang et al. (2011), um algoritmo em lógica *Fuzzy* altera a velocidade linear do robô de acordo com o erro de orientação do mesmo, objetivando a redução dos efeitos de escorregamento. Desta maneira, com o aumento do erro de orientação durante o deslocamento sobre escada, o algoritmo *Fuzzy* reduz a velocidade linear com o intuito de concentrar o movimento na velocidade angular e assim restabelecer a orientação desejada. O erro de orientação é minimizado a partir de um controle proporcional, de modo que a medida de orientação é obtida através de um giroscópio apenas.

Já em Adiwahono et al. (2014), o controle cinemático *Fuzzy* aplicado a um robô móvel com esteiras é utilizado para minimizar os erros de orientação e posição em relação à escada.

Para ilustrar comparativamente as características desta dissertação em relação aos estudos mencionados, a tabela da figura 1.10 classifica os trabalhos no que diz respeito à detecção e modelagem de escadas, localização no ambiente, estratégia de controle e configuração da postura do robô.

		Helmick et al. (2002)	Zhang et al. (2011)	Kalantari et al. (2009)	Mourikis et al. (2007)	Carbonara et. al (2014)	Luo et al. (2013b)	Adiwahono et al. (2014)
Detecção e Modelagem da escada	Laser scan		X					X
	Câmera RGB-D						X	
	Câmera Monocular	X			X	X		X
Localização no ambiente								
Localização na escada		X		X	X			X
Controle	Cinemático			X				
	Dinâmico	X						
	Lógica Fuzzy		X		X			X
Reconfiguração da postura	X	X	X					

Figura 1.10: Tabela com tarefas realizadas na locomoção semiautônoma em escadas e os autores que desenvolveram trabalhos relacionados ao tema

## 1.4 Estrutura do Texto

Esta dissertação é dividida em oito capítulos.

No capítulo 2 é formulado o problema de locomoção semiautônoma em escadas, considerando inicialmente a subida, realizada por um robô móvel com esteiras do tipo *EOD*. Neste capítulo são definidas as considerações sobre o ambiente, a escada, o robô e a interação entre estes elementos.

Os dois métodos de detecção e modelagem de escadas avaliados para a solução do problema são descritos no capítulo 3. Os conceitos e algoritmos aplicados em cada método são detalhados e os resultados experimentais apresentados.

No capítulo 4 é apresentado o método de localização utilizado para estimar a posição do robô, durante o deslocamento do ponto inicial até a posição a frente da escada. São apresentados os conceitos de localização baseada em mapa, a localização utilizando o Filtro de Kalman Estendido (EKF) e os algoritmos para extração e associação dos pontos de referência. O método para localização durante o deslocamento sobre a escada também é apresentado. As simulações e resultados experimentais são descritos ao final do capítulo.

A metodologia de reconfiguração das pernas robóticas para executar a locomoção sobre a escada durante a subida é apresentada no capítulo 5. Para cada etapa definida: transição do solo para escada, locomoção sobre a escada e aterrissagem suave no piso superior, é descrita a respectiva configuração das pernas robóticas a ser utilizada. Ao final do capítulo são apresentadas as simulações da metodologia desenvolvida.

O capítulo 6 apresenta a estratégia de controle cinemático utilizada para conduzir o robô no deslocamento durante a subida da escada. Inicialmente é estabelecido o modelo cinemático do robô móvel com esteiras e direção diferencial, em seguida é apresentado o controle cinemático baseado em rastreamento de trajetórias. A estabilidade do sistema em malha fechada é analisada através da teoria de Lyapunov e validada a partir de simulações.

No capítulo 7 é formulado o problema de locomoção em escadas durante a descida, utilizando um robô móvel com esteiras e pernas articuladas. Desta maneira, é proposta uma metodologia para a descida da escada utilizando as ferramentas para a locomoção durante a subida apresentadas nos capítulos anteriores.

O capítulo 8 apresenta as conclusões sobre o estudo desenvolvido e os trabalhos futuros.

## Capítulo 2

# Locomoção semiautônoma em escadas durante a subida

O problema de locomoção semiautônoma em escadas é extremamente amplo e complexo do ponto de vista das variantes que podem ser inseridas no robô, ambiente e escada propriamente dita. Grande parte dos estudos (Helmick et al. 2002, Kalandari et al. 2009, Zhang et al. 2011) considera apenas a detecção da escada e a locomoção sobre a mesma como tarefas integrantes do problema, não contemplando a localização no ambiente.

Neste trabalho, o problema de locomoção semiautônoma em escadas durante a subida consiste primeiramente em utilizar os sensores de um robô com esteiras do tipo *EOD* para detectar e modelar a escada, a partir de uma determinada posição. Em seguida, o robô deve ser capaz de localizar-se no ambiente e realizar o deslocamento até a posição mais próxima possível à frente da escada. Após, deve ser executada a estratégia de controle para a locomoção propriamente dita sobre a escada durante a subida, realizando por fim uma aterrissagem suave no piso superior. Ressalta-se que as ações mencionadas são realizadas sem a intervenção do operador, que deve ter a possibilidade apenas de alterar a velocidade linear do robô, enquanto executa o deslocamento sobre a escada.

Além da locomoção sobre a escada durante a subida, existe ainda o problema de descida da escada. Desta maneira, a formulação do problema de descida e a solução proposta são apresentadas no capítulo 7.

Um exemplo da trajetória percorrida pelo robô, durante o deslocamento da posição inicial até o piso superior à escada, pode ser visualizado na figura 2.1.

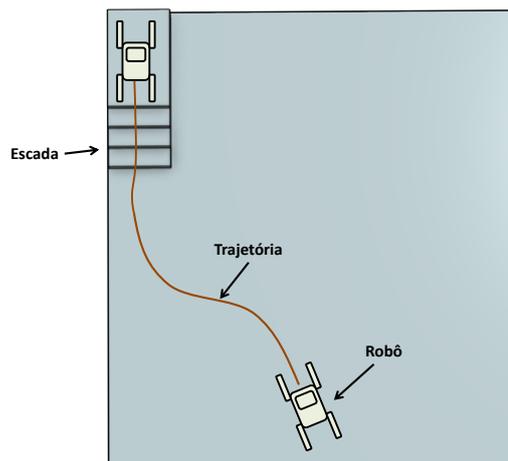


Figura 2.1: Exemplo da trajetória percorrida pelo robô do ponto inicial até o final da escada

Deste modo, o problema locomoção semiautônoma em escadas durante a subida pode ser dividido nas subtarefas listadas a seguir:

- Detecção e modelagem da escada;
- Localização no ambiente e aproximação;
- Estratégia de controle para locomoção sobre a escada; e
- Aterrissagem suave no piso superior.

O estudo desenvolvido neste trabalho foi baseado nos robôs móveis com esteiras do tipo EOD. Desta maneira, a seção 2.1 descreve as características gerais dos robôs móveis com esteiras, incluindo os sensores utilizados durante os experimentos.

Em virtude da variedade de modelos, geometria e tipos de escadas existentes, a seção 2.2 descreve as hipóteses relativas ao formato da escada adotada no problema e a sua interação com robô. Já a seção 2.3 descreve as hipóteses relativas ao ambiente onde são realizados o deslocamento e a localização do robô.

## 2.1 Robôs móveis com esteiras do tipo EOD

Os robôs móveis com esteiras do tipo EOD são comumente utilizados em aplicações militares, entretanto por suas características e peculiaridades, este tipo de sistema também é utilizado em aplicações de resgate e exploração (USAR - Urban Search and Rescue).

O objetivo dos robôs móveis do tipo EOD é deslocar-se em diferentes tipos de terrenos, executar exploração em locais de múltiplos pavimentos, realizar busca e remoção de artefatos explosivos, ou até mesmo desarme de bombas se possível. Uma das habilidades presentes neste tipo de robô que o auxilia na execução das tarefas mencionadas é a capacidade superar diferentes tipos de obstáculos como escadas.

Para deslocar-se no ambiente, o robô dispõe de duas esteiras que podem ser compostas por materiais em metal e borracha, o que atribui resistência estrutural e amplia a tração com o terreno. Os movimentos lineares e angulares do robô são realizados por meio da rotação diferencial das esteiras.

Os robôs do tipo EOD apresentam usualmente quatro pernas articuladas por juntas de revolução, duas traseiras e duas dianteiras, que o auxiliam a superar diferentes obstáculos. As pernas robóticas podem mover-se independentemente ou podem apresentar movimento simultâneo dentro de cada par e independente entre os pares.

É comum os robôs desta classe apresentarem ainda um manipulador com três juntas de revolução e um efetuador na extremidade, permitindo manipular cargas como um artefato explosivo, por exemplo. A figura 2.2 apresenta robôs EOD em operação.



(a) Robô deslocando-se em escadas



(b) Transporte de artefato explosivo realizado pelo robô

Figura 2.2: Plataformas robóticas em aplicações EOD

Para navegação no ambiente, o robô pode dispor de diversos tipos de sensores. Os principais utilizados são as câmeras monoculares e panorâmicas, os *lasers scan*, sensores 3D como o *Kinect* e sensores inerciais como IMU (*Inertial Measurement Unit*), além de sensores de posicionamento como GPS (*Global Positioning System*).

Em linhas gerais, as câmeras fornecem a visão do ambiente durante a operação do robô, de modo que as panorâmicas podem apresentar visão de 180° a 360° em alta definição, sendo posicionadas geralmente na parte superior do chassi.

Os sensores *laser scan* apresentam diversas funcionalidades em aplicações EOD, dentre as quais podemos destacar: localização no ambiente, identificação de obstáculos e pontos de referência, além de mapeamento. Para realizar as funcionalidades mencionadas, os *lasers* são frequentemente posicionados horizontalmente, podendo em alguns casos escanear verticalmente para identificação de objetos ou mapeamento 3D. Durante a locomoção em terrenos irregulares e com obstáculos, os sensores inerciais como a IMU auxiliam a operação, fornecendo informações das acelerações e orientação do robô.

Os sensores 3D como o *Kinect* e câmeras RGB-D apresentam a capacidade de fornecer imagens em três dimensões do ambiente, sendo desta maneira largamente utilizados em tarefas de mapeamento, modelagem do ambiente e detecção de objetos.

Os atuadores do robô são compostos principalmente por motores elétricos responsáveis pelo movimento das esteiras, pernas articuladas e manipulador.

A operação do robô é realizada através de uma estação de trabalho que possui computador, *joystick* e sistema de comunicação *wireless* entre estação e o robô, sendo limitada pela potência do sinal que as antenas são capazes de transmitir e receber. A comunicação pode ainda ser realizada por cabo, limitando o raio de ação do robô pelo risco de travamentos entre os cabos e os obstáculos do ambiente.

O sistema de controle embarcado dos robôs é executado através de microcontroladores responsáveis por realizar o processamento de sinais digitais e analógicos, além de enviar comandos para os *drivers* que controlam os atuadores. A comunicação entre drivers de potência dos motores e o processador embarcado é frequentemente realizada através de redes de comunicação como a rede CAN (*Controller Area Network*). Desta maneira, é possível controlar os motores das esteiras, braços e juntas do manipulador através de uma mesma rede CAN que é largamente utilizada em aplicações industriais por sua robustez e confiabilidade.

O diagrama esquemático do controle embarcado do robô é representado na figura 2.3.

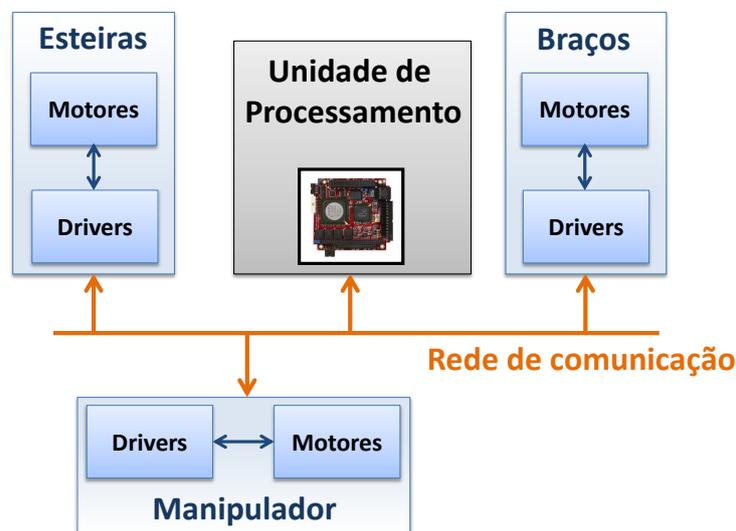


Figura 2.3: Diagrama simplificado do sistema de controle embarcado de um robô móvel com esteiras.

Portanto, neste trabalho foram considerados os robôs móveis com esteiras do tipo EOD, de modo que apresentam quatro pernas articuladas, sendo duas dianteiras e duas traseiras. Foi considerado ainda que o robô dispõe de dois sensores *laser scan* em sua plataforma, dispostos nas orientações vertical e horizontal. Cada orientação

permite diferentes formas de percepção do ambiente, de modo que os *lasers* são utilizados tanto para detecção e localização de escadas, quanto para localização do robô no ambiente. Considerou-se ainda a disponibilidade de um sensor 3D como o *Kinect*, a fim de capturar imagens do ambiente e permitir o estudo de uma das técnicas de detecção e modelagem de escadas descritas no texto. Por fim, utilizou-se ainda os dados de sensores inerciais como a IMU.

A figura 2.4 ilustra um exemplo de robô com esteiras e pernas articuladas considerado no problema.

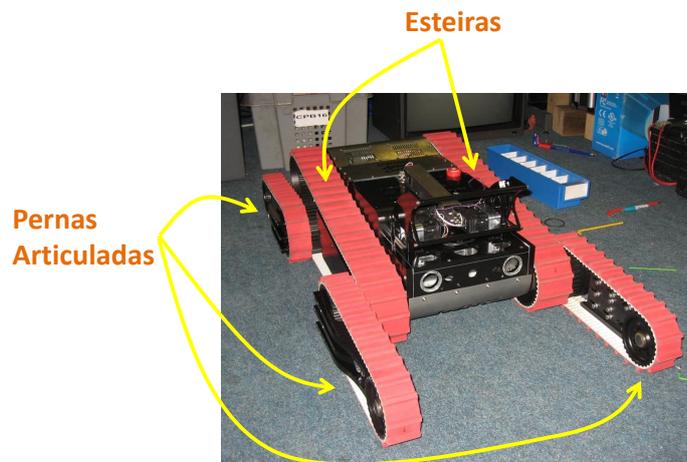


Figura 2.4: Exemplo de robô móvel com esteiras considerado no problema de locomoção semi-autônoma em escadas. Fonte: [www2.warwick.ac.uk/fac/sci/eng/meng/wmr/projects/rescue/](http://www2.warwick.ac.uk/fac/sci/eng/meng/wmr/projects/rescue/)

## 2.2 Classificação das escadas

A circulação vertical de pessoas entre pavimentos consecutivos com diferentes níveis é realizada através de elementos como escadas, rampas e elevadores. As escadas são extremamente comuns em ambientes urbanos, estando presentes em prédios, casas, lojas, fábricas, dentre outros.

A escolha do tipo de escada a ser utilizada para superar o desnível em questão, está diretamente ligada ao ângulo entre os níveis dos pisos que serão conectados. Desta maneira, as escadas são classificadas, em uma primeira instância, através da sua inclinação entre os pavimentos. A figura 2.5 ilustra os ângulos de inclinação e os respectivos modelos de escadas utilizados.

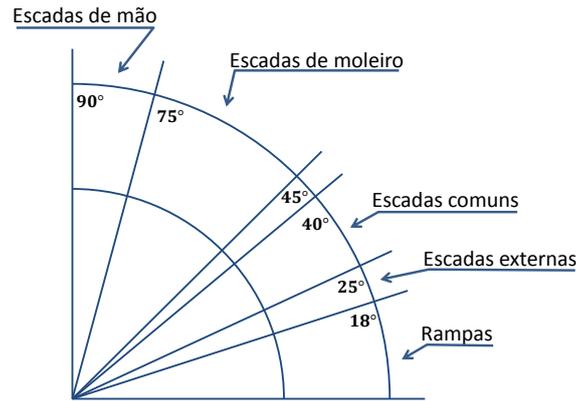


Figura 2.5: Diagrama com os ângulos das escadas e a respectiva nomenclatura associada.

Com base nas inclinações, este trabalho considera na definição do problema as escadas comuns e escadas externas, ou seja, são aquelas que variam em um ângulo entre  $18^\circ$  e  $40^\circ$ . Para ângulos reduzidos, são utilizadas rampas, que não configuram obstáculos desafiadores aos robôs móveis com esteiras, já ângulos acima de  $40^\circ$  podem tornar extremamente complexo o deslocamento sobre a escada, em virtude do escorregamento entre esteiras e degraus.

As escadas podem receber uma segunda classificação quanto ao seu formato geométrico. Desta maneira, de acordo com o traçado definido para a circulação de pessoas entre pavimentos, as escadas são classificadas como: escada reta, escada em “L”, escada em “U”, escada em curva e escada em caracol. A figura 2.6 ilustra cada modelo de escada mencionado.

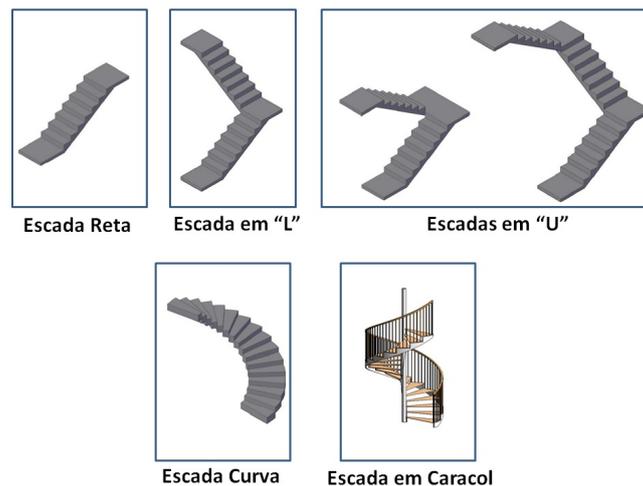


Figura 2.6: Classificação das escadas quanto à geometria

As escadas em “L” e em “U” apresentam patamares (níveis) intermediários entre o pavimento inferior e o pavimento superior, de modo que os conjuntos de degraus entre dois patamares é o que se denomina lance. Tendo em vista a classificação da

escada quanto à geometria, o problema proposto nesta dissertação considera somente as escadas retas com apenas um lance, ou seja, não existem níveis intermediários entre o piso inferior e o superior.

Os elementos que constituem a escada possuem uma nomenclatura oriunda da construção civil. As partes frontal e superior dos degraus são chamadas de espelho e base, respectivamente. A figura 2.7 apresenta dois modelos de escadas, o primeiro modelo sem espelhos nos degraus e o segundo modelo com espelhos.

Com a finalidade de simplificar o problema proposto, neste trabalho considerou-se apenas as escadas com espelhos, pois este modelo auxilia a tarefa de detecção e identificação da geometria da escada.

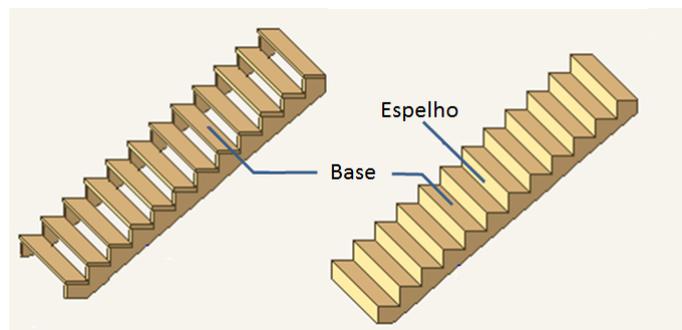


Figura 2.7: Modelo de escada sem espelho (à esquerda) e com espelho (à direita)

Ressalta-se ainda que a escada deve ser limitada lateralmente por uma ou duas paredes. Esta consideração é necessária para que o robô possa estimar sua posição e orientação em relação às laterais da escada, durante o deslocamento sobre a mesma.

A localização inicial da escada é desconhecida, desta maneira o robô deve detectá-la para então iniciar a aproximação. Desta maneira, a orientação inicial do robô está direcionada para a escada, de modo que foi desconsiderada do problema a tarefa de busca da escada no ambiente.

A modelagem da interação entre o robô e a escada é uma tarefa complexa, tendo em vista as considerações dinâmicas necessárias ao modelo. Desta maneira, uma simplificação consiste em considerar a escada como um plano inclinado nos pontos de contato entre as esteiras e os degraus. Já as forças de interação normais no ponto de contato podem ser consideradas uniformes, de modo que o atrito entre as esteiras e escada apresentam magnitude suficiente para impulsionar o robô sem que ocorram escorregamentos.

A modelagem do contato entre esteiras e escada como em um plano inclinado e as forças de interação uniformes são aproximações semelhantes àsquelas utilizadas em Freitas (2008) para o problema de reconfiguração de robôs móveis em terrenos irregulares.

Com base nas aproximações descritas, ao desprezar o escorregamento entre esteiras e escadas e considerar o deslocamento do robô em velocidades baixas, os efeitos

dinâmicos podem ser desprezados, permitindo realizar uma modelagem puramente cinemática para descrever o deslocamento do robô sobre a escada.

## 2.3 Descrição do ambiente de operação

Após descrever o robô e a escada a ser superada no problema proposto, é necessário definir o ambiente no qual estes elementos estão inseridos.

Existem diversos tipos de ambientes e terrenos por onde robôs do tipo EOD podem realizar missões de busca e exploração. Ao definir as escadas como obstáculo integrante do problema estudado, as possibilidades relacionadas ao tipo de ambiente tornam-se mais restritas a um grupo específico de locais.

Como já mencionado, as escadas são utilizadas usualmente na construção civil para circulação vertical de pessoas entre pavimentos de níveis diferentes. A estrutura vertical dos prédios com inúmeros pavimentos tornou ubíqua a utilização de escadas neste tipo de construção. Deve ser ressaltado que a adoção de escadas para circulação vertical em prédios também apresenta razões de segurança, visto que as mesmas representam o principal meio para fuga em casos de emergência.

Devido a larga utilização de escadas em ambientes urbanos, mais precisamente em prédios, o problema proposto neste trabalho considera o ambiente no qual a escada e o robô estão inseridos nos mesmos moldes do interior de um prédio. A característica marcante das construções prediais é a presença de estruturas ortogonais e planas (pisos, paredes e portas), além de corredores para circulação de pessoas. A figura 2.8 ilustra a planta de um ambiente característico do interior de prédios.

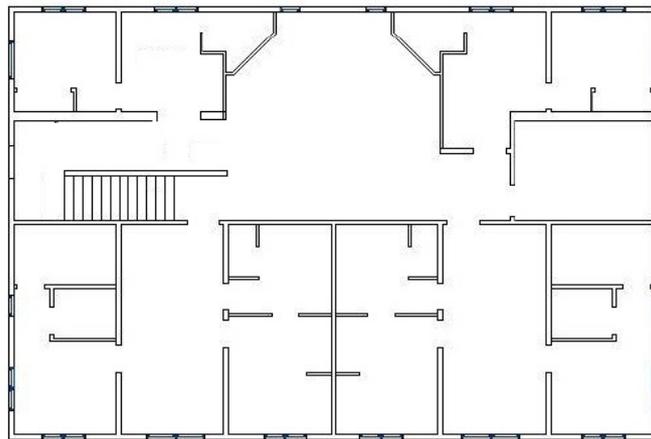


Figura 2.8: Planta do interior de um prédio, ilustrando o tipo de ambiente considerado no problema proposto.

O ambiente considerado nesta dissertação apresenta um mapa conhecido a priori, de modo que também são conhecidas a posição e orientação iniciais do robô. Para este problema foi considerado ainda que a trajetória da posição inicial até o ponto

próximo à escada é livre de obstáculos. Entretanto, existem estudos (Brooks et al. 2009) baseados em planejamento de trajetórias e desvio de obstáculos que podem ser utilizados caso a hipótese considerada seja relaxada.

Ressalta-se ainda que o movimento do robô, antes da locomoção na escada, se restringe ao plano 2D do solo, não havendo rampas ou inclinações no trajeto até a posição mais próxima da escada.

## 2.4 Conclusões

Portanto, neste capítulo foi definido o problema de locomoção semiautônoma em escadas durante a subida, sendo dividido em quatro etapas:

- Detecção e modelagem da escada;
- Localização no ambiente e aproximação;
- Estratégia de controle para locomoção sobre a escada; e
- Aterrissagem suave no piso superior.

As hipóteses relacionadas ao ambiente, o robô e a escada considerados no problema foram apresentadas, de modo que as características de cada um desses elementos foram descritas, bem como as relações de interação entre eles.

## Capítulo 3

# Métodos de detecção e modelagem de escadas

Como definido no capítulo 2, a localização da escada é desconhecida inicialmente, tornando necessário definir uma metodologia para detecção e modelagem da mesma. Atualmente, são utilizados sensores como câmeras monoculares, *lasers scan* e sensores 3D como o *Kinect* ou câmeras RGB-D, para realizar tal tarefa.

Os métodos baseados em câmeras monoculares realizam, em sua maioria, o processamento de imagem para detectar as linhas das bordas da escada, como ocorre em Carbonara & Guaragnella (2014) e Helmick et al. (2002). Entretanto, estes métodos tendem a ser mais suscetíveis às condições de luminosidade do ambiente pelas características intrínsecas das câmeras.

Uma das alternativas adotadas para reduzir a sensibilidade à luminosidade do ambiente é a utilização dos *lasers scan* e sensores 3D. Os métodos baseados nos sensores mencionados identificam a geometria repetitiva dos degraus da escada e a partir das coordenadas 2D ou 3D capturadas, são obtidos o modelo da escada e sua localização, além da sua detecção.

Deste modo, em Zhang et al. (2011) é apresentado um método para detecção e localização de escadas através de *lasers* posicionados horizontalmente e verticalmente, assim como em Delmerico et al. (2013) e Luo et al. (2013b) são apresentados métodos para detecção e modelagem de escadas utilizando sensores 3D.

Neste trabalho, duas metodologias são estudadas para a solução do problema de detecção e localização das escadas:

- Detecção de escadas utilizando sensores *laser*; e
- Detecção e modelagem de escadas utilizando imagem 3D densa.

### 3.1 Detecção de escadas utilizando sensores *laser*

Esta abordagem foi baseada no trabalho desenvolvido em Zhang et al. (2011), que também apresenta uma metodologia para solução do problema de locomoção sobre escadas.

O método em questão utiliza basicamente dois sensores *laser*, um posicionado verticalmente e outro posicionado horizontalmente, ambos na parte superior e à frente do robô.

Para auxiliar a compreensão do método, as próximas subseções descrevem o funcionamento do sensor e o método propriamente dito.

#### 3.1.1 Sensor *laser scan*

Os sensores *laser scan* são instrumentos largamente utilizados na indústria e na robótica em diversas aplicações como: navegação, posicionamento, medidas em superfície de materiais, dimensionamento, dentre outras.

A descrição rudimentar para o princípio de funcionamento do sensor consiste basicamente na emissão de feixes de luz e na medida do tempo em que o feixe incide no obstáculo, é refletido e coletado pelas lentes do sensor. A partir do tempo medido, é possível obter a distância percorrida pelo feixe e, conseqüentemente, a distância do sensor ao obstáculo.

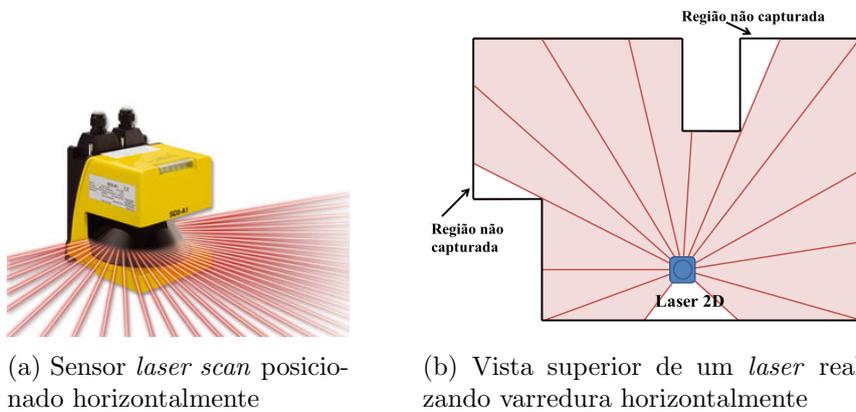


Figura 3.1: Ilustrações representativas do sensor *laser scan* em funcionamento

Para obter o conjunto de medidas em duas dimensões como ocorre neste tipo de sensor, os feixes de luz são emitidos com diversos ângulos em um mesmo plano de escaneamento, como por exemplo o plano horizontal do ambiente. Deste modo, ao final de uma varredura, para cada feixe de luz  $i$  é obtido um ângulo  $\alpha_{Li}$  e uma distância  $d_{Li}$  do sensor ao ponto de reflexão do feixe. A figura 3.2 ilustra o feixe de luz emitido pelo *laser* e os parâmetros  $d_{Li}$  e  $\alpha_{li}$  mencionados.

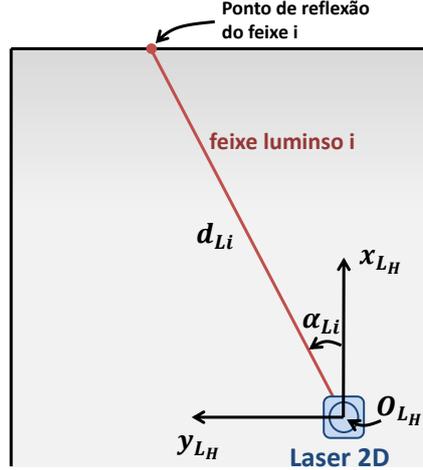


Figura 3.2: Exemplo do feixe luminoso do *laser* com ângulo  $\alpha_{Li}$  e distância  $d_{Li}$  do sensor ao ponto de reflexão

As coordenadas cartesianas  $(x_{Li}, y_{Li})$  de cada feixe de luz  $i$  em relação ao sistema de coordenadas  $O_{LH}$  do *laser* posicionado horizontalmente, são obtidas da seguinte maneira:

$$\begin{bmatrix} x_{Li} \\ y_{Li} \end{bmatrix} = \begin{bmatrix} d_{Li} \cos(\alpha_{Li}) \\ d_{Li} \sin(\alpha_{Li}) \end{bmatrix}, \quad \text{com } i = 1, 2, \dots, N_L \quad (3.1)$$

onde  $N_L$  é o número de feixes de luz disparados em uma varredura.

O ângulo máximo e mínimo possível dos feixes em uma varredura pode variar de 0 a 360° ou de -45° a 225°, por exemplo, dependendo das características de cada modelo do *laser*. Ressalta-se ainda que o intervalo angular entre feixes consecutivos é denominado resolução do *laser*.

### 3.1.2 Descrição do método

Como já mencionado, o método em questão utiliza dois sensores *laser scan* posicionados na parte superior e à frente do robô, para realizar os escaneamentos verticais e horizontais do ambiente. A figura 3.3 demonstra os *lasers* realizando os escaneamentos vertical e horizontal.

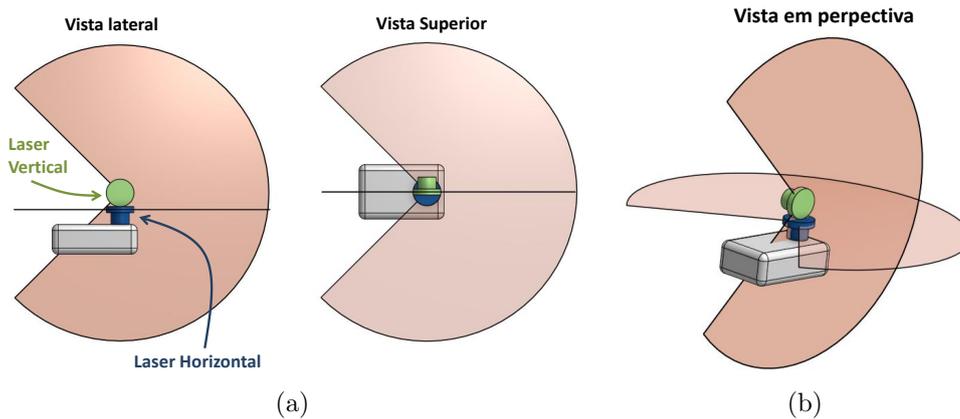


Figura 3.3: (a) apresenta as vistas lateral e superior dos *lasers* vertical e horizontal realizando o escaneamento do ambiente e (b) apresenta a vista em perspectiva

O posicionamento vertical permite que o *laser* capture a geometria característica dos degraus da escada e deste modo o algoritmo consegue identificar a presença ou não da escada na direção do escaneamento. Já o *laser* horizontal permite determinar a localização da escada e sua orientação através da captura dos segmentos de linha que compõem o ambiente. Desta maneira, o método é dividido em duas etapas: a detecção da escada e a localização da mesma.

Para auxiliar a compreensão do método, a figura 3.4 apresenta os sistemas de coordenadas adotados, de modo que  $O_{L_H}$  e  $O_{L_V}$  são os sistemas de coordenadas dos *lasers* horizontal e vertical, respectivamente,  $O_R$  é o sistema de coordenadas do robô e  $O_B$  é o referencial inercial.

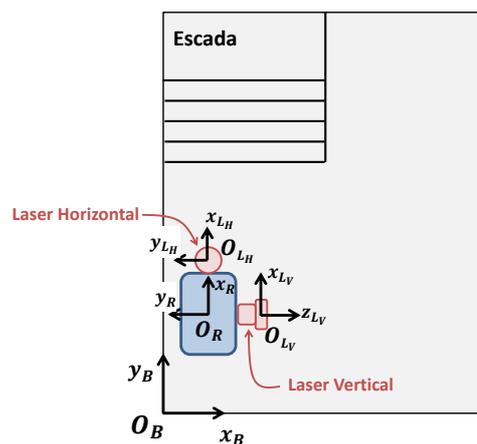


Figura 3.4: Vista superior do posicionamento dos sistemas de coordenadas  $O_{L_H}$ ,  $O_{L_V}$ ,  $O_R$  e  $O_B$  em relação à escada

### Etapa de detecção da escada

Como já mencionado, a etapa de detecção utiliza o *laser* na posição vertical para identificar a geometria da escada. A figura 3.5 demonstra como a varredura do *laser*

captura os degraus.

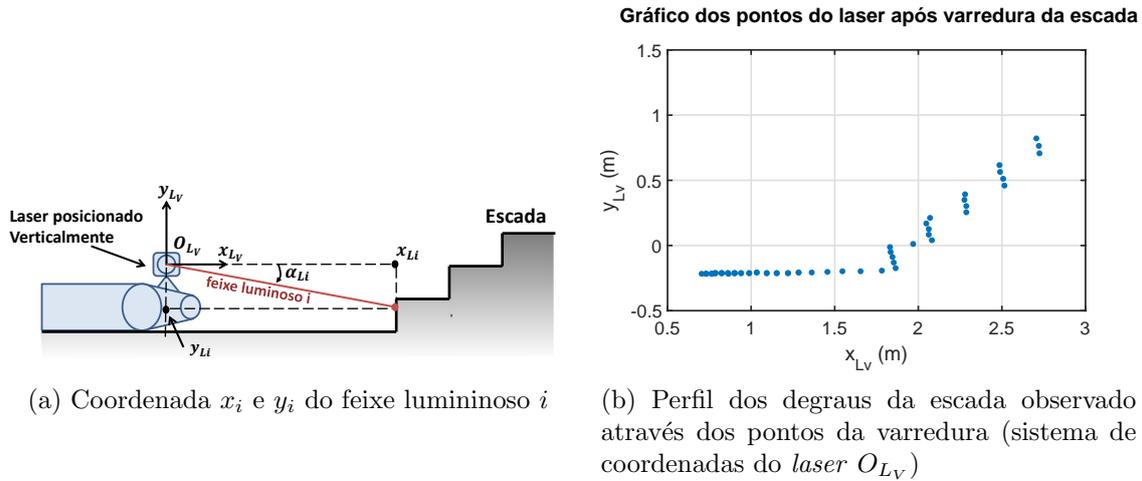


Figura 3.5: Sensor *laser* realizando varredura vertical da escada.

O método consiste em avaliar as diferenças dimensionais das coordenadas  $x$  e  $y$  entre o ponto atual do *laser* e o ponto anterior, percorrendo assim a sequência de dados do sensor do início ao fim.

A coordenada  $x$  de pontos consecutivos no piso, por exemplo, apresentam uma diferença mínima, por outro lado ao analisar os pontos consecutivos frontais ao degrau, essa diferença será próxima de zero. Ao encontrar uma diferença próxima de zero entre as coordenadas  $x$  de pontos consecutivos, caracteriza-se a possível presença de um segmento vertical pertencente a um degrau ou uma parede, por exemplo.

Mais pontos do *laser* são verificados sequencialmente e validados se as diferenças das coordenadas  $x$  forem próximas de zero, até que um número mínimo de pontos com reduzida diferença de coordenada  $x$  é obtido. Em seguida, calcula-se a diferença entre a coordenada  $y$  do primeiro e último pontos pertencentes ao segmento, obtendo assim a altura do mesmo. É avaliado então se a altura do segmento está entre limites mínimo e máximo, descartando segmentos pertencentes a uma parede, por exemplo. Caso o segmento não seja descartado, o procedimento continua até serem detectados dois segmentos de reta consecutivos. Calcula-se então a diferença em coordenada  $x$  dos dois segmentos, determinando a profundidade de um possível degrau. É verificado se a profundidade calculada está entre limites mínimo e máximo, validando finalmente a existência de um degrau.

O algoritmo prossegue até encontrar um número mínimo de degraus, confirmando a existência da escada, ou até encerrar os pontos daquela varredura, determinando que a escada não foi encontrada.

Para auxiliar a compreensão da abordagem, a seguir são descritos o pseudocódigo 1 do método apresentado e a notação utilizada:

- a)  $x_{max}$ : Máxima diferença tolerada entre pontos consecutivos na direção- $x$ , para que o ponto corrente pertença ao segmento de linha corrente;
- b)  $n_{pts}$ : Número de pontos no segmento de linha corrente;
- c)  $n_{pmin}$ : Número mínimo de pontos em um segmento para ser considerado um possível degrau;
- d)  $y_0$ : Coordenada  $y$  do ponto inicial do segmento de linha atual;
- e)  $y_e$ : Coordenada  $y$  do ponto final do segmento de linha atual;
- f)  $h_{lS} := (y_e - y_0)$ : Altura do segmento de linha atual em que  $h_{lSmin}$  e  $h_{lSmax}$  são os limites inferiores e superiores para um degrau válido, respectivamente;
- g)  $x_c$ : Coordenada  $x$  do segmento de linha corrente e  $x_p$  a coordenada do segmento de linha anterior;
- h)  $d_{lS} := (x_c - x_p)$ : Profundidade entre segmentos de linha verticais consecutivos com  $d_{lSmin}$  e  $d_{lSmax}$  sendo os limites mínimo e máximo para profundidade válida de um degrau, respectivamente;
- i)  $n_g$ : Número de segmentos de linha de degraus detectados; e
- j)  $n_{gmin}$ : Número mínimo de degraus consecutivos encontrados, para caracterizar se o escaneamento atual do *laser* detectou uma escada.

---

**Pseudocódigo 1** Detecção de escadas utilizando *laser scan*

---

```
1: Obter os pontos de um escaneamento vertical do laser ( $p_{t1}, p_{t2}, p_{t3}, \dots, p_{tend}$ )
2: Inicializar as variáveis  $n_{pts} = y_s = y_e = x_c = x_p = n_g = 0$ .
3: for  $i = 1$  to end do
4:   obter  $x_i$  e  $y_i$  do ponto  $p_{ti}$  do laser
5:   if  $x_c == 0$  then
6:      $x_c = x_i$ 
7:      $y_0 = y_i$ 
8:   else if  $|x_c - x_i| \leq x_{max}$  then
9:      $n_{pts} = n_{pts} + 1$ 
10:  else if  $n_{pts} > n_{pmin}$  then
11:     $y_e = y_{i-1}$ 
12:     $h_{lS} = y_e - y_0$ 
13:    if  $h_{lSmin} \leq h_{lS} \leq h_{lSmax}$  then
14:       $d_{lS} = x_c - x_p$ 
15:      if  $d_{lSmin} \leq d_{lS} \leq d_{lSmax}$  then
16:         $n_g = n_g + 1$ 
17:         $x_p = x_c$ 
18:        if  $n_g > n_{gmin}$  then
19:          return Escada detectada
20:        end if
21:      else
22:         $x_p = x_c$ 
23:         $n_g = 1$ 
24:      end if
25:    end if
26:     $x_c = x_i$ 
27:     $y_0 = y_i$ 
28:     $n_{pts} = 0$ 
29:  else
30:     $x_c = x_i$ 
31:     $y_0 = y_i$ 
32:     $n_{pts} = 0$ 
33:  end if
34: end for
35: return Escada não encontrada
```

---

### Etapa de localização da escada

Para localizar a escada, utiliza-se um sensor *laser* horizontal para escanear o ambiente e a partir do método de extração de linhas, é possível identificar qual das linhas representa o início da escada. Contudo, para aplicar o método é necessário adicionar duas hipóteses à formulação do problema realizada no capítulo 2. As hipóteses seguem abaixo:

**Hipótese 3.1** *A largura da escada é conhecida.*

**Hipótese 3.2** *Nos limites laterais da escada, não há paredes alinhadas com o primeiro degrau da mesma.*

A hipótese 3.2 pode ser melhor observada na figura 3.6.

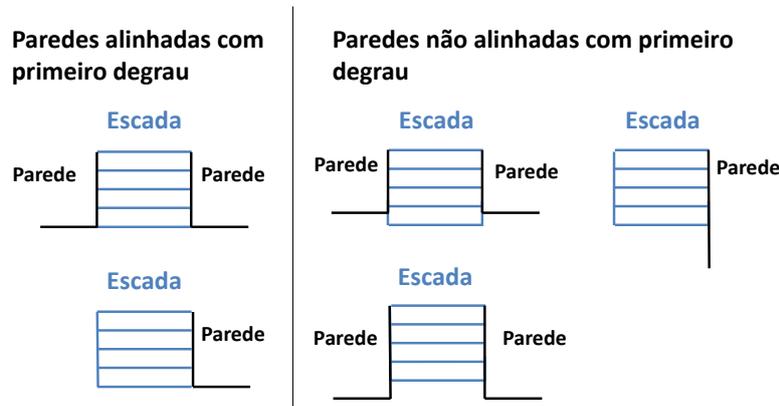


Figura 3.6: Ilustração demonstrando casos com paredes alinhadas e não alinhadas com o primeiro degrau da escada

O método para extração de linhas proposto em Zhang et al. (2011) se assemelha à técnica Incremental (Nguyen et al. 2005). A técnica Incremental consiste basicamente em calcular os parâmetros da linha inicial utilizando dois pontos, em seguida é verificado se o próximo ponto satisfaz a condição para pertencer à linha. Satisfeita a condição, o ponto é adicionado à linha, que tem seus parâmetros recalculados. Caso não seja satisfeita a condição, a linha é encerrada, iniciando novamente o algoritmo a partir do próximo ponto. Esta sequência de eventos é executada até alcançar o último ponto do escaneamento corrente do *laser*. Ao final do método são retornadas as linhas extraídas que melhor se ajustam aos pontos analisados.

Para tornar o método de extração de linhas mais robusto à presença de ruídos do sensor e *outliers*<sup>1</sup>, decidiu-se utilizar o método *Random Sample Consensus (Ransac)* (Fischler & Bolles 1981) para a extração de linhas.

O *Ransac* é amplamente utilizado para ajuste de modelo a dados experimentais. Sua principal característica é a robustez a ruídos nos dados e rapidez, visto que não exige que sejam avaliados todos os pontos da amostra para estimar o modelo. Deste modo, optou-se por aplicar o método *Ransac* para extrair as linhas do escaneamento horizontal do *laser* e determinar a localização da escada.

O pseudocódigo 2 explicita como se processa o método *Ransac* aplicado à extração de linhas. Entretanto, antes de realizar a extração de linhas, propõe-se aplicar dois tipos de segmentação dos dados para obter melhores resultados.

<sup>1</sup> *outliers* neste contexto significa pontos isolados que se tornam distantes da sequência de pontos da varredura do *laser*. São erros grosseiros nos dados.

A primeira segmentação é baseada na distância entre pontos consecutivos da varredura do *laser*. Deste modo, quando as distâncias calculadas são maiores que um limite máximo, os grupos de pontos são separados em dois grupos (*clusters*) distintos e assim continua o processo de segmentação até o último ponto do escaneamento. A segmentação pode ser observada na figura 3.7a.

Já a segunda segmentação é baseada na distância do ponto à linha que une os pontos extremos do grupo avaliado. Desta maneira, verifica-se para cada ponto no grupo, se a distância até a linha formada pelos extremos é maior que um limite máximo. Se a distância medida for maior, então o grupo é segmentado em dois, de modo que o algoritmo é aplicado recursivamente nos grupos que surgem a cada iteração, até que não apareçam novos *clusters*. A segmentação em questão pode ser observada na figura 3.7b.

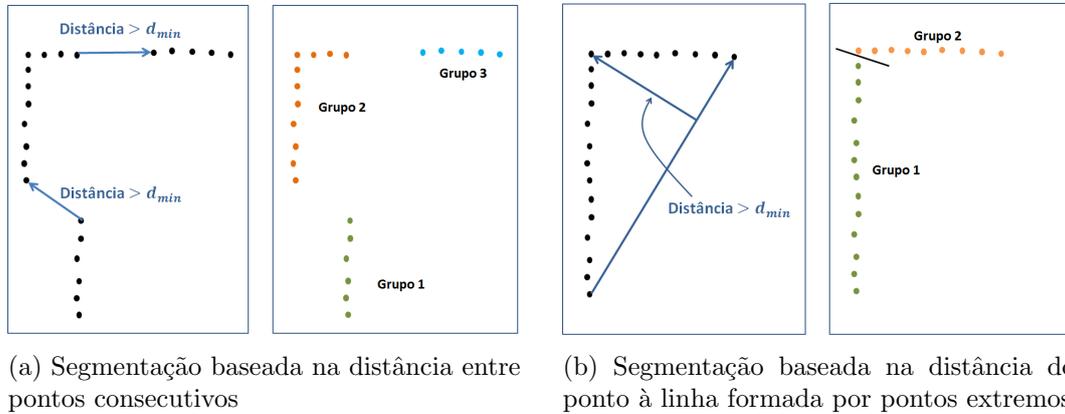


Figura 3.7: Segmentações aplicadas no método de detecção de escada utilizando sensor *laser*

Após realizar as segmentações descritas, aplica-se o algoritmo de extração de linhas baseado no *Ransac*. Antes de apresentar o pseudocódigo para extração de linhas é necessário definir a notação utilizada:

- a)  $N$ : Número de grupos de pontos encontrados após a segmentação;
- b)  $n_{iter}$ : Número de iterações realizadas pelo algoritmo
- c)  $cl_i$ :  $i$ -ésimo grupo de pontos segmentado;
- d)  $m_0, b_0$ : Modelo da reta inicial ( $y = m_0x + b_0$ );
- e)  $d_j$ : Distância do  $j$ -ésimo ponto à reta;
- f)  $d_{thres}$ : Distância máxima do ponto à reta para considerá-lo pertencente ao modelo;
- g)  $pts_{in}$ : Grupo de pontos que satisfazem a distância máxima à reta modelo;
- h)  $n_{in}$ : Número de pontos pertencentes a  $pts_{in}$
- i)  $n_{thres}$ : Número mínimo de pontos que devem pertencer ao modelo para considerá-lo válido;
- j)  $m_{md}, b_{md}$ : Melhor modelo de reta ajustado pelo método de mínimos quadrados.

O pseudocódigo do algoritmo é descrito abaixo:

---

**Pseudocódigo 2** Extração de linhas utilizando *Ransac*

---

```

1: Para cada grupo de pontos segmentado ( $cl_0, cl_1, \dots, cl_N$ )
2: for  $k = 1 : n_{iter}$  do
3:   Inicialize o modelo  $(m_0, b_0)$  da reta utilizando dois pontos aleatórios.
4:   Para cada ponto pertencente a  $cl_i$ , calcule a distância  $d_j$  do ponto à reta modelo.
5:   Armazenar os pontos do grupo  $cl_i$  tal que  $d_j < d_{thres}$ .
6:   O resultado do passo anterior é o cálculo de  $pts_{in}$  e  $n_{in}$ .
7:   if  $(n_{in} > n_{thres}$  and  $n_{in} > n_{md})$  then
8:     Calcular novo modelo  $(m_{md}, b_{md})$  com os  $pts_{in}$ .
9:      $n_{md} = n_{in}$ 
10:  end if
11: end for
12: return  $m_{md}, b_{md}$ .

```

---

Tendo em vista que a largura da escada é conhecida (hipótese 3.2), para encontrar a linha que corresponde ao início da escada, basta que duas condições sejam satisfeitas: a linha extraída do algoritmo deve possuir a largura correspondente a da escada, e a distância do primeiro degrau até a linha extraída deve ser o menor possível, ou seja, menor que um limite máximo.

Após encontrar a linha horizontal correspondente ao primeiro degrau da escada, a sua orientação pode ser facilmente determinada obtendo-se o vetor ortogonal à linha mencionada. O ângulo  $\phi_S$  entre o referido vetor e o eixo  $x_B$  do sistema de coordenadas inercial  $O_B$  define a orientação da escada.

**Observação 3.1** *A alteração da etapa de extração de linhas do método de localização de escadas apresentado em Zhang et al. (2011), substituindo o algoritmo de extração de linhas original pelas etapas de segmentação dos dados do laser e aplicação do algoritmo Ransac, tornou o método de localização de escadas mais rápido, visto que o Ransac não utiliza todos os dados da amostra para realizar o ajuste do modelo, além de torná-lo mais robusto a ruídos no sensor. Desta maneira, podemos considerar as alterações como contribuições para melhoria da estratégia de detecção e modelagem de escadas original.*

### 3.1.3 Simulações e Experimentos

O método de detecção de escadas baseado em *lasers scan* foi avaliado através de experimentos utilizando dois sensores, um posicionado verticalmente e outro horizontalmente. Os dados do *laser* foram obtidos através da plataforma *ROS (Robot Operating System)* que consiste em um *framework* para desenvolvimento de softwares de robótica. O algoritmo de detecção e localização por sua vez foi implementado utilizando o software Matlab.

O posicionamento dos *lasers* horizontal e vertical e seus respectivos sistemas de coordenadas  $O_{L_H}$  e  $O_{L_V}$ , além do sistema de coordenadas inercial  $O_B$  estão representados na figura 3.8.

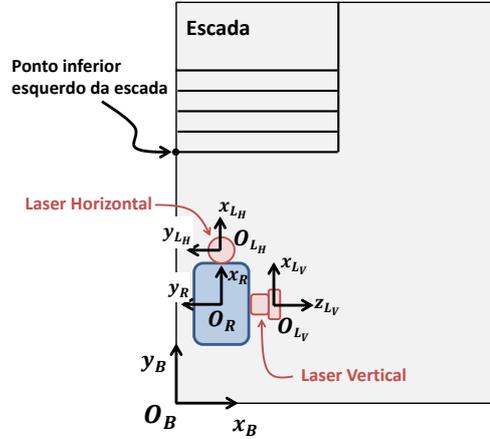


Figura 3.8: Vista superior do posicionamento dos sistemas de coordenadas  $O_{L_H}$ ,  $O_{L_V}$ ,  $O_R$  e  $O_B$  em relação à escada

A escada utilizada no experimento apresenta cinco degraus com altura ( $h_S = 0.17$  m) e profundidade ( $d_S = 0.30$  m). As dimensões totais da escada são: altura ( $H_S = 0.85$  m), largura ( $W_S = 1.2$  m) e profundidade ( $D_S = 1.68$  m).

A posição dos *lasers* horizontal e vertical em relação ao sistema de coordenadas do robô ( $O_R$ ) é definida pelos vetores  $(p_{L_H})_R = [0.25 \ 0 \ 0.20]$  m e  $(p_{L_V})_R = [0 \ -0.12 \ 0.26]$  m, respectivamente. O subscrito  $( )_R$  indica que as posições foram representadas no sistema de coordenadas  $O_R$  solidário ao robô. A orientação da escada é  $\phi_S = 90^\circ$  medida em relação ao eixo  $x_B$  do sistema de coordenadas inercial  $O_B$ .

No primeiro experimento, o robô com os dois *lasers* acoplados em seu chassi foi posicionado a uma distância em profundidade (direção  $-y_B$ ) de 2.73 m, distância lateral (direção  $x_B$ ) de 0.41 m, ambas medidas em relação ao ponto inferior esquerdo da escada (figura 3.8).

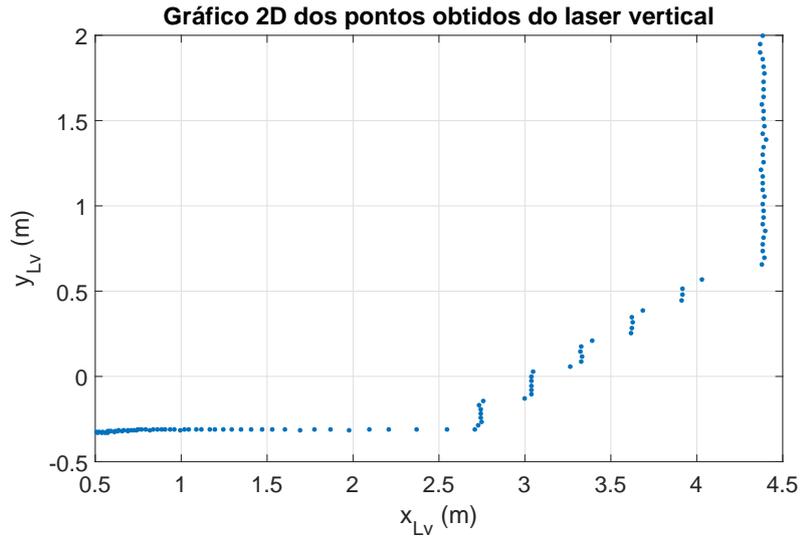


Figura 3.9: Gráfico das coordenadas 2D dos pontos capturados a partir do *laser* vertical (sistema de coordenadas  $O_{L_V}$ )

Para detecção da escada foram realizados cinco escaneamentos do *laser*, de modo que a figura 3.9 apresenta os ponto 2D de um dos escaneamentos.

Os principais parâmetros definidos nesta etapa são os limites de altura para um degrau válido  $(h_{LSmin}, h_{LSmax}) = (0.10, 0.30)$  m, os limites de profundidade  $(d_{LSmin}, d_{LSmax}) = (0.20, 0.45)$  m, o número mínimo de degraus  $n_{gmin} = 3$  para se considerar uma escada, a diferença máxima  $x_{max} = 0.02$  m para o ponto pertencer ao segmento de linha e por fim o número de pontos mínimo  $n_{pmin} = 4$  para se considerar um segmento de linha.

Os resultados da primeira etapa de detecção são demonstrados na tabela 3.1 abaixo:

	<b>Experimental</b>	<b>Real</b>
Altura média dos degraus ( $h_S$ [m])	0.122	0.17
Profundidade média dos degraus ( $d_S$ [m])	0.303	0.30
Número de degraus detectados ( $n_g$ )	3	5

Tabela 3.1: Dados do primeiro experimento de detecção de escadas utilizando *lasers scan*

Como pode ser observado a estimativa da profundidade dos degraus ( $d_S$ ) apresentou erro de apenas 0.003 m (1%) em relação ao valor real, contudo a medida da altura dos degraus ( $h_S$ ) apresentou erro mais elevado de 0.048 m (28%). Ressalta-se ainda que, o método foi capaz de detectar apenas 3 dos 5 degraus da escada.

O erro mais elevado na estimativa da altura ( $h_S$ ) e a não detecção dos últimos dois degraus da escada ocorrem devido ao fato dos feixes do *laser* não capturarem por completo os degraus superiores, pois estes se tornam oclusos pelos degraus anteriores. O problema de oclusão pode ser observado na figura 3.10.

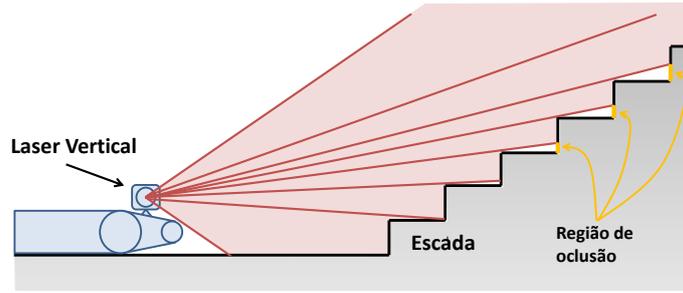


Figura 3.10: Ilustração do problema de oclusão dos degraus da escada capturados pelo *laser*

Ainda na primeira etapa foram obtidas as coordenadas do primeiro e segundo degraus da escada representadas no sistema de coordenadas do robô  $O_R$ , sendo elas  $(p_{1s})_R = [2.71 \quad -0.12 \quad -0.1]$  m e  $(p_{2s})_R = [3.01 \quad -0.12 \quad -0.1]$  m, respectivamente.

**Observação 3.2** A posição  $(p_{2s})_R$  do segundo degrau da escada foi utilizada, pois posicionou-se o laser horizontal a uma altura do solo que permitiu capturar apenas o referido degrau, desta maneira para determinar a localização da escada, foi necessário utilizar o ponto  $(p_{2s})_R$ .

Após executar a primeira etapa de detecção é realizada a segunda etapa de localização da escada.

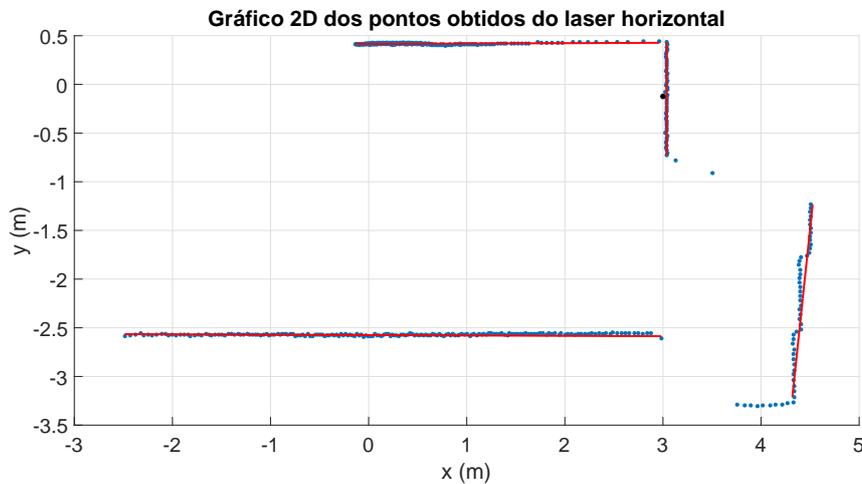


Figura 3.11: Gráfico das coordenadas 2D dos pontos capturados a partir do *laser* horizontal e linhas extraídas pelo método *Ransac* (sistema de coordenadas  $O_R$ )

Como é demonstrado na figura 3.11, as linhas em vermelho foram detectadas através do método *Ransac* e o ponto na cor preta representa a posição  $(p_{2s})_R$  do segundo degrau da escada capturado durante a primeira etapa do algoritmo. Desta maneira, a estimativa da posição da escada obtida através do algoritmo foi  $[-0.018 \quad 5.99 \quad 0]$  m, enquanto o valor real da posição é  $[0 \quad 6 \quad 0]$  m, ambas medidas em relação ao sistema de coordenadas  $O_B$ . Portanto, o erro de estimativa da posição da escada é de 0.019 m (0.3%), o que caracteriza um bom resultado. Ressalta-se

que a localização da escada é representada pelo ponto inferior à esquerda da mesma, como representado na figura 3.8. A orientação da escada obtida em relação ao eixo  $x_B$  foi  $\phi_S = 90^\circ$ .

Apesar de ser capaz de detectar a escada e estimar a sua localização com boa precisão, o método apresenta problemas em duas situações: quando a distância do robô à escada é elevada e quando existem ruídos nas medidas do *laser*. Os ruídos nos dados do *laser* dificultam a identificação das linhas verticais através do método, enquanto que o aumento da distância entre sensor e escada intensifica o problema de oclusão já discutido anteriormente, causando falhas na detecção dos degraus. O gráfico da figura 3.12 apresenta a varredura vertical para o robô posicionado a uma distância em profundidade (direção  $-y_B$ ) de 6 m em relação ao ponto inferior esquerdo da escada.

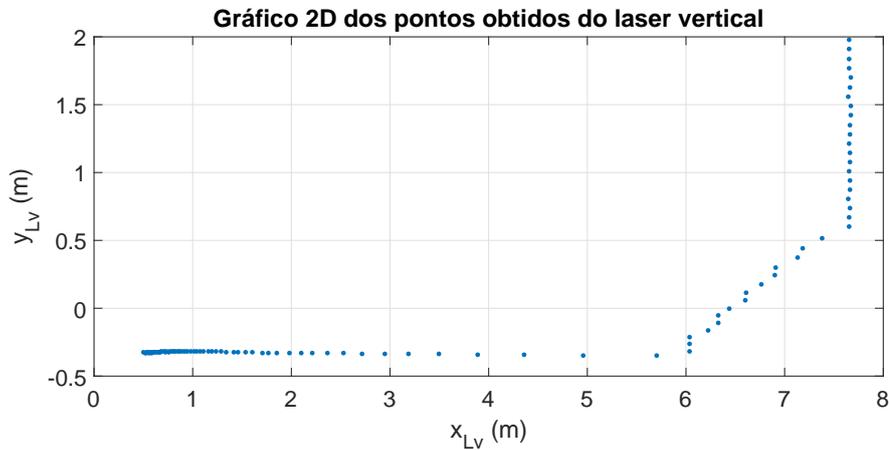


Figura 3.12: Gráfico das coordenadas 2D dos pontos capturados a partir do *laser* vertical (sistema de coordenadas  $O_{Lv}$ )

Como pode ser observado, as linhas verticais dos degraus não aparecem no gráfico em virtude da distância do sensor à escada e, conseqüentemente, do problema de oclusão. Portanto, foi concluído através de experimentos que o método não consegue identificar a escada para distâncias superiores a 5 m.

## 3.2 Detecção e modelagem de escadas utilizando imagem 3D densa

No estudo desenvolvido em Delmerico et al. (2013) é proposto método para realizar a detecção da escada e ajustar os dados obtidos a um modelo, utilizando um sensor 3D como as câmeras RGB-D ou *Kinect* desenvolvido pela empresa *Microsoft*. Um dos principais aspectos do método de detecção em questão é fazer uso das descon-

tinuidades características da geometria das escadas, avaliando assim a forma como as descontinuidades são evidenciadas nos dados produzidos pelo sensor.

Para facilitar a compreensão da metodologia estudada, nas próximas subseções são descritos o sensor 3D *Kinect* e o método propriamente dito.

### 3.2.1 Sensor 3D *Kinect*

O *Kinect* foi desenvolvido pela empresa *Microsoft* em conjunto com a companhia *Prime Sense* para ser aplicado como sensor de movimentos em jogos de videogames. Atualmente, sensores 3D como o *Kinect* e câmeras RGB-D são largamente utilizados em aplicações robóticas como localização e mapeamento (Endres et al. 2012), navegação autônoma (Valenti et al. 2014), rastreamento e detecção (Shao et al. 2013), dentre outras.

O *Kinect* é composto basicamente por um projetor infravermelho, uma câmera infravermelho e uma câmera RGB . Os elementos responsáveis por obter os dados 3D do ambiente são basicamente o projetor infravermelho e a câmera infravermelho. A figura 3.13 mostra a disposição dos sensores e projetor no *Kinect*.



Figura 3.13: Câmeras e projetor que compõem o *Kinect*

A descrição de modo simplificado do princípio de funcionamento do *Kinect* consiste em um mapeamento 3D baseado na projeção de padrões. O projetor infravermelho realiza a projeção de um conjunto de padrões no ambiente, que apresenta objetos com diferentes formas e profundidades em relação ao sensor.

Os padrões projetados na superfície dos objetos sofrem distorções, alongamentos que variam de acordo com a distância entre a superfície do objeto e o sensor. Deste modo, os padrões distorcidos são mapeados e comparados com os padrões de referência e, através de processamento de imagens, as distâncias dos padrões projetados ao sensor são calculadas baseadas nas distorções.

Assim sendo, o sensor *Kinect* encontra as coordenadas 3D de diversos pontos no ambiente que está sendo capturado. A figura 3.14a demonstra as distorções que ocorrem nos padrões projetados para diferentes distâncias e a figura 3.14b apresenta uma imagem real em infravermelho com os padrões criados pelo *Kinect* .

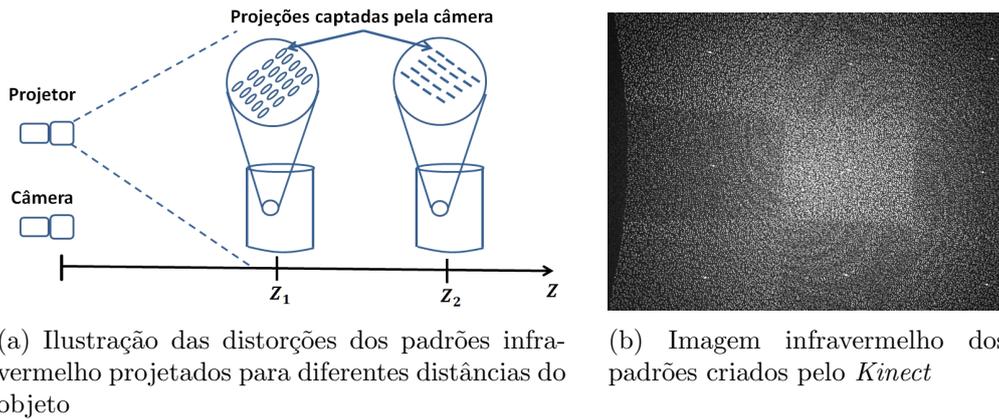


Figura 3.14: Exemplo do mapeamento 3D realizado pelo *Kinect*.

O *Kinect* fornece diversos tipos de imagens e dados, os principais utilizados nesta dissertação foram o *point cloud* e a *depth image*. O *point cloud* consiste no mapeamento 3D do ambiente, ou seja, é um conjunto de pontos com as coordenadas  $x, y$  e  $z$  representando as superfícies do ambiente. Já a *depth image* consiste em uma imagem em que a cor de cada pixel é definida por uma relação com a distância do ponto da superfície representado pelo pixel ao sensor. A *depth image* pode ser representada em escala de cinza ou em escala de cores.

### 3.2.2 Descrição do método

O método de detecção e modelagem baseado em sensores 3D pode ser dividido em duas etapas: a primeira é a detecção das bordas da escada e a segunda é a modelagem e localização da escada.

A primeira etapa leva em consideração a descontinuidade característica dos degraus da escada e a consequente percepção desta característica na imagem em profundidade.

Como já foi mencionado, a *depth image* exibe em cada pixel níveis de intensidade em escala de cinza. Estes níveis de intensidade são proporcionais à distância do sensor à superfície representada pelo pixel em questão. Deste modo, ao exibir a *depth image* de uma escada, é possível observar descontinuidades da imagem em escala de cinza devido às diferenças de profundidade de um degrau para outro. Na figura 3.15 observam-se as descontinuidades em escala de cinza na imagem de uma escada.

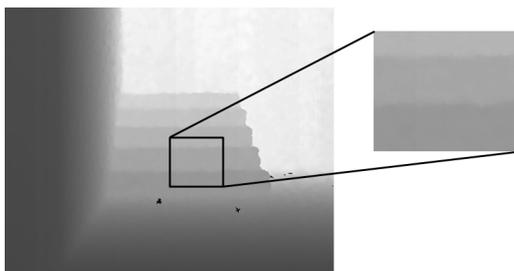


Figura 3.15: Descontinuidades observadas na imagem em profundidade (*depth image*) de uma escada

Após realizar a detecção de bordas através do método de *Canny* (Canny 1986), todas as bordas da *depth image* são evidenciadas, inclusive as bordas dos degraus, devido às diferenças de intensidade já mencionadas. Em seguida, o método *Probabilistic Hough Transform* (Kiryati et al. 1991) é aplicado para que sejam extraídas as linhas retas correspondentes às bordas da imagem. As linhas pertencentes às bordas dos degraus formam um conjunto de retas paralelas, independente da rotação do sensor *Kinect* em relação à escada. A figura 3.16 demonstra a sequência de imagens após a detecção das bordas e detecção das linhas mencionadas.

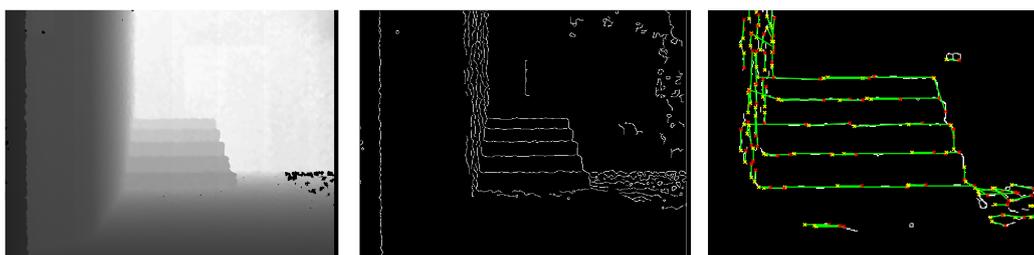


Figura 3.16: As figuras demonstram a aplicação do método de detecção de escadas utilizando sensor 3D. Da esquerda para a direita, a primeira figura demonstra a *depth image*, a segunda demonstra a detecção de bordas através do método de *Canny* e a última apresenta em verde as linhas detectadas a partir da imagem das bordas.

Em seguida, as linhas com os ângulos semelhantes são agrupadas em *clusters* (agrupamentos), e o grupo com maior número de linhas e ângulo próximo ao ângulo horizontal é escolhido como um possível *cluster* das linhas pertencentes aos degraus. O *point cloud* referente à *depth image* é obtido, e os pontos em 3D referentes às linhas agrupadas são também armazenados. A partir desta etapa, os pontos 3D obtidos possivelmente correspondem às linhas das bordas dos degraus da escada.

Através do método de mínimos quadrados, um plano é ajustado aos pontos 3D extraídos, em seguida, é verificado se o ângulo deste plano com o plano do solo está entre limites mínimo e máximo. Em caso positivo, o grupo de pontos referentes às linhas das bordas dos degraus são armazenados, em caso negativo os dados são descartados e uma nova captura da escada é realizada.

A seguir são apresentados o pseudocódigo do método descrito e as definições estabelecidas:

- a)  $Im_D$ : *Depth image* da escada obtida pelo *Kinect*;
- b)  $P_{cloud}$ : Conjunto de pontos 3D (*point cloud*) correspondente à *depth image* ( $Im_D$ );
- c)  $Im_E$ : Imagem das bordas detectadas através do método de *Canny* aplicado à imagem ( $Im_D$ );
- d)  $L$ : Conjunto de linhas detectadas da imagem das bordas  $Im_E$ ;
- e)  $H_\alpha$ : Histograma dos ângulos das linhas em  $L$ ;
- f)  $L'$ : Linhas extraídas do histograma ( $H_\alpha$ ) com maior frequência e direção horizontal;
- g)  $H_{col}$ : Histograma do número de linhas em cada coluna da imagem das bordas ( $Im_E$ );
- h)  $b_{rect}$ : Caixa representada na imagem  $Im_E$  que delimita o conjunto com maior número de linhas em  $L'$  sobrepostas verticalmente na imagem;
- i)  $P'_{cloud}$ : Pontos 3D correspondentes às linhas em  $L'$  no interior de  $b_{rect}$ ;
- j)  $pl_{temp}$ : Plano temporário ajustado ao conjunto de pontos  $P'_{cloud}$  pelo método de mínimos quadrados;
- k)  $\alpha_{temp}$ : Ângulo diedral entre o plano  $pl_{temp}$  e o plano do solo.
- l)  $n_{pl}$ : Vetor normal ao plano da escada.
- m)  $n_{horiz}$ : Vetor normal ao plano do solo.

---

**Pseudocódigo 3** Detecção das bordas da escada utilizando o sensor 3D *Kinect*

---

- 1: Entrada: Imagem em profundidade  $Im_D$  e o *point cloud*  $P_{cloud}$  correspondente, fornecido pelo sensor 3D (*Kinect*).
  - 2: Aplicar *Canny Edge Detection* em  $Im_D$  para produzir a imagem de bordas  $Im_E$ .
  - 3: Setar para zero todos os *pixels* em  $Im_E$  vizinhos dos *pixels* com valor de profundidade 0 na *depth image* ( $Im_D$ ).
  - 4: Gerar um conjunto de linhas candidatas  $L$ , aplicando *Probabilistic Hough Transform* em  $Im_E$ .
  - 5: Unir linhas colineares e gerar um histograma  $H_\alpha$  dos ângulos das linhas de  $L$ .
  - 6: Extrair as linhas referentes ao intervalo do histograma com maior frequência e gerar  $L'$ .
  - 7: A partir das linhas  $L'$ , gerar o histograma  $H_{col}$  com o número de linhas em cada coluna da imagem  $Im_E$ .
  - 8: Encontrar a máxima frequência em  $H_{col}$  e obter as colunas mais à esquerda e mais à direita na imagem  $Im_E$  que apresentam a frequência máxima encontrada.
  - 9: Obter as linhas superior e inferior na imagem  $Im_E$  que se encontram dentro das colunas definidas no passo anterior.
  - 10: Com os limites superior/inferior e mais a esquerda/direita, definir a caixa limitadora  $b_{rect}$  na imagem  $Im_E$  com os limites mencionados.
  - 11: Remover todas as linhas de  $L'$  que não estão dentro de  $b_{rect}$ .
  - 12: **if** (número de linhas em  $L'$ ) < 3 **then**
  - 13:     **return**  $P'_{cloud} = 0$
  - 14: **else**
  - 15:     Extrair de  $P_{cloud}$  os pontos 3D correspondentes às linhas em  $L'$  e gerar o conjunto  $P'_{cloud}$ .
  - 16:     Ajustar o plano  $pl_{temp}$  aos pontos em  $P'_{cloud}$  aplicando o método de mínimos quadrados.
  - 17:     Calcular o ângulo diedral ( $\alpha_{temp} = \arccos(n_{pl} \cdot n_{horiz})$ ) entre o plano  $pl_{temp}$  e o plano do solo.
  - 18:     **if**  $0 < \alpha_{temp} < \alpha_{max}$  **then**
  - 19:         **return**  $P'_{cloud}$
  - 20:     **else**
  - 21:         **return**  $P'_{cloud} = 0$
  - 22:     **end if**
  - 23: **end if**
- 

Após diversas capturas da escada, novas detecções das bordas são realizadas e novos pontos em 3D são armazenados, formando um grupo denso de pontos 3D das bordas dos degraus da escada. Neste momento, inicia-se a segunda etapa do método, a modelagem e localização da escada.

Para a modelagem da escada, utiliza-se um modelo geral que é capaz de fornecer informações como altura e profundidade média dos degraus, o número de degraus, além da inclinação, orientação e localização da escada.

O modelo em questão consiste em um plano inclinado  $pl$  limitado por uma caixa limitadora  $B_l$ . O plano inclinado é formado pelas linhas pertencentes às bordas dos degraus da escada.

Desta maneira, os parâmetros do modelo são definidos como segue:

- a)  $B_{l,cx}, B_{l,cy}, B_{l,cz}$ : Coordenadas do centroide da caixa delimitadora  $B_l$ ;

- b)  $H_{B_l}, W_{B_l}, D_{B_l}$ : Dimensões de altura, largura e profundidade da caixa delimitadora  $B_l$ , respectivamente;
- c)  $pl$ : Plano da escada formado pelas linhas das bordas dos degraus;
- d)  $h_S$ : Altura dos degraus da escada;
- e)  $d_S$ : Profundidade dos degraus da escada;
- f)  $\alpha_S$ : Ângulo de inclinação do plano ( $pl$ ) em relação ao solo, ou seja, ângulo de inclinação da escada.

O modelo do plano inclinado  $pl$  utilizado no método é definido pela seguinte equação:

$$ax + by + cz = d \quad (3.2)$$

Da mesma maneira, o modelo da caixa delimitadora  $B_l$  é definido pelas equações:

$$y = B_{l,cy} \pm \frac{H_{B_l}}{2} \quad (3.3)$$

$$a(x - B_{l,cx}) + c(z - B_{l,cz}) \pm \frac{D_{B_l}}{2}(\sqrt{a^2 + c^2}) = 0 \quad (3.4)$$

$$-c(x - B_{l,cx}) + a(z - B_{l,cz}) \pm \frac{W_{B_l}}{2}(\sqrt{a^2 + c^2}) = 0 \quad (3.5)$$

onde a equação 3.3 representa os planos superior e inferior da caixa delimitadora  $B_l$ , a equação 3.4 representa os planos frontal e posterior, e, por fim, a equação 3.5 representa os planos laterais.

A figura 3.17 ilustra o plano inclinado  $pl$ , formado pelos pontos 3D das bordas dos degraus, e a caixa delimitadora  $B_l$ .

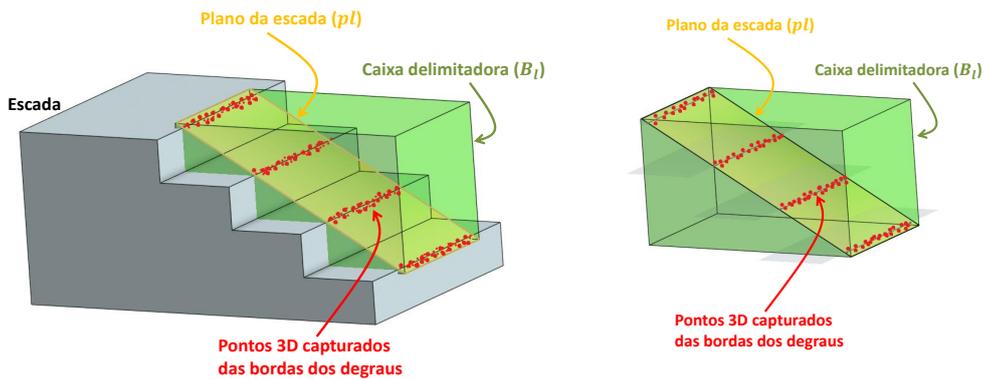


Figura 3.17: Exemplo do plano  $pl$  ajustado aos pontos 3D das bordas dos degraus e a caixa delimitadora  $B_l$ , ambos obtidos através do método de detecção e modelagem de escadas utilizando o *Kinect*.

Para simplificar o método de modelagem proposto em Delmerico et al. (2013) e atender às necessidades do problema considerado nesta dissertação, algumas al-

terações foram realizadas. O método original apresenta a capacidade de modelar múltiplas escadas simultaneamente, entretanto o problema proposto não exige a competência mencionada, de modo que os passos executados para modelar mais de uma escada foram retirados.

Após realizar entre cinco e dez capturas utilizando o método de detecção, o conjunto denso de pontos 3D das bordas dos degraus passa a ser denotado  $E_{cloud}$ .

Em seguida, ajusta-se o plano  $pl$  aos dados através do método *Ransac* (Fischler & Bolles 1981). Nesta etapa foi aplicado o mesmo pseudocódigo 2 do algoritmo *Ransac* descrito na subseção 3.1.2. As diferenças básicas entre os dois algoritmos estão centradas na escolha dos pontos para iniciar o modelo, que para o plano são escolhidos três pontos ao invés de dois, além da diferença no ajuste do plano pelo método de mínimos quadrados, pois as equações da reta e do plano exigem ajustes distintos.

Após o ajuste do plano  $pl$ , um paralelepípedo é ajustado aos dados para que possa ser definido o centroide  $(B_{l,cx}, B_{l,cy}, B_{l,cz})$  e as dimensões  $(H_{B_l}, W_{B_l}, D_{B_l})$  da caixa delimitadora  $B_l$ . O paralelepípedo em questão é ajustado paralelamente ao plano do piso, sendo rotacionado para alinhar-se com a direção do plano  $pl$ .

Após definir o centroide  $(B_{l,cx}, B_{l,cy}, B_{l,cz})$ , é calculado o plano  $pt$  ortogonal transversal ao plano da escada  $pl$  e que passa pelo centroide  $(B_{l,cx}, B_{l,cy}, B_{l,cz})$ . O plano em questão pode ser observado na figura 3.18.

Definindo-se o plano ortogonal transversal, os pontos 3D em  $E_{cloud}$  são projetados neste plano. A projeção produz agrupamentos distintos de pontos que representam cada borda da escada. Desta maneira, ao calcular a média das diferenças em altura e profundidade entre os grupos de pontos consecutivos, são obtidas a altura média  $h_S$  e profundidade média  $d_S$  dos degraus da escada.

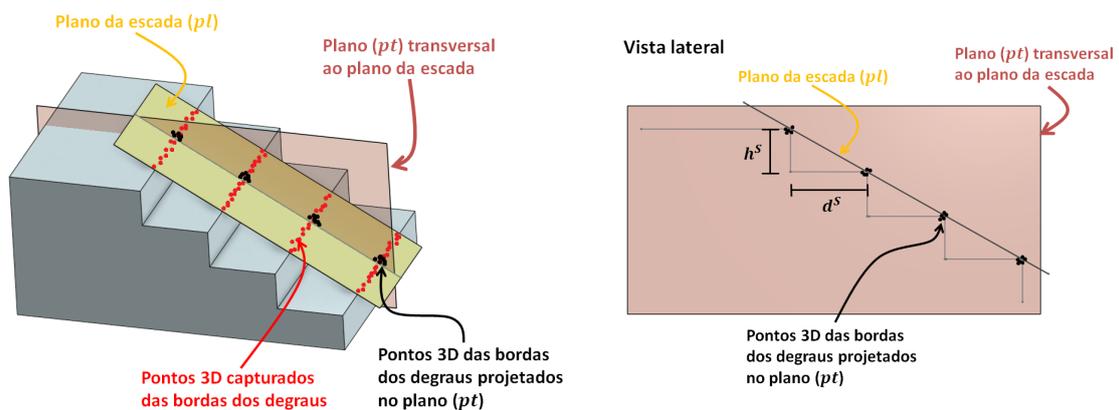


Figura 3.18: As figuras ilustram o plano  $pt$  transversal ortogonal ao plano  $pl$  e os pontos 3D das bordas dos degraus projetados no plano  $pt$ .

A seguir é apresentado o pseudocódigo do método descrito:

---

**Pseudocódigo 4** Modelagem e localização da escada utilizando imagem 3D densa

---

- 1: Inicializar um *point cloud*  $E_{cloud}$  com todos os pontos 3D ( $P'_{cloud}$ ) das capturas realizadas.
  - 2: Ajustar um plano  $pl$  aos dados  $E_{cloud}$  utilizando o *Ransac*.
  - 3: Calcular o ângulo diedral  $\alpha_S$  do plano  $pl$  relativo ao plano do solo.
  - 4: Remover os *outliers* de  $E_{cloud}$  utilizando o método *Ransac* aplicado anteriormente.
  - 5: Ajustar uma caixa delimitadora  $B_l$  aos dados em  $E_{cloud}$  e obter as dimensões ( $H_{B_l}, W_{B_l}, D_{B_l}$ ) e o centroide ( $B_{l,cx}, B_{l,cy}, B_{l,cz}$ ).
  - 6: Projetar os pontos  $E_{cloud}$  no plano  $p_t$  transversal e ortogonal a  $p_l$  e que passa pelo centroide ( $B_{l,cx}, B_{l,cy}, B_{l,cz}$ ).
  - 7: Armazenar em  $C$  os grupos de pontos projetados no passo anterior.
  - 8: Encontre os centroides de cada grupo de pontos em  $C$ .
  - 9: Arranjar os grupos de pontos  $C$  em ordem crescente de altura.
  - 10: Obter as dimensões dos degraus ( $h_S, d_S$ ), através da média das diferenças em altura e profundidade entre os centroides dos grupos de pontos consecutivos em  $C$
- 

Por fim, após obter dados do modelo da escada como localização do centroide ( $B_{l,cx}, B_{l,cy}, B_{l,cz}$ ), dimensões de altura ( $h_S$ ) e profundidade ( $d_S$ ) dos degraus, além do ângulo de inclinação da escada ( $\alpha_S$ ), a sua orientação é definida projetando o vetor normal ao plano  $pl$  da escada no plano do solo (plano  $XY$ ).

### 3.2.3 Simulações e experimentos

O método de detecção e modelagem de escadas baseado em imagem 3D densa foi validado através de experimentos reais utilizando o sensor *Kinect*. Tanto o algoritmo de detecção das bordas dos degraus, quanto o algoritmo de modelagem e localização da escada foram implementados no software *Matlab*, que apresenta diversas ferramentas para processamento de imagem e dados.

Os sistemas de coordenadas utilizados nos testes estão representados na figura 3.19, de modo que  $O_K$  representa o sistema de coordenadas do sensor *Kinect* e  $O_B$  define o sistema de coordenadas inercial.

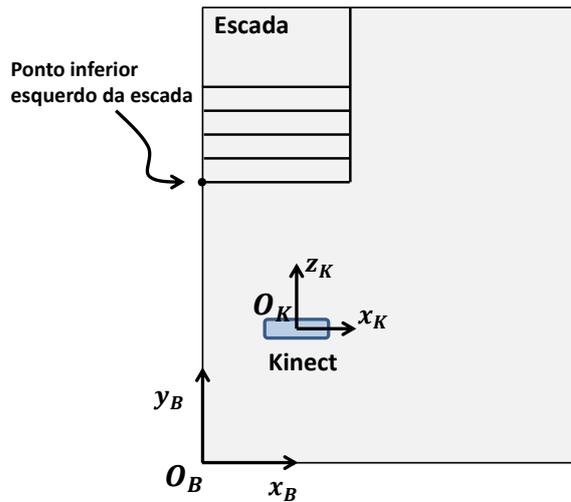


Figura 3.19: Vista superior do posicionamento dos sistemas de coordenadas inercial  $O_B$  e sistema de coordenadas do *Kinect*  $O_K$  em relação à escada

A escada detectada apresenta 5 degraus com altura ( $h_S = 0.17$  m) e profundidade ( $d_S = 0.30$  m). As dimensões totais da escada são: altura ( $H_S = 0.85$  m), largura ( $W_S = 1.2$  m) e profundidade ( $D_S = 1.68$  m). A posição do ponto inferior à esquerda da escada medido em relação ao sistema de coordenadas  $O_B$  foi definida  $((p_{1s})_B = [0 \ 6 \ 0]$  m). A escada apresenta orientação  $\phi_S = 90^\circ$  medida em relação ao eixo  $x_B$ .

O sensor *Kinect* foi posicionado a uma distância em profundidade de 3.28 m (direção  $-y_B$ ) e distância lateral de 0.62 m (direção  $x_B$ ), ambas medidas em relação ao ponto inferior esquerdo da escada (vide figura 3.19). A altura do sensor em relação ao solo é de 0.55 m.

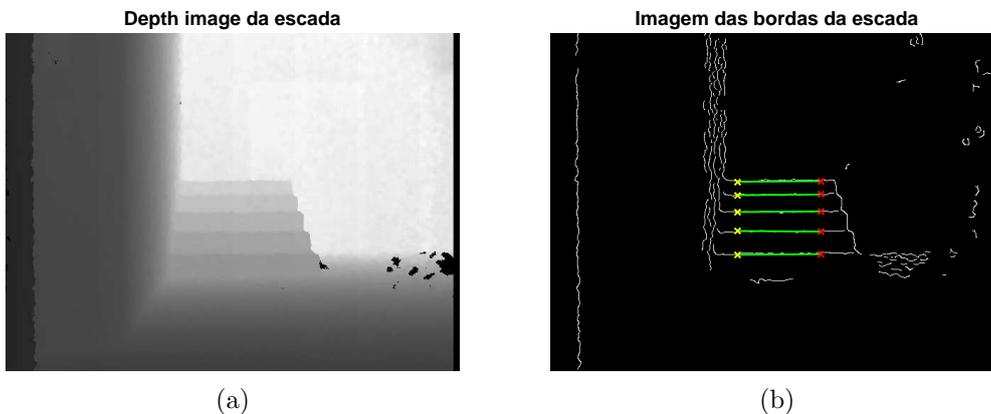


Figura 3.20: (a) apresenta a *depth image* da escada e (b) apresenta a imagem das bordas com as linhas da escada capturadas. Ambas as imagens foram capturadas no experimento com o *Kinect* posicionado a 3.285 m em profundidade (direção  $-y_B$ ) e a 0.62 m lateralmente (direção  $x_B$ ) medidos em relação ao ponto inferior esquerdo da escada.

Foram realizadas 10 capturas da escada para executar a primeira etapa do método, de modo que uma delas é demonstrada através da figura 3.20. O quadro à esquerda apresenta uma das *depth image* capturadas e o quadro à direita apresenta o resultado da detecção de bordas e obtenção das linhas referentes às bordas dos degraus. Os parâmetros utilizados para detecção das bordas pelo método de *Canny* foram:  $[0.07 \ 0.12]$ ; e os parâmetros do *Probabilistic Hough Transform* foram:  $\Delta r_h = 1$  pixel,  $\Delta \theta_h = \pi/180$  rad,  $th_h = 15$ . Onde a)  $\Delta r_h$ : é a resolução do parâmetro de *Hough*  $r_h$  em pixel; b)  $\Delta \theta_h$ : é a resolução do parâmetro de *Hough*  $\theta_h$  em radianos; c)  $th_h$ : é o número mínimo de pontos para detectar a linha.

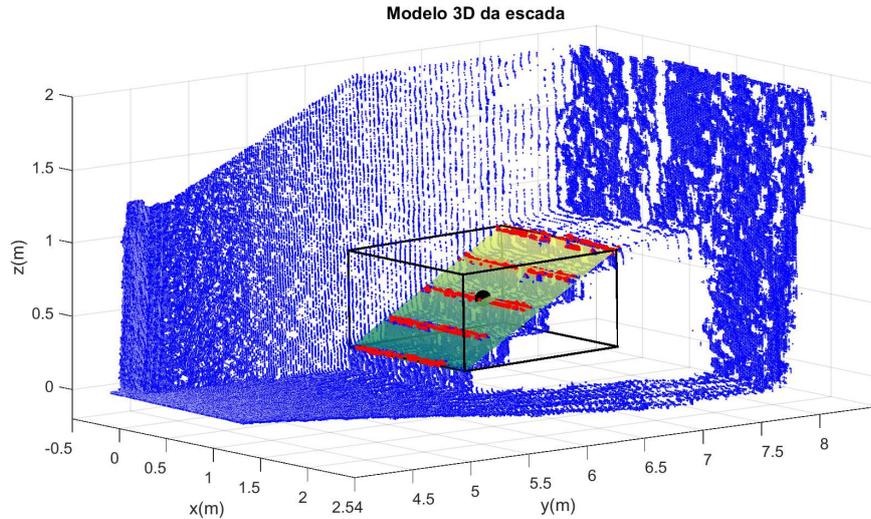


Figura 3.21: Modelagem 3D da escada realizada experimentalmente com o *Kinect* posicionado a 3.285 m em profundidade (direção  $-y_B$ ) e a 0.62 m lateralmente (direção  $y_B$ ) medidos em relação ao ponto inferior esquerdo da escada.

A figura 3.21 ilustra o modelo 3D da escada obtido aplicando-se a segunda etapa do método. Em vermelho observam-se os pontos 3D capturados das bordas dos degraus, já a caixa delimitadora é definida pelas linhas pretas, também em preto está o centroide da caixa delimitadora representado por uma esfera. É possível observar ainda o plano inclinado definido pelas bordas dos degraus.

Os parâmetros principais utilizados nesta etapa estão relacionados ao ajuste do plano aos dados através do método *Ransac*. Desta maneira, foi estabelecido  $n_{iter} = 150$  e  $th_r = 0.5 \cdot (\text{número de pontos 3D das bordas})$ , onde  $n_{iter}$  é o número de iterações realizadas pelo método e  $th_r$  é o número mínimo de pontos para ser considerado um modelo válido. Já o erro máximo para um ponto ser considerado pertencente ao modelo foi definido como  $e_{rmax} = 0.03$  m.

Os resultados obtidos para este experimento são demonstrados na tabela abaixo:

	<b>Experimental</b>	<b>Real</b>
Altura média dos degraus ( $h_S$ [m])	0.16	0.17
Profundidade média dos degraus ( $d_S$ [m])	0.32	0.30
Ângulo de inclinação do plano da escada ( $\alpha_S$ [graus])	26.7°	29.5°
Centroide ( $[B_{l,cx}, B_{l,cy}, B_{l,cz}]$ [m]) da caixa delimitadora	[0.64, 6.61, 0.49]	
Dimensões ( $[H_{B_i}, W_{B_i}, D_{B_i}]$ [m]) da caixa delimitadora	[0.66, 1.23, 1.31]	

Tabela 3.2: Resultados experimentais para detecção e modelagem de escadas utilizando sensor 3D

Como pode ser observado na tabela 3.2, os erros de medida da altura e profundidade dos degraus foram 0.01 m (5.9%) e 0.02 m (6.6%) respectivamente, o que indica um bom resultado em virtude dos ruídos presentes nas medidas do sensor *Kinect*. O erro do ângulo de inclinação da escada foi de 2.8° (9.5%), fato que demonstra também uma boa estimativa realizada pelo método.

A tabela 3.3 apresenta a estimativa de posição e orientação da escada através do método.

	<b>Experimental</b>	<b>Real</b>
$(p_{1s})_B$ [m]	[0.02 6 0]	[0 6 0]
$\phi_S$ [graus]	88°	90°

Tabela 3.3: Estimativa da posição e orientação da escada através do método de detecção e modelagem de escadas utilizando sensor 3D

Desta maneira, o erro de estimativa da posição foi de 0.02 m (0.3%) em módulo e o erro de orientação foi 2°(2.2%).

Para avaliar a robustez do método a variações de orientação do sensor em relação à escada, um novo experimento foi realizado com a orientação do *Kinect* defasada em 120° graus medidos em relação ao eixo  $x_B$ . O sensor foi posicionado a uma distância em profundidade (direção  $-y_B$ ) de 2.74 m, distância lateral (direção  $x_B$ ) de 2.27 m, ambas medidas em relação ao ponto inferior esquerdo da escada. A altura do sensor em relação ao solo é de 0.26 m.

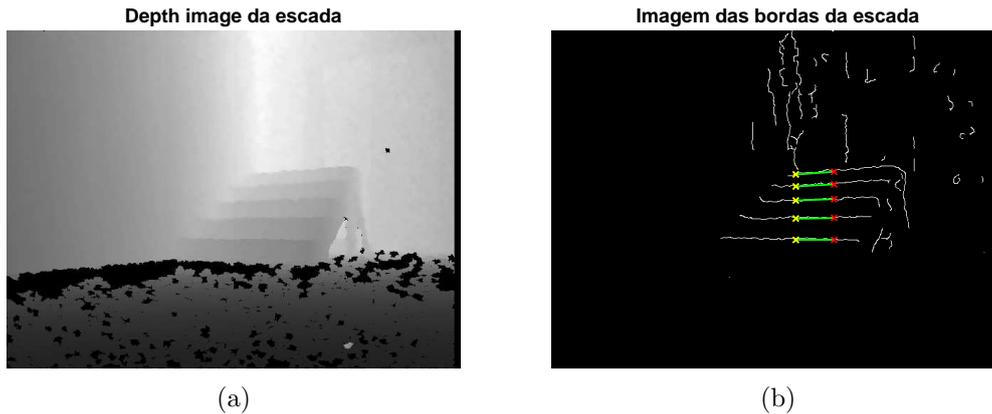


Figura 3.22: (a) apresenta a *depth image* de escada e (b) apresenta a imagem das bordas com as linhas da escada capturadas. Ambas as imagens foram capturadas no experimento utilizando o *Kinect* defasado por  $120^\circ$  em orientação, posicionado a 2.74 m em profundidade (direção  $-y_B$ ) e a 2.27 m lateralmente (direção  $x_B$ ) medidos em relação ao ponto inferior mais à esquerda da escada.

Assim como no primeiro experimento, foram realizadas 10 capturas da escada para executar a primeira etapa do método, de modo que uma delas é demonstrada através da figura 3.22. O quadro à esquerda apresenta uma das *depth image* capturadas, em que é possível observar o ruído produzido pelo sensor, e o quadro à direita apresenta o resultado da detecção de bordas e a extração das linhas referentes às bordas dos degraus.

Os mesmos parâmetros do método de *Canny* e *Probabilistic Hough Transform* mencionados anteriormente foram utilizados neste experimento.

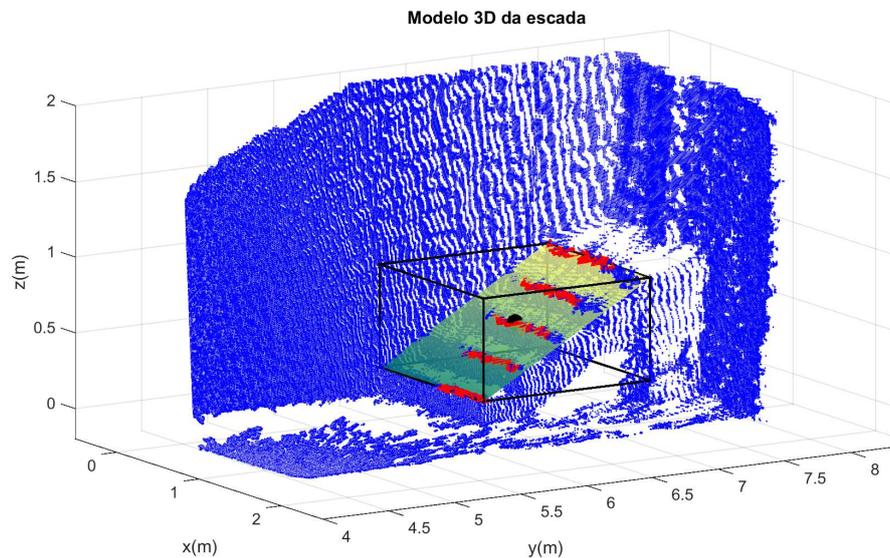


Figura 3.23: Modelagem 3D da escada realizada experimentalmente com o *Kinect* defasado por  $120^\circ$  graus em orientação, posicionado a 2.74 m em profundidade (direção  $-y_B$ ) e a 2.27 m lateralmente (direção  $x_B$ ) medidos em relação ao ponto inferior mais à esquerda da escada.

O modelo 3D da escada obtido aplicando-se a segunda etapa do método é demonstrado na figura 3.23. Novamente foram utilizados os mesmos parâmetros do

primeiro experimento.

Os resultados obtidos são demonstrados na tabela abaixo:

	<b>Experimental</b>	<b>Real</b>
Altura média dos degraus ( $h_S$ [m])	0.16	0.17
Profundidade média dos degraus ( $d_S$ [m])	0.31	0.30
Ângulo de inclinação do plano da escada ( $\alpha_S$ [graus])	28.2°	29.5°
Centroide ( $[B_{l,cx}, B_{l,cy}, B_{l,cz}]$ [m]) da caixa delimitadora	[0.64, 6.62, 0.48]	
Dimensões ( $[H_{B_l}, W_{B_l}, D_{B_l}]$ [m]) da caixa delimitadora	[0.68, 1.27, 1.26]	

Tabela 3.4: Resultados do segundo experimento do método de detecção e modelagem de escadas utilizando sensor 3D

A partir da tabela 3.4, temos que os erros de medida da altura e profundidade dos degraus foram de 0.01 m (5.9%) e 0.01 m (3.3%), respectivamente, o que indica novamente um bom resultado, tendo em vista a presença de ruídos do sensor *Kinect*. O erro do ângulo de inclinação da escada foi de 1.3° (4.4%), comprovando assim a boa estimativa realizada pelo método e sua robustez a variações de orientação do sensor em relação à escada.

A tabela 3.5 apresenta a estimativa de posição e orientação da escada através do método.

	<b>Experimental</b>	<b>Real</b>
$(p_{1s})_B$ [m]	[0 5.99 0]	[0 6 0]
$\phi_S$ [graus]	89°	90°

Tabela 3.5: Estimativa da posição e orientação da escada através do método de detecção e modelagem de escadas utilizando sensor 3D

Portanto, os erros de estimativa da posição da escada e da sua orientação foram de 0.01 m (0.17%) em módulo e 1° (1.1%), respectivamente.

### 3.2.4 Conclusões

Portanto, neste capítulo foram apresentadas duas abordagens para detecção e modelagem de escadas, sendo a primeira baseada nos sensores *laser scan* e a segunda baseada em sensores 3D como *Kinect* e câmeras RGB-D.

Ambas as metodologias foram descritas e validadas através de experimentos reais.

O método baseado em sensores *lasers* é capaz de realizar a detecção e localização da escada com bons resultados, entretanto o método apresenta falhas quando existem ruídos nos dados dos sensores ou quando a distância do sensor à escada é elevada.

Para o método de detecção e modelagem baseado em imagem 3D densa, os resultados obtidos durante os experimentos foram satisfatórios, tendo em vista o

nível dos ruídos do sensor *Kinect*. O método foi capaz ainda de detectar e modelar a escada com a orientação do sensor defasada em relação à orientação da escada. Desta maneira, foi comprovada a eficácia do método para a solução do problema.

# Capítulo 4

## Método de localização em mapa conhecido

Em aplicações robóticas de exploração e resgate, a tarefa de localização é utilizada para determinar a posição dos robôs durante a operação, mapear o ambiente e seus obstáculos, fornecer dados ao controle do robô em rastreamento de trajetórias, além de permitir armazenar todo o percurso realizado durante uma tarefa de exploração, por exemplo.

Para conduzir sistemas robóticos de uma posição inicial até uma posição desejada no ambiente, é necessário definir ferramentas que permitam estimar sua posição e orientação em relação a um referencial inercial, fornecendo dados para a operação e controle do robô.

Obter a estimativa da posição e orientação do robô em relação a um referencial inercial, a partir das informações dos sensores, é o que se denomina localização. Quando aplicado aos robôs móveis, o problema de localização consiste basicamente em estimar as coordenadas cartesianas  $(x, y, z)$  e os ângulos de orientação (*roll*- $\varphi$ , *pitch*- $\theta$ , *yaw*- $\phi$ ) dos robôs em relação a um dado mapa do ambiente com sistema de coordenadas fixo  $O_B$ . Ressalta-se que existem diferentes representações da orientação do robô além da *roll-pitch-yaw*, como a representação em ângulos de Euler, por exemplo.

Quando o deslocamento dos robôs móveis se restringe ao plano 2D do ambiente, o problema de localização é reduzido para a estimativa das coordenadas  $(x, y)$  e ângulo de orientação (*yaw*- $\phi$ ) do robô em relação ao sistema de coordenadas inercial  $O_B$ .

Uma das soluções clássicas para o problema de localização é conhecida como *dead reckoning*. O *dead reckoning* permite estimar a posição atual do robô, a partir da estimativa de posição anterior e medida de velocidade em um dado intervalo de tempo. Desta maneira, ao integrar a velocidade no intervalo de tempo medido, obtêm-se a estimativa do deslocamento realizado e, conseqüentemente, da posição e

orientação finais no instante seguinte.

Em aplicações com robôs móveis, é comum a utilização de sensores como *encoders* para medir o número de rotações efetuadas pelas rodas e inferir o deslocamento realizado. A estimativa da posição e orientação a partir de sensores como o *encoders* também é conhecida como localização odométrica. Outros tipos de sensores podem ser adotados para inferir o deslocamento efetuado e, conseqüentemente, a posição do robô, entretanto realizar o *dead reckoning* a partir da odometria produz erros acumulativos na estimativa de posição e orientação, demonstrando uma fraqueza do método.

Os ruídos de medidas dos sensores, imprecisões no modelo como medida incorreta do raio das rodas, além dos escorregamentos entre rodas e terreno, produzem erros que se acumulam com o passar do tempo de operação, gerando grande imprecisão na medida de posição e orientação em relação aos respectivos valores reais.

Uma das formas de redução dos erros acumulativos produzidos pelo *dead reckoning* consiste em utilizar sensores para obter medidas de *landmarks* com posição conhecida. Os *landmarks* são pontos de referência como paredes, portas, quinas, colunas, ou até mesmo objetos artificiais posicionados no ambiente com o propósito específico de auxiliar na localização de sistemas robóticos.

A identificação de *landmarks* pode ser realizada através de sensores como *lasers scan*, câmeras monoculares, sensores 3D (*Kinect*), sonares, dentre outros.

Para obter a estimativa de posição e orientação do robô, as medidas dos *landmarks* calculadas através dos sensores devem ser unidas aos dados da odometria. Existem diversas abordagens utilizadas para unir os dados referidos, entretanto dois dos principais métodos aplicados são: o Filtro de Kalman Estendido (EKF) e o *Monte Carlo Localization (MCL)*. Ambos os métodos são baseados em uma análise probabilística do problema, de modo que a diferença básica entre as duas abordagens está na forma como a função densidade de probabilidade do estado do sistema (posição e orientação) é representada.

Como já mencionado, o problema de localização definido em (Thrun et al. 2005) consiste na realização da estimativa de posição e orientação em relação a um mapa do ambiente. O mapa em questão pode ser conhecido a priori ou ser construído simultaneamente à localização, o que se denomina *Simultaneous Localization and Mapping (SLAM)*. Para os dois tipos de problemas, mapa conhecido ou desconhecido a priori, é necessário definir qual o tipo de representação do mapa a ser utilizada. Por conseguinte, existem duas principais abordagens, a representação métrica e a representação topológica.

No mapeamento métrico, o ambiente é representado por um conjunto de coordenadas no espaço 2D, desta maneira os *landmarks*, objetos e obstáculos do ambiente são representados geometricamente por linhas, círculos ou pontos com posição bem

definida. Já a representação topológica pode ser vista como uma representação semelhante aos grafos, onde os nós no grafo indicam os *landmarks* ou pontos significativos do ambiente, e os links entre os nós são definidos pelas informações das trajetórias que ligam os locais relevantes.

## 4.1 Metodologia aplicada

A tarefa principal de localização nesta dissertação consiste em estimar a posição  $(x, y)$  e orientação  $(\phi)$  de um robô móvel com esteiras do tipo *EOD*, em relação ao sistema de coordenadas fixo  $O_B$ . A posição inicial do robô é conhecida, de modo que o método deve ser capaz de estimar a localização do sistema robótico, do início do deslocamento até a posição próxima à escada.

Além da tarefa principal, existe ainda a tarefa secundária de localização durante a locomoção sobre a escada. Esta, por sua vez, consiste em estimar a posição  $(x)$  e orientação  $(\phi)$  do robô em relação ao sistema de coordenadas da escada  $O_S$ .

Desta maneira, a localização neste trabalho é dividida em duas subtarefas: localização durante o deslocamento até a escada e localização durante a locomoção sobre a escada.

Como já definido no capítulo 2, o ambiente no qual o robô está inserido é característico do interior de prédios com formatos geométricos lineares dispostos ortogonalmente como portas, paredes e quinas.

Definiu-se o deslocamento até a escada sendo restrito ao plano do solo, ou seja, plano  $XY$  do sistema  $O_B$ . Já para locomoção sobre a escada, o movimento do robô passa a ser restrito ao plano inclinado  $pl$  definido pelas linhas dos degraus da escada, ou seja, o plano  $XY$  do sistema  $O_S$ .

O mapa do ambiente foi estabelecido através da representação métrica, de modo que os *landmarks* correspondem às paredes e portas, sendo descritos por linhas com coordenadas polares  $(\rho_i, \theta_i)$  e comprimento  $l_i$ , onde o índice  $i$  representa a  $i$ -ésima linha do mapa.

Com o intuito de realizar as medidas de posição e orientação, o robô dispõe da odometria e do *laser scan* posicionado horizontalmente.

Desta maneira, a partir da definição do problema de localização e das hipóteses aplicadas ao ambiente, robô e interação entre ambos, foram definidas as duas metodologias de localização utilizadas neste trabalho.

## 4.2 Localização durante a aproximação até a escada

A etapa de localização durante a aproximação até a escada é responsável por estimar as coordenadas cartesianas  $(x, y)$  e orientação  $(\phi)$  do robô em relação ao sistema de coordenadas inercial  $O_B$ . A figura 4.1 ilustra a localização do robô, apresentando os sistemas de coordenadas inercial da base  $O_B$ , sistema de coordenadas do robô  $O_R$  e do sensor *laser*  $O_{LH}$ .

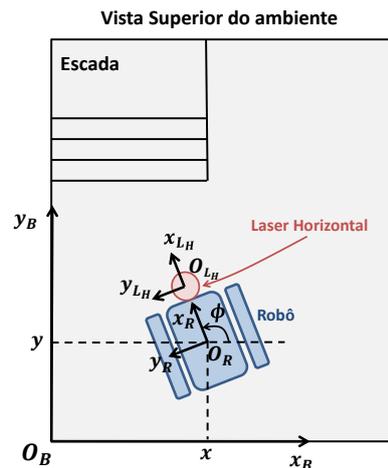


Figura 4.1: Vista superior do robô posicionado no ambiente durante a etapa de localização para a aproximação até a escada

A primeira etapa de localização utiliza o Filtro de Kalman Estendido (EKF) (Brown & Hwang 1997) para realizar a fusão de duas medidas de posição, uma calculada através da odometria e outra obtida a partir do sensor *laser scan* posicionado horizontalmente.

Para calcular a posição do robô a partir do *laser*, primeiramente são extraídas as linhas do ambiente aplicando-se o método *Ransac* (Fischler & Bolles 1981) aos dados do sensor. Ressalta-se que o método *Ransac* foi escolhido em virtude da sua robustez a ruídos nos dados e rapidez, visto que não são necessários todos os pontos medidos no escaneamento para estimar o modelo.

Após realizar a transformação de posição das linhas extraídas do sistema de coordenadas do sensor ( $O_{LH}$ ) para o sistema de coordenadas inercial da base ( $O_B$ ), é observado o deslocamento de posição e orientação entre as linhas medidas e as linhas representadas no mapa. Esta diferença ocorre devido ao movimento de translação e rotação realizado pelo robô. Desta maneira, torna-se necessário solucionar o problema de associação entre os dados do *laser* e do mapa, para então obter as medidas de posição e orientação a partir do referido sensor.

Foi proposto um método semelhante ao utilizado em Armesto & Tornero (2006) para associação de linhas (*landmarks*) e obtenção da medida de posição e orientação.

Este tipo de tarefa também é conhecida como *matching* de *landmarks*. A abordagem aplicada define uma medida de erro em distância e orientação entre linhas para associá-las. Em seguida, realizando duas minimizações, uma para o erro de orientação e outra para o erro de translação entre as linhas, são calculados o deslocamento realizado pelo robô e a sua rotação, no intervalo de tempo entre dois escaneamentos do *laser*.

Por fim, o Filtro de Kalman Estendido (EKF) é responsável por obter a melhor estimativa da posição, utilizando os dados da odometria e a medida de posição calculada a partir do *laser*. Deve ser ressaltado que a palavra melhor neste contexto consiste em obter uma estimativa da posição e orientação do robô com incerteza inferior às incertezas da odometria e medida de posição a partir do *laser*.

#### 4.2.1 Método de extração de linhas

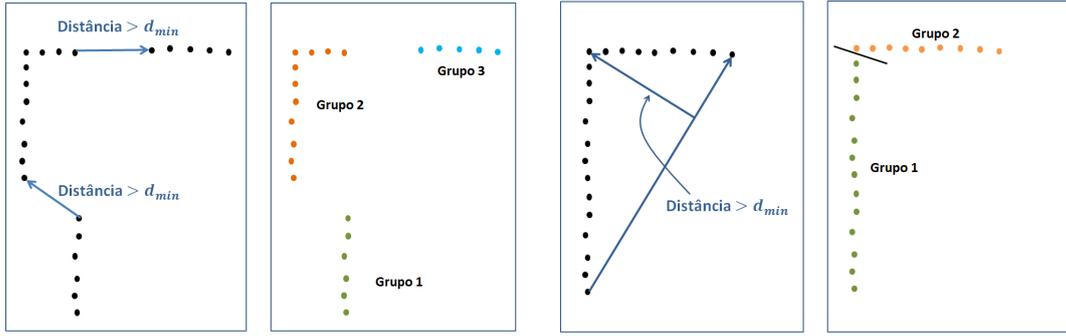
O *laser* horizontal é utilizado para calcular as medidas de posição e orientação do robô, desta maneira é necessário definir um método para extrair as linhas utilizando os dados do *laser* e associá-las aos dados do mapa, para então obter as medidas de posição e orientação do robô fornecidas pelo sensor.

Ao realizar o escaneamento com o *laser scan*, o robô armazena um conjunto de pontos 2D que representam o ambiente a sua volta. Antes de efetuar a extração de linhas propriamente dita, são realizadas duas segmentações dos pontos 2D, da mesma maneira como apresentado na seção 3.1.2.

A primeira etapa da segmentação separa os dados de acordo com a distância entre pontos consecutivos do escaneamento, ou seja, as distâncias entre pontos consecutivos é avaliada do primeiro ao último dado da varredura, se a distância entre esses pontos for maior que um limite máximo, então os dados são divididos em dois grupos distintos (*clusters*). O método prossegue gerando novos *clusters* caso a condição seja satisfeita, até o último ponto do escaneamento.

Em seguida, a segunda etapa da segmentação é aplicada a cada *cluster* encontrado anteriormente. Deste modo, para cada ponto do *cluster* é verificado se a distância do ponto à reta que une os pontos extremos do *cluster* é maior que um limite máximo. Caso positivo, o grupo de pontos é dividido em dois grupos distintos e o método é aplicado novamente a esses dois *clusters*, prosseguindo recursivamente até que nenhum novo *cluster* seja encontrado.

A figura 4.2 ilustra novamente as segmentações realizadas.



(a) Segmentação baseada na distância entre pontos consecutivos (b) Segmentação baseada na distância do ponto à linha formada por pontos extremos

Figura 4.2: Segmentações aplicadas no método de extração de linhas durante a localização

Após a etapa de segmentação dos dados do *laser*, aplica-se o método *Ransac* (Fischler & Bolles 1981) para extração das linhas do ambiente. Desta maneira, o mesmo algoritmo *Ransac* aplicado na detecção de escadas (seção 3.1.2 - pseudocódigo 2) foi destinado a esta etapa. A diferença básica entre os métodos *Ransac* aplicados está na parametrização final das linhas que, para o caso em questão, é realizada através das coordenadas polares  $(\rho_{li}, \theta_{li})$ .

Além dos parâmetros mencionados, são armazenados ainda o comprimento de cada linha  $(l_i)$  e o vetor unitário  $(u_{li})$  que representa a direção da linha extraída. A figura 4.3 demonstra os parâmetros das linhas extraídas necessários para execução do *matching* de *landmarks*.

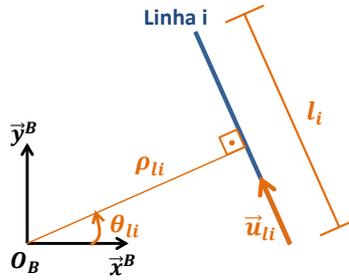


Figura 4.3: Parâmetros da linha extraída através do método *Ransac*

É importante ressaltar que para o correto funcionamento do método de *matching* de *landmarks* descrito na próxima seção, é necessário que os vetores  $(u_{li})$  sejam calculados sempre no mesmo sentido, pois a associação entre as linhas extraídas e as linhas do mapa depende diretamente do sentido dos referidos vetores. Portanto, o sentido do vetor  $u_{li}$  é definido sempre à direita da reta que liga a origem do *laser*  $O_{LH}$  à linha  $i$  representada pelo vetor  $u_{li}$ .

A seguir são apresentados os parâmetros e o pseudocódigo 5 do método de extração de linhas utilizado.

- a)  $N$ : Número de grupos de pontos encontrados após a segmentação;

- b)  $n_{iter}$ : Número de iterações realizadas pelo algoritmo;
- c)  $cl_i$ :  $i$ -ésimo grupo de pontos segmentado;
- d)  $(\rho_0, \theta_0)$ : Parâmetros do modelo da reta inicial ( $\rho = \cos(\theta_0)x + \sin(\theta_0)y$ );
- e)  $d_j$ : Distância do  $j$ -ésimo ponto à reta;
- f)  $d_{thres}$ : Distância máxima do ponto à reta para considerá-lo pertencente ao modelo;
- g)  $pts_{in}$ : Grupo de pontos que satisfazem a distância máxima à reta modelo;
- h)  $n_{in}$ : Número de pontos pertencentes a  $pts_{in}$
- i)  $n_{thres}$ : Número mínimo de pontos que devem pertencer ao modelo para considerá-lo válido;
- j)  $(\rho_{md}, \theta_{md})$ : Melhor modelo de reta ajustado pelo método de mínimos quadrados.

---

### Pseudocódigo 5 Extração de linhas utilizando *Ransac*

---

```

1: Para cada grupo de pontos segmentado ( $cl_0, cl_1, \dots, cl_N$ )
2: for  $k = 1 : n_{iter}$  do
3:   Inicialize o modelo  $(\rho_0, \theta_0)$  da reta utilizando dois pontos aleatórios.
4:   Para cada ponto pertencente a  $cl_i$ , calcule a distância  $d_j$  do ponto à reta modelo.
5:   Armazenar os pontos pertencentes ao grupo  $cl_i$  tal que  $d_j < d_{thres}$ .
6:   O resultado do passo anterior é o cálculo de  $pts_{in}$  e  $n_{in}$ .
7:   if ( $n_{in} > n_{thres}$  and  $n_{in} > n_{md}$ ) then
8:     Calcular novo modelo  $(\rho_{md}, \theta_{md})$  através do método de mínimos quadrados, utilizando
       os  $pts_{in}$ .
9:      $n_{md} = n_{in}$ 
10:   end if
11: end for
12: return  $\rho_{md}, \theta_{md}$ .

```

---

Finalizada a etapa de extração das linhas do ambiente e parametrização das mesmas, aplica-se o método de *matching* de *landmarks* para estimar a transformação entre os dados medidos e as informações do mapa, inferindo, conseqüentemente, a translação e rotação realizadas pelo robô.

#### 4.2.2 Matching de *landmarks*

O problema de associação de dados pode ser visto como um problema de transformação de coordenadas, de modo que o objetivo é encontrar qual a rotação e translação que aplicadas ao robô, permitem alinhar os *landmarks* medidos com os *landmarks* do mapa.

Quando o robô se desloca no ambiente, as varreduras do *laser* também sofrem as transformações de translação e rotação realizadas pelo robô. Estas transformações

são facilmente observadas no plano cartesiano. Assim, a figura 4.4 ilustra as transformações do escaneamento e dos *landmarks* observados, após o deslocamento do robô.

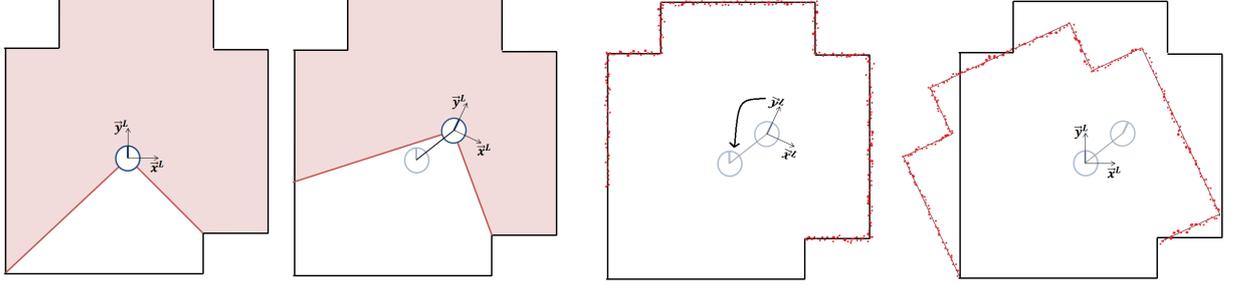


Figura 4.4: A primeira imagem demonstra o robô na posição inicial. Após o robô realizar uma translação e rotação, o *laser* passa a capturar pontos diferentes do ambiente como é demonstrado na segunda ilustração, já na terceira figura, observam-se os pontos capturados do *laser*, por fim a última ilustração demonstra a transformação dos dados do sistema de coordenadas do *laser*, após o deslocamento do robô, para o sistema de coordenadas do mesmo antes do deslocamento.

Como pôde ser observado na figura 4.4, existe uma rotação e translação que aplicadas às linhas extraídas do *laser* fazem com que as mesmas sejam alinhadas aos pontos de referência do mapa.

A abordagem proposta para solucionar o problema de *matching* de dados é fundamentada na minimização de duas funções custo com o intuito de estimar a rotação e translação do robô.

Para cada função custo é definida uma função de erro utilizada a priori somente para associar os *landmarks* medidos com os *landmarks* do mapa.

O método é dividido em duas etapas: a primeira, responsável por estimar a rotação entre as linhas e a segunda etapa que estima a translação entre as mesmas.

### Primeira etapa: Estimativa da rotação entre os *landmarks*

Tendo em vista o objetivo desta etapa que consiste em calcular a rotação que alinha os *landmarks*, ao avaliar a geometria do problema, foi definida a seguinte função erro para a associação entre as linhas extraídas e as linhas do mapa:

$$e_{rij} = \|p_{ai} - q_{aj}\| + \|p_{bi} - q_{bj}\| + \eta_{rot} |\arccos(u_{li}^{dT} u_{lj}^m)| \quad (4.1)$$

onde  $\eta_{rot}$  é a constante que penaliza o erro de orientação entre as linhas, os pares  $p_{ai}$  e  $p_{bi}$  são os pontos 3D inicial e final da linha extraída  $i$ , já os pares  $q_{aj}$  e  $q_{bj}$  são os pontos 3D inicial e final da linha  $j$  pertencente ao mapa. Os índices  $()^d$  e  $()^m$ , indicam as linhas extraídas e as linhas do mapa, respectivamente. Assim,  $u_{li}^d \in \mathbb{R}^3$  e  $u_{lj}^m \in \mathbb{R}^3$  são os vetores unitários que representam as direções das referidas linhas.

**Observação 4.1** Deve ser ressaltado que apesar dos pontos e vetores serem representados no espaço 3D, a função erro  $e_{rij}$  de associação de landmarks foi avaliada apenas para linhas em um mesmo plano, visto que o problema de localização definido nesta dissertação considera a estimativa de posição e orientação do robô somente no plano 2D do ambiente.

A função  $e_{rij}$  penaliza com maior intensidade o erro de orientação entre as linhas, sendo representado pelo último termo da equação (4.1), enquanto que atribui menor peso ao erro da distância entre as linhas, sendo definido pelos dois primeiros termos da equação. Isto se deve ao interesse, nesta primeira etapa, em associar as linhas com orientações mais próximas o possível.

Desta maneira, a função  $e_{rij}$  é aplicada a todas as combinações entre as linhas extraídas e linhas do mapa, de modo que para um dado *landmark* apenas a associação com menor erro é validada. A figura 4.5 exemplifica como a rotação influencia na associação entre os *landmarks*.

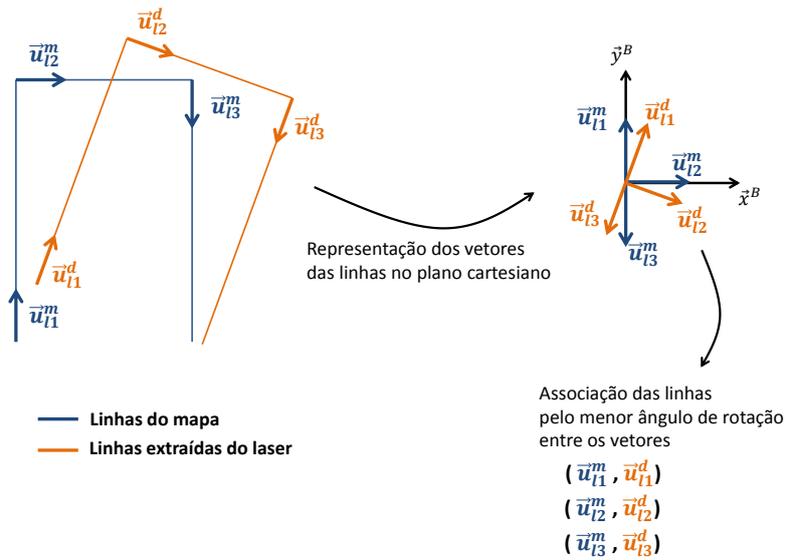


Figura 4.5: A figura ilustra como ocorre a associação entre as linhas extraídas e as linhas mapeadas através da relação de rotação entre seus vetores direcionais.

Como já mencionado, os vetores  $u_{li}^d$  devem ser calculados sempre no sentido à direita da reta que liga a origem  $O_{LH}$  do *laser* à linha  $i$ . Esta imposição é extremamente importante para que haja uma associação correta entre as linhas, visto que a inversão de um dos vetores  $u_{l1}^d, u_{l2}^d$  ou  $u_{l3}^d$ , representados na figura 4.5, provoca um aumento considerável no erro de associação  $e_{rij}$  com aos vetores  $u_{l1}^m, u_{l2}^m$  ou  $u_{l3}^m$ , devido ao aumento do ângulo entre os vetores associados. Fato que produz erros incorretos e conseqüentemente prejudica a associação dos dados.

Após definir a associação correta entre os *landmarks* medidos com os respectivos pertencentes ao mapa, a função custo  $E_{\phi}(R_i)$  define o erro de orientação entre os

vetores  $u_{li}^d$  e  $u_{lj}^m$  associados.

$$E_\phi(R_l) = \sum_{i=1}^{N_l} \|u_{li}^m - R_l u_{li}^d\|^2 \quad (4.2)$$

onde  $N_l$  é o número de linhas associadas e  $R_l \in SO(3)$  é a matriz de rotação elementar ao redor do eixo  $z_B$ .

Desta maneira, para alinhar as direções das linhas extraídas com o mapa, basta encontrar a matriz de rotação  $R_l$  que minimiza a função custo  $E_\phi(R_l)$ .

Para obter a matriz de rotação  $R_l$  ótima, define-se a matriz  $S_{uu^T}$  como segue:

$$S_{uu^T} = \sum_{i=1}^{n_l} u_{li}^d u_{li}^{m^T} \quad (4.3)$$

**Observação 4.2** *Durante o cálculo da matriz de rotação ótima, os vetores unitários  $u_{li}^d$  e  $u_{li}^m$  foram considerados como pertencentes ao  $\mathbb{R}^3$ , visto que o método proposto por Arun et al. (1987) permite estimar a rotação ótima que alinha vetores tridimensionais. Desta maneira, para representar o caso mais genérico, a descrição do método considera as três coordenadas, de modo que o caso planar definido nesta dissertação também pode ser solucionado pelo método em questão.*

Utilizando a decomposição em valores singulares da matriz  $S_{uu^T}$ , temos:

$$S_{uu^T} = U \Sigma V^T, \quad \text{onde } U^T U = V^T V = I \quad (4.4)$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0 \quad (4.5)$$

A partir do estudo desenvolvido em Arun et al. (1987) é possível concluir que:

$$R_l = V U^T \quad \text{se } \det(V U^T) > 0 \quad (4.6)$$

$$R_l = V J U^T \quad \text{se } \det(V U^T) < 0, \quad \text{onde } J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4.7)$$

**Observação 4.3** *O cálculo da matriz de rotação  $R_l$  ótima através da decomposição em valores singulares da matriz  $S_{uu^T}$  pode produzir duas respostas distintas, uma matriz de rotação ou uma matriz de reflexão. Ambas as matrizes são ortogonais, entretanto representam operações geométricas distintas. Quando o determinante  $\det(V U^T)$  se torna negativo,  $R_l$  representa a matriz de reflexão, impossibilitando*

o cálculo do ângulo de rotação ótimo  $\phi_l$  necessário para estimar a orientação do robô. Desta maneira, torna-se necessário inserir na resposta a matriz de correção  $J$ , transformando a matriz de reflexão na matriz de rotação ótima  $R_l = VJU^T$ .

A prova matemática do cálculo da matriz ótima  $R_l$  é apresentada no apêndice A.

Após calcular a matriz de rotação  $R_l$  ótima, é possível obter qual o ângulo  $\phi_l$  da rotação realizada pelo robô. Assim, considerando as matrizes de rotação elementar em torno do eixo  $z_B$  com o formato a seguir:

$$R_l = \begin{bmatrix} \cos(\phi_l) & -\sin(\phi_l) & 0 \\ \sin(\phi_l) & \cos(\phi_l) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Desta maneira, o ângulo  $\phi_l$  da rotação realizada pode ser definido por:

$$\phi_l = \text{atan2}(r_{21}, r_{11}) \quad (4.8)$$

onde a função  $\text{atan2}$  calcula o inverso da tangente para os quatro quadrantes.

Após estimar a matriz de rotação  $R_l$  ótima, aplica-se a rotação sobre as linhas extraídas de forma a alinhar as direções entre linhas extraídas e linhas do mapa associadas. Em seguida, é realizada a segunda etapa do *matching* de dados, a correção da translação entre os *landmarks*.

### Segunda etapa: Estimativa da translação entre os *landmarks*

Esta segunda etapa tem por objetivo estimar a translação que alinha perfeitamente os *landmarks* a partir de uma associação a ser estabelecida. Novamente são definidas uma nova função custo a ser minimizada e uma nova função de erro para determinar a associação entre os *landmarks*.

Diferentemente da função  $e_{rij}$  definida pela equação (4.1), a nova função deve penalizar com maior intensidade ambos, as distâncias entre as linhas e o erro de orientação entre suas direções.

Para simplificar a notação, os vetores apresentados a seguir serão representados no  $\mathbb{R}^2$ , pois estamos considerando a localização no plano 2D do ambiente, como já mencionado anteriormente.

Assim, a função de erro  $e_{tij}$  foi proposta como segue:

$$e_{tij} = \eta_{trans} |(v_i^{dT} v_j^m) \rho_{li}^d - \rho_{lj}^m| + \eta_{rot} |\arccos(u_{li}^{dT} u_{lj}^m)| \quad (4.9)$$

$$\text{com } v_i^d = \begin{bmatrix} \cos(\theta_{li}) \\ \sin(\theta_{li}) \end{bmatrix}, \quad v_j^m = \begin{bmatrix} \cos(\theta_{lj}) \\ \sin(\theta_{lj}) \end{bmatrix}$$

onde  $v_i^d$  e  $v_j^m$  são vetores unitários ortogonais às direções das linhas extraídas e linhas do mapa respectivamente. Os parâmetros  $\eta_{trans}$  e  $\eta_{rot}$  definem os pesos atribuídos aos erros da distância e orientação entre as linhas.

Baseando-se na equação (4.9), é possível observar que o primeiro termo é o responsável por atribuir uma medida de distância entre as linhas. Esta medida é calculada projetando-se a distância da linha extraída à origem sobre a direção  $v_j^m$  ortogonal à linha do mapa, e, em seguida, é subtraída a projeção pela distância da linha do mapa à origem. Para eliminar possíveis dúvidas relacionadas ao cálculo da distância entre as linhas, a figura 4.6 ilustra geometricamente como é realizado o cálculo em questão.

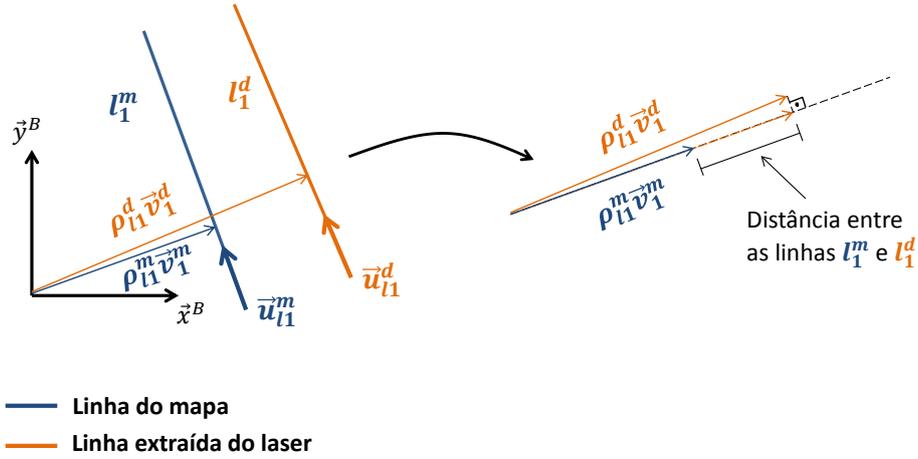


Figura 4.6: Exemplo geométrico do cálculo da distância entre linhas extraídas e as linhas associadas do mapa.

A projeção realizada na equação (4.9) contribui para definir a distância entre as linhas associadas ainda que a primeira etapa de minimização da orientação não realize o alinhamento perfeito entre as linhas extraídas e linhas do mapa. Os ruídos do *laser* e o método não determinístico de extração de linhas (*Ransac*) podem gerar variações de orientação nas linhas extraídas, fato que produz esta pequena diferença de orientação entre as linhas associadas.

Após realizar a segunda associação entre as linhas, define-se a nova função custo a ser minimizada, a fim de estimar a translação que alinha os *landmarks* associados.

Denotando a nova função custo como  $E_T(T_l)$ , temos que:

$$E_T(T_l) = \sum_{i=1}^{N_l} [(T_l^T v_i^d + \rho_{li}^d) v_i^{dT} v_i^m - \rho_{li}^m]^2 \quad (4.10)$$

onde  $T_l = [T_x, T_y]$  é o vetor de translação aplicado às linhas extraídas para que sejam alinhadas às linhas do mapa. Portanto, para alcançar o objetivo desta etapa, é necessário encontrar o vetor  $T_l$  que minimiza a função custo  $E_T(T_l)$  definida pela equação (4.10).

Uma das formas de se obter a translação  $T_l$  ótima é descrita como segue:

Definindo o produto escalar  $\gamma_i = v_i^{dT} v_i^m$  e substituindo os vetores  $T_l = [T_x, T_y]$ ,  $v_i^d = [v_{xi}^d, v_{yi}^d]$  e  $v_i^m = [v_{xi}^m, v_{yi}^m]$  na equação (4.10), temos:

$$E_T(T_x, T_y) = \sum_{i=1}^{N_l} [(T_x v_{xi}^d + T_y v_{yi}^d + \rho_{li}^d) \gamma_i - \rho_{li}^m]^2 \quad (4.11)$$

Os extremos da função custo ocorrem quando  $dE_T/dT_x = 0$  e  $dE_T/dT_y = 0$ , assim:

$$\begin{aligned} \frac{dE_T}{dT_x} &= \sum_{i=1}^{N_l} 2[(T_x v_{xi}^d + T_y v_{yi}^d + \rho_{li}^d) \gamma_i - \rho_{li}^m] v_{xi}^d \gamma_i = 0 \\ &= \sum_{i=1}^{N_l} 2[T_x (v_{xi}^d \gamma_i)^2 + T_y v_{xi}^d v_{yi}^d \gamma_i^2 + (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{xi}^d \gamma_i] = 0 \\ &= T_x \sum_{i=1}^{N_l} (v_{xi}^d \gamma_i)^2 + T_y \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 + \sum_{i=1}^{N_l} (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{xi}^d \gamma_i = 0 \end{aligned} \quad (4.12)$$

isolando à esquerda da igualdade os somatórios da equação (4.12) com os parâmetros  $T_x$  e  $T_y$  multiplicados, enquanto o somatório sem os referidos parâmetros são isolados à direita da igualdade, temos:

$$T_x \sum_{i=1}^{N_l} (v_{xi}^d \gamma_i)^2 + T_y \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 = \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{xi}^d \gamma_i \quad (4.13)$$

Da mesma forma para  $dE_T/dT_y = 0$ , temos:

$$\begin{aligned} \frac{dE_T}{dT_y} &= \sum_{i=1}^{N_l} 2[(T_x v_{xi}^d + T_y v_{yi}^d + \rho_{li}^d) \gamma_i - \rho_{li}^m] v_{yi}^d \gamma_i = 0 \\ &= \sum_{i=1}^{N_l} 2[T_x v_{xi}^d v_{yi}^d \gamma_i^2 + T_y (v_{yi}^d \gamma_i)^2 + (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{yi}^d \gamma_i] = 0 \\ &= T_x \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 + T_y \sum_{i=1}^{N_l} (v_{yi}^d \gamma_i)^2 + \sum_{i=1}^{N_l} (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{yi}^d \gamma_i = 0 \end{aligned} \quad (4.14)$$

isolando à esquerda da igualdade os somatórios da equação (4.14) com os parâmetros  $T_x$  e  $T_y$  multiplicados, enquanto o somatório sem os referidos parâmetros são isolados à direita da igualdade, temos:

$$T_x \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 + T_y \sum_{i=1}^{N_l} (v_{yi}^d \gamma_i)^2 = \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{yi}^d \gamma_i \quad (4.15)$$

Transformando as equações (4.13) e (4.15) na forma matricial:

$$\begin{bmatrix} \sum_{i=1}^{N_l} (v_{xi}^d \gamma_i)^2 & \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 \\ \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 & \sum_{i=1}^{N_l} (v_{yi}^d \gamma_i)^2 \end{bmatrix} \begin{bmatrix} T_x \\ T_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{xi}^d \gamma_i \\ \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{yi}^d \gamma_i \end{bmatrix} \quad (4.16)$$

$$\text{definindo } A = \begin{bmatrix} \sum_{i=1}^{N_l} (v_{xi}^d \gamma_i)^2 & \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 \\ \sum_{i=1}^{N_l} v_{xi}^d v_{yi}^d \gamma_i^2 & \sum_{i=1}^{N_l} (v_{yi}^d \gamma_i)^2 \end{bmatrix}$$

$$A \begin{bmatrix} T_x \\ T_y \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{xi}^d \gamma_i \\ \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{yi}^d \gamma_i \end{bmatrix} \quad (4.17)$$

Conclui-se que a translação ótima que minimiza a função custo  $E_T(T_l)$  é definida por:

$$T_l = \begin{bmatrix} T_x \\ T_y \end{bmatrix} = A^{-1} \begin{bmatrix} \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{xi}^d \gamma_i \\ \sum_{i=1}^{N_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{yi}^d \gamma_i \end{bmatrix}, \quad \text{se } \det(A) \neq 0 \quad (4.18)$$

Os casos degenerativos em que não é possível calcular a translação ótima ocorrem quando  $\det(A) = 0$ . A figura 4.7 ilustra um caso em que existem infinitas soluções para a translação ótima.

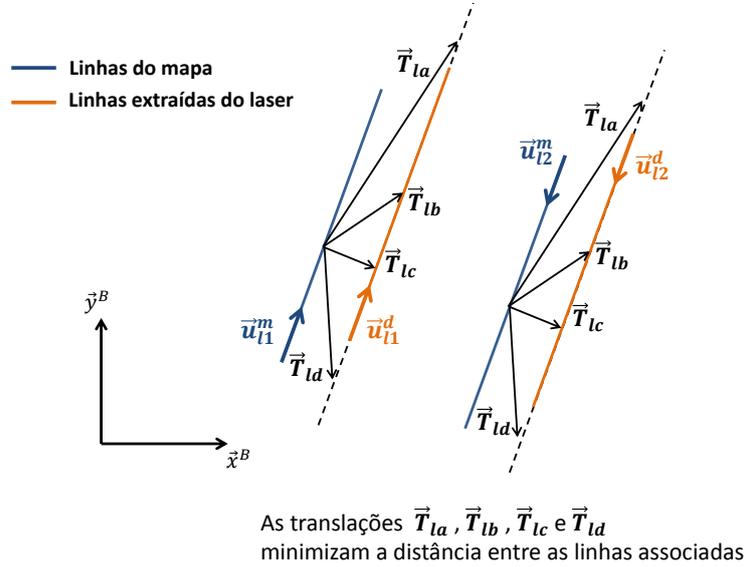


Figura 4.7: Exemplo geométrico da minimização da distância entre as linhas associadas para o caso em que existe mais de uma solução

Portanto, para garantir o cálculo da translação ótima é necessário que existam pelo menos duas linhas do mapa não paralelas e associadas às respectivas linhas extraídas.

Ao final das duas etapas de *matching* dos *landmarks*, torna-se disponível a

translação  $T_l$  e o ângulo  $\phi_l$  da rotação realizadas pelo robô. Considerando conhecida a posição e orientação do robô antes do deslocamento executado, a nova medida da configuração do robô, após o movimento, é obtida adicionando-se a translação  $T_l$  e o ângulo de rotação  $\phi_l$  à posição e orientação imediatamente anteriores ao deslocamento do robô.

Em posse das novas informações de posição e orientação medidas a partir do *laser*, é aplicado então o Filtro de Kalman Estendido (EKF), visando a obtenção da melhor estimativa da configuração do robô.

### 4.2.3 Localização utilizando o Filtro de Kalman Estendido (EKF)

O Filtro de Kalman Estendido (EKF) é uma ferramenta utilizada frequentemente em aplicações robóticas para realizar a fusão de dados de sensores e estimativa de posição e orientação. Estudos como Armesto & Tornero (2006), Christiand et al. (2010), Lionis & Kyriakopoulos (2002), Lizarralde et al. (2003) executam a fusão de dados da odometria de robôs móveis com informações de variados sensores como *lasers*, câmeras monoculares, ultrassom, dentre outros, através do método EKF.

O EKF consiste basicamente em uma variante do Filtro de Kalman clássico, permitindo ampliar o método original para aplicações em sistemas não lineares.

O Filtro de Kalman foi proposto na década de 60 por Rudolf Emil Kalman com o propósito de filtragem e predição em sistemas lineares.

Através de uma abordagem recursiva o Filtro de Kalman é capaz de estimar o estado de um processo aleatório, utilizando as medidas e o modelo do processo. Para tal, o método utiliza quatro considerações essenciais (Thrun et al. 2005):

- O processo aleatório é definido como do tipo Markov cuja principal propriedade é a definição dos estados futuros do processo como dependentes apenas do estado atual, ou seja, são independentes do passado do processo;
- A função densidade de probabilidade do estado seguinte deve ser linear nos argumentos, ou seja, o modelo do sistema deve ser linear com ruído gaussiano adicionado;
- A função densidade de probabilidade da medida do sistema também deve ser linear nos argumentos, ou seja, o modelo da medida do sistema deve ser linear com ruído gaussiano adicionado;
- A função densidade de probabilidade do estado inicial do sistema deve apresentar uma distribuição normal.

Desta maneira, a variável aleatória do sistema representado no filtro de Kalman

possui a seguinte função densidade de probabilidade:

$$f_{\mathbf{x}} = \det(2\pi\boldsymbol{\Sigma})^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right\} \quad (4.19)$$

onde  $\mathbf{x} \in \mathbb{R}^n$  é a variável aleatória gaussiana contínua no tempo que representa o estado do sistema,  $\boldsymbol{\mu}$  é a média e  $\boldsymbol{\Sigma}$  é a matriz de covariância.

Considerando um modelo contínuo do sistema sem entradas, a equação de estados discreta que modela o sistema pode ser obtida realizando amostras do processo em instantes  $(t_0, t_1, t_2, \dots, t_k, \dots)$ . Desta maneira, o modelo discreto do sistema sem entradas é definido por:

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{w}_k \quad (4.20)$$

onde  $\mathbf{w}_k$  representa um vetor cujos elementos são sequências de variáveis aleatórias gaussianas com média zero. Deste modo, define-se  $\mathbf{Q}_k$  como a matriz de covariância da variável  $\mathbf{w}_k$ .

Para descrever o Filtro de Kalman é necessário definir a medida do processo como:

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (4.21)$$

Consequentemente, observando as equações (4.21) e (4.20) que definem o sistema, a nomenclatura de cada termo é realizada como segue:

- a)  $\mathbf{x}_k$ : Vetor de estado que representa o processo aleatório no instante  $t_k$ ;
- b)  $\Phi_k$ : Matriz de transição que relaciona diretamente o estado  $\mathbf{x}_{k+1}$  com o estado  $\mathbf{x}_k$ , sem o ruído  $\mathbf{w}_k$ ;
- c)  $\mathbf{w}_k$ : Vetor cujos elementos são variáveis aleatórias gaussianas com média zero e matriz de covariância  $\mathbf{Q}_k$ ;
- d)  $\mathbf{z}_k$ : Medida do sistema que apresenta ruído gaussiano adicionado com média zero, sendo representado pelo vetor  $\mathbf{v}_k$ ;
- e)  $\mathbf{H}_k$ : Matriz que relaciona a medida  $\mathbf{z}_k$  com o estado do sistema  $\mathbf{x}_k$ , sem a presença do ruído.

Deste modo, a descrição do Filtro de Kalman para calcular a melhor estimativa do estado  $\mathbf{x}_k$  é apresentada a seguir. Deve ser ressaltado que o símbolo chapéu ( $\hat{\phantom{x}}$ ) representa a estimativa da variável e o símbolo menos ( $\bar{\phantom{x}}$ ) no sobrescrito indica a melhor estimativa antes de incorporar a medida do processo.

Assim, o erro da estimativa a priori é definido por:

$$e_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (4.22)$$

A matriz de covariância associada ao erro  $e_k^-$ :

$$\mathbf{P}_k^- = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \quad (4.23)$$

O ganho do filtro de Kalman é definido por:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (4.24)$$

A atualização da estimativa, utilizando as medidas do sistema, é descrita pela equação (4.25) abaixo:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (4.25)$$

Desta maneira, a variável  $\hat{\mathbf{x}}_k$  representa a estimativa atualizada. O erro da estimativa atualizada é definido como segue:

$$e_k = \mathbf{x}_k - \hat{\mathbf{x}}_k \quad (4.26)$$

Consequentemente, a matriz de covariância  $\mathbf{P}_k$  do erro da estimativa atualizada  $e_k$  é definida por:

$$\begin{aligned} \mathbf{P}_k &= E[e_k e_k^T] \\ \mathbf{P}_k &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \end{aligned}$$

a partir de Brown & Hwang (1997) é possível concluir que:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (4.27)$$

Para calcular a estimativa a priori do estado no instante posterior  $t_{k+1}$ , temos:

$$\hat{\mathbf{x}}_{k+1}^- = \Phi_k \hat{\mathbf{x}}_k \quad (4.28)$$

sendo o erro da estimativa a priori no instante posterior  $t_{k+1}$ :

$$\begin{aligned} e_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- \\ &= (\Phi_k \mathbf{x}_k + \mathbf{w}_k) - \Phi_k \hat{\mathbf{x}}_k \\ e_{k+1}^- &= \Phi_k e_k + \mathbf{w}_k \end{aligned} \quad (4.29)$$

Assim, a matriz de covariância associada ao erro  $e_{k+1}^-$  se torna:

$$\begin{aligned} \mathbf{P}_{k+1}^- &= E[e_{k+1}^- e_{k+1}^{-T}] \\ &= E[(\Phi_k e_k + \mathbf{w}_k)(\Phi_k e_k + \mathbf{w}_k)^T] \\ \mathbf{P}_{k+1}^- &= \Phi_k \mathbf{P}_k \Phi_k^T + \mathbf{Q}_k \end{aligned} \quad (4.30)$$

Desta maneira, o conjunto de equações (4.24),(4.25),(4.27),(4.28) e (4.30) constituem o método de Kalman para obter a estimativa ótima do processo, utilizando medidas do sistema. A figura 4.8 ilustra a sequência recursiva do método.

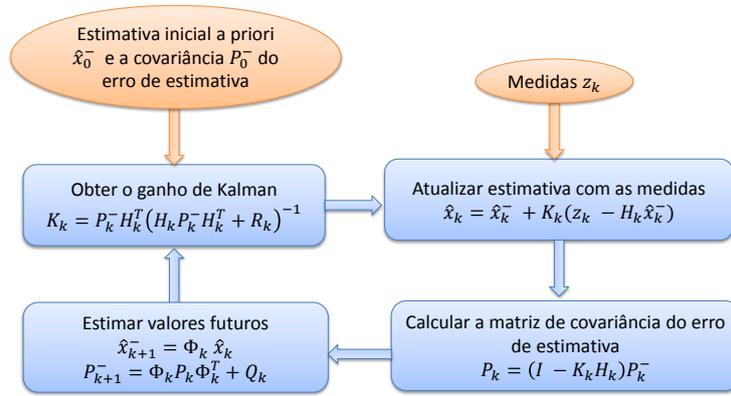


Figura 4.8: Diagrama do Filtro de Kalman

As equações do filtro de Kalman apresentadas até o momento foram derivadas a partir de um sistema sem entradas, entretanto os resultados apresentados também podem ser obtidos para um sistema com entradas, como é realizado em (Thrun et al. 2005).

Em virtude das características dos sistemas de uma maneira geral, na prática, a maior parte das aplicações não atendem à hipótese de linearidade do sistema e medidas. Por conseguinte, para superar a limitação imposta pela hipótese de linearidade, utiliza-se o Filtro de Kalman Estendido (EKF) que é uma derivação do método clássico de Kalman.

O modelo não linear utilizado no EKF é definido abaixo:

$$\mathbf{x}_k = g(\mathbf{x}_k, \mathbf{u}_k) + \delta_k \quad (4.31)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (4.32)$$

onde

- $\mathbf{x}_k$ : Vetor de estado que representa o processo aleatório no instante  $t_k$ ;
- $g(\mathbf{x}_k, \mathbf{u}_k)$ : Função não linear que relaciona diretamente o estado  $\mathbf{x}_{k+1}$  com o estado  $\mathbf{x}_k$  sem o ruído gaussiano ( $\delta_k$ );

- c)  $\mathbf{u}_k$ : Variável de entrada determinística do sistema;
- d)  $\delta_k$ : Ruído do sistema com média zero e matriz de covariância  $\mathbf{Q}_k$ ;
- e)  $\mathbf{z}_k$ : Medida do sistema que apresenta ruído gaussiano com média zero representado pelo vetor  $\mathbf{v}_k$ ;
- f)  $\mathbf{v}_k$ : Ruído de medida com média zero e matriz de covariância  $\mathbf{R}_k$ ;
- g)  $h(\mathbf{x}_k)$ : Função não linear que relaciona a medida  $\mathbf{z}_k$  com a variável de estado  $\mathbf{x}_k$  sem a presença do ruído.

A principal característica do EKF é a linearização das funções  $g(\mathbf{x}_k, \mathbf{u}_k)$  e  $h(\mathbf{x}_k)$  do sistema em torno da trajetória estimada. Assim, a linearização de  $g(\mathbf{x}_k, \mathbf{u}_k)$  em torno do estado estimado  $\hat{\mathbf{x}}_k$  se torna:

$$g(\mathbf{x}_k, \mathbf{u}_k) = g(\hat{\mathbf{x}}_k, \mathbf{u}_k) + \left[ \frac{dg}{d\mathbf{x}_k} \right]_{(\hat{\mathbf{x}}_k, \mathbf{u}_k)} (\mathbf{x}_k - \hat{\mathbf{x}}_k) \quad (4.33)$$

$$\mathbf{G}_k = \left[ \frac{dg}{d\mathbf{x}_k} \right]_{(\hat{\mathbf{x}}_k, \mathbf{u}_k)} \quad (4.34)$$

Do mesmo modo, a linearização da função  $h(\mathbf{x}_k)$  em torno do estado estimado a priori  $\hat{\mathbf{x}}_k^-$

$$h(\mathbf{x}_k) = h(\hat{\mathbf{x}}_k^-) + \left[ \frac{dh}{d\mathbf{x}_k} \right]_{\hat{\mathbf{x}}_k^-} (\mathbf{x}_k - \hat{\mathbf{x}}_k^-) \quad (4.35)$$

$$\mathbf{H}_k = \left[ \frac{dh}{d\mathbf{x}_k} \right]_{\hat{\mathbf{x}}_k^-} \quad (4.36)$$

Após calcular as matrizes  $\mathbf{H}_k$  e  $\mathbf{G}_k$  linearizadas, as equações do Filtro de Kalman Estendido (EKF) são obtidas da mesma maneira como o filtro clássico apresentado anteriormente.

Para aplicar o EKF ao problema proposto nesta dissertação o processo aleatório é definido pelas equações da odometria do robô e as medidas de posição são calculadas a partir do método de *matching* de dados do *laser* apresentado na seção 4.2.2.

A seguir, as subseções 4.2.4 e 4.2.5 descrevem o modelo da odometria do robô móvel com esteiras e o filtro EKF aplicado ao problema proposto, respectivamente.

#### 4.2.4 Modelo da odometria do robô móvel com esteiras

Em virtude da semelhança estrutural entre o deslocamento de um robô do tipo EOD com esteiras e os robôs móveis com duas rodas, as equações de movimento da odometria utilizadas neste trabalho foram baseadas no modelo cinemático de um robô móvel com direção diferencial. O modelo cinemático desta categoria de robôs

é descrita com detalhes no capítulo 6, cabendo a esta subseção apenas descrever o modelo da odometria utilizado no trabalho.

Usualmente os robôs móveis apresentam *encoders* que medem as rotações realizadas pelas rodas e, conseqüentemente, permitem calcular a medida da posição e orientação do robô, através da utilização dos dados no modelo da odometria.

Considerando como entrada do sistema robótico as velocidades angular e linear  $\omega_k$  e  $v_k$  constantes no intervalo  $[t_k, t_{k+1})$ , uma das aproximações possíveis apresentada em Siciliano et al. (2009), para obter a medida de posição e orientação do robô após o deslocamento, é observada a seguir:

$$\begin{aligned}x_{k+1} &= x_k + v_k \Delta t \cos(\phi_k) \\y_{k+1} &= y_k + v_k \Delta t \sin(\phi_k) \\ \phi_{k+1} &= \phi_k + \omega_k \Delta t\end{aligned}\tag{4.37}$$

colocando a expressão (4.37) sob a forma matricial, temos:

$$\begin{bmatrix}x_{k+1} \\ y_{k+1} \\ \phi_{k+1}\end{bmatrix} = \begin{bmatrix}x_k \\ y_k \\ \phi_k\end{bmatrix} + \begin{bmatrix}\Delta t \cos(\phi_k) & 0 \\ \Delta t \sin(\phi_k) & 0 \\ 0 & \Delta t\end{bmatrix} \begin{bmatrix}v_k \\ \omega_k\end{bmatrix}\tag{4.38}$$

onde

- a)  $(x_k, y_k)$ : Coordenadas  $x$  e  $y$  da posição do robô medidas através da odometria no instante  $t_k$ ;
- b)  $\phi_k$ : Medida de orientação do robô obtida através da odometria no instante  $t_k$ ;
- c)  $(x_{k+1}, y_{k+1})$ : Coordenadas  $x$  e  $y$  da posição do robô medidas através da odometria no instante  $t_{k+1}$ ;
- d)  $\phi_{k+1}$ : Medida de orientação do robô obtida através da odometria no instante  $t_{k+1}$ ;
- e)  $v_k$ : Velocidade linear de entrada;
- f)  $\omega_k$ : Velocidade angular de entrada;
- g)  $\Delta t = t_{k+1} - t_k$ : Duração do intervalo de amostragem.

Considerando um robô móvel com direção diferencial, é possível representar as equações da odometria (4.38) em termos das velocidades angulares das rodas esquerda  $\omega_{lk}$  e direita  $\omega_{rk}$ . A relação direta entre velocidades  $(v_k, \omega_k)$  e as velocidades

de rotação das rodas ( $\omega_{lk}$ ,  $\omega_{rk}$ ) é definida como

$$v_k = \frac{r_e}{2}(\omega_{rk} + \omega_{lk}) \quad (4.39)$$

$$\omega_k = \frac{r_e}{d_R}(\omega_{rk} - \omega_{lk}) \quad (4.40)$$

onde  $r_e$  é o raio da roda ou o raio da engrenagem principal que traciona as esteiras para o caso do robô do tipo EOD, e  $d_R$  é distância entre o centro das duas rodas ou esteiras.

Substituindo as equações (4.39),(4.40) no modelo da odometria (4.38):

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \Delta t \cos(\phi_k) & 0 \\ \Delta t \sin(\phi_k) & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \frac{r_e}{2}(\omega_{rk} + \omega_{lk}) \\ \frac{r_e}{d_R}(\omega_{rk} - \omega_{lk}) \end{bmatrix} \quad (4.41)$$

As equações da odometria (4.41) podem ainda ser representadas em termos da rotação realizada por cada roda ( $\Delta\theta_{rk}$ ,  $\Delta\theta_{lk}$ ), medida através dos *encoders*. Desta maneira:

$$\begin{aligned} \Delta\theta_{rk} &= \omega_{rk}\Delta t \\ \Delta\theta_{lk} &= \omega_{lk}\Delta t \end{aligned} \quad (4.42)$$

Reordenando a expressão (4.41) para que o parâmetro  $\Delta t$  seja multiplicado às velocidades  $\omega_{rk}$  e  $\omega_{lk}$ , temos:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \cos(\phi_k) & 0 \\ \sin(\phi_k) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r_e}{2}(\omega_{rk}\Delta t + \omega_{lk}\Delta t) \\ \frac{r_e}{d_R}(\omega_{rk}\Delta t - \omega_{lk}\Delta t) \end{bmatrix} \quad (4.43)$$

Substituindo (4.42) na expressão anterior (4.43):

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} + \begin{bmatrix} \cos(\phi_k) & 0 \\ \sin(\phi_k) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r_e}{2}(\Delta\theta_{rk} + \Delta\theta_{lk}) \\ \frac{r_e}{d_R}(\Delta\theta_{rk} - \Delta\theta_{lk}) \end{bmatrix} \quad (4.44)$$

Desta maneira, a equação (4.44) representa o modelo da odometria em função do ângulo da rotação realizada por cada roda.

Para simplificar as expressões e não sobrecarregar a notação, nas próximas subseções utiliza-se o modelo da odometria (4.38) em função das velocidades linear  $v_k$  e angular  $\omega_k$  do robô.

## 4.2.5 Filtro EKF aplicado ao robô móvel com esteiras

Após apresentar as equações da odometria em função das velocidades linear e angular do robô, é possível, então, descrever o filtro EKF aplicado ao modelo de odometria de um robô móvel com esteiras.

Em primeiro lugar, os vetores de estado  $\mathbf{x}_k$  e  $\mathbf{x}_{k+1}$  que representam o processo aleatório nos instantes  $(t_k, t_{k+1})$  são definidos pelas coordenadas de posição e orientação do modelo odométrico (4.38) nos mesmos instantes de tempo:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix}, \quad \mathbf{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix} \quad (4.45)$$

Utilizando a representação acima em conjunto com a equação da odometria (4.38), é possível representar o processo aleatório que define o movimento do robô em termos de uma parcela livre de ruídos e outra componente de ruído gaussiano:

$$\underbrace{\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \phi_{k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \underbrace{\begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix}}_{\mathbf{x}_k} + \underbrace{\begin{bmatrix} \Delta t \cos(\phi_k) & 0 \\ \Delta t \sin(\phi_k) & 0 \\ 0 & \Delta t \end{bmatrix}}_{g(\mathbf{x}_k, \mathbf{u}_k)} \underbrace{\begin{bmatrix} v_k \\ \omega_k \end{bmatrix}}_{\mathbf{u}_k} + \mathcal{N}(0, \mathbf{Q}_k) \quad (4.46)$$

onde,

- a)  $\mathbf{u}_k = [v_k \ \omega_k]$ : Variável de entrada do processo aleatório no instante  $t_k$ ;
- b)  $\mathcal{N}(0, \mathbf{Q}_k)$ : Ruído aleatório gaussiano com média zero e matriz de covariância  $\mathbf{Q}_k$ ;
- c)  $g(\mathbf{x}_k, \mathbf{u}_k)$ : Função não linear que relaciona o estado posterior  $\mathbf{x}_{k+1}$  ao estado atual  $\mathbf{x}_k$  sem a adição ruído  $\mathcal{N}(0, \mathbf{Q}_k)$ .

Existem diversas incertezas que podem ser incorporadas ao modelo de odometria (4.38) e, conseqüentemente, ao processo aleatório (4.46), como erros de medida no raio da engrenagem  $r_e$  e largura entre as esteiras  $d_R$ , ou incertezas na medida de velocidade. Desta maneira, por simplificação, foi incorporada à odometria uma incerteza através do ruído gaussiano  $\mathcal{N}(0, \mathbf{Q}_k)$  que apresenta três componentes, sendo um componente para cada elemento do vetor de estado  $\mathbf{x}_k$ .

Como já mencionado anteriormente, a matriz  $\mathbf{G}_k$  é obtida derivando  $g(\mathbf{x}_k, \mathbf{u}_k)$  em relação ao estado  $\mathbf{x}_k$  e, em seguida, avaliando a derivada da função em relação

ao estado estimado  $\hat{\mathbf{x}}_k$ :

$$\mathbf{G}_k = \left[ \frac{\partial g(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right]_{\hat{\mathbf{x}}_k, \mathbf{u}_k} = \begin{bmatrix} 1 & 0 & -v_k \Delta t \sin(\hat{\phi}_k) \\ 0 & 1 & v_k \Delta t \cos(\hat{\phi}_k) \\ 0 & 0 & 1 \end{bmatrix} \quad (4.47)$$

Para fins de simplificação, foi considerado que não existe correlação entre cada um dos três componentes do ruído  $\mathcal{N}(0, \mathbf{Q}_k)$ , assim a matriz de covariância  $\mathbf{Q}_k$  associada ao ruído, se torna diagonal:

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix} \quad (4.48)$$

Onde  $\sigma_x$ ,  $\sigma_y$  e  $\sigma_\phi$  representam o desvio padrão de cada componente do ruído.

O EKF exige ainda que a equação referente às medidas do processo também seja linearizada.

A partir do método de *matching* de dados descrito na seção 4.2.2, é possível dispor do vetor das coordenadas de posição e orientação do robô, a partir do *laser*. No caso ideal, esta medida seria perfeita e igual ao vetor de estado  $\mathbf{x}_k$ , entretanto em virtude dos ruídos de medida introduzidos pelo sensor *laser* e pelo método de *matching*, o vetor de medidas  $\mathbf{z}_k$  foi definido por duas parcelas. A primeira parcela é idêntica ao vetor  $\mathbf{x}_k$  e a segunda é definida por ruídos gaussianos.

$$\mathbf{z}_k = \underbrace{\mathbf{x}_k}_{h(\mathbf{x}_k)} + \mathcal{N}(0, \mathbf{R}_k) \quad (4.49)$$

onde  $\mathcal{N}(0, \mathbf{R}_k)$  representa o ruído associado à medida com média zero e matriz de covariância  $\mathbf{R}_k$ . Por simplificação, foi considerado que os componentes do vetor de ruído  $\mathcal{N}(0, \mathbf{R}_k)$  não são correlacionados, desta maneira a matriz de covariância  $\mathbf{R}_k$  é expressa da seguinte forma:

$$\mathbf{R}_k = \begin{bmatrix} \sigma_{Lx}^2 & 0 & 0 \\ 0 & \sigma_{Ly}^2 & 0 \\ 0 & 0 & \sigma_{L\phi}^2 \end{bmatrix} \quad (4.50)$$

onde  $\sigma_{Lx}$ ,  $\sigma_{Ly}$  e  $\sigma_{L\phi}$  representam o desvio padrão de cada componente do ruído. A matriz  $\mathbf{H}_k$  é obtida realizando a derivada de  $h(\mathbf{x}_k)$  em relação ao vetor de estado  $\mathbf{x}_k$  e avaliando em  $\hat{\mathbf{x}}_k^-$ , assim:

$$\mathbf{H}_k = \left[ \frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}_k} \right]_{\hat{\mathbf{x}}_k^-} = I \quad (4.51)$$

Ou seja, a matriz  $\mathbf{H}_k$  é representada pela matriz identidade.

Após obter a estimativa a priori inicial  $\hat{\mathbf{x}}_0^-$  e a respectiva matriz de covariância  $\mathbf{P}_0^-$  do erro de estimativa a priori, temos que o ganho do filtro de Kalman é definido por:

$$\mathbf{H}_k = I \quad , \quad \mathbf{R}_k = \begin{bmatrix} \sigma_{Lx}^2 & 0 & 0 \\ 0 & \sigma_{Ly}^2 & 0 \\ 0 & 0 & \sigma_{L\phi}^2 \end{bmatrix} \quad (4.52)$$

$$\mathbf{K}_k^- = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (4.53)$$

Após adquirir a medida  $\mathbf{z}_k$  a partir do método de *matching* de dados do *laser*, é possível então obter a estimativa a priori da medida ( $\hat{\mathbf{z}}_k^-$ ) e a estimativa do estado do sistema ( $\hat{\mathbf{x}}_k$ ).

$$\hat{\mathbf{z}}_k^- = h(\hat{\mathbf{x}}_k^-) = \hat{\mathbf{x}}_k^- \quad (4.54)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \quad (4.55)$$

Em seguida, atualiza-se o estado estimado a priori  $\hat{\mathbf{x}}_{k+1}^-$  e a matriz de covariância  $\mathbf{P}_{k+1}^-$  do erro associado

$$\mathbf{u}_k = \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \quad (4.56)$$

$$\hat{\mathbf{x}}_{k+1}^- = g(\hat{\mathbf{x}}_k, \mathbf{u}_k) \quad , \quad \hat{\mathbf{x}}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\phi}_k \end{bmatrix} \quad (4.57)$$

$$\hat{\mathbf{x}}_{k+1}^- = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\phi}_k \end{bmatrix} + \begin{bmatrix} \Delta t \cos(\hat{\phi}_k) & 0 \\ \Delta t \sin(\hat{\phi}_k) & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} v_k \\ \omega_k \end{bmatrix} \quad (4.58)$$

$$\mathbf{Q}_k = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix} \quad (4.59)$$

$$\mathbf{P}_{k+1}^- = \mathbf{G}_k \mathbf{P}_k \mathbf{G}_k^T + \mathbf{Q}_k \quad (4.60)$$

Após a atualização da estimativa a priori e da respectiva matriz de covariância do erro de estimativa, a sequência é iniciada novamente. Deste modo, durante o deslocamento do robô, as medidas do *laser* e da odometria são utilizadas para obter a melhor estimativa de posição e orientação através do Filtro de Kalman Estendido (EKF).

**Observação 4.4** *As abordagens de localização como em (Jang et al. 2005, Siegwart et al. 2004, Thrun et al. 2005) usualmente utilizam medidas de landmarks extraídos do ambiente através de sensores como lasers ou câmeras monoculares, de modo que as medidas de posição e orientação dos referidos landmarks são utilizadas no Filtro de Kalman Estendido (EKF), linearizando o modelo das medidas. Diferentemente das abordagens mencionadas, esta dissertação propõe a localização utilizando o EKF em conjunto com o método de matching de landmarks que, por sua vez, fornece diretamente a medida de posição e orientação do robô. Desta maneira, o modelo da medida se torna linear, reduzindo possíveis efeitos da não linearidade das medidas na estimativa ótima da posição e orientação do robô.*

*Ressalta-se ainda que o método de matching de landmarks apresenta uma maior robustez a ruídos no sensor e rapidez em relação a outros métodos, pois os landmarks são extraídos utilizando a segmentação dos dados e o método Ransac que permite eliminar ruídos e estimar modelos com uma amostra inferior à quantidade total dos dados. Desta maneira, o método de matching de landmarks proposto se torna mais robusto e reduz o tempo de processamento em comparação com abordagens existentes (Choi et al. 2012, Li et al. 2010) em que são utilizados ponto a ponto os dados da varredura para realizar o matching dos escaneamentos do laser.*

### 4.3 Localização durante a locomoção na escada

Após realizar a primeira etapa da localização, durante o deslocamento do robô até a posição mais próxima à escada, é necessário cumprir a segunda etapa que consiste em estimar a posição ( $x$ ) e orientação ( $\phi$ ) do robô durante a locomoção sobre a escada. Para tal, a hipótese que define a existência de pelo menos uma das paredes laterais à escada, deve ser válida.

Durante a locomoção sobre a escada, optou-se por utilizar apenas as informações de um *laser* posicionado paralelamente ao plano da escada para obter os dados da localização.

O método de localização proposto utiliza a varredura do *laser* durante o deslocamento sobre a escada e extrai as linhas correspondentes às paredes laterais, semelhante à primeira etapa da localização. Em seguida, os *landmarks* extraídos são comparados aos *landmarks* do mapa através da abordagem desenvolvida e conseqüentemente são calculadas a coordenada  $x$  da posição do robô em relação às laterais da escada e sua orientação  $\phi$ .

O deslocamento sobre o plano da escada, bem como os sistemas de coordenadas da escada  $O_S$ , do robô  $O_R$  e do sensor *laser* horizontal  $O_{L_H}$  são demonstrados na figura 4.9.

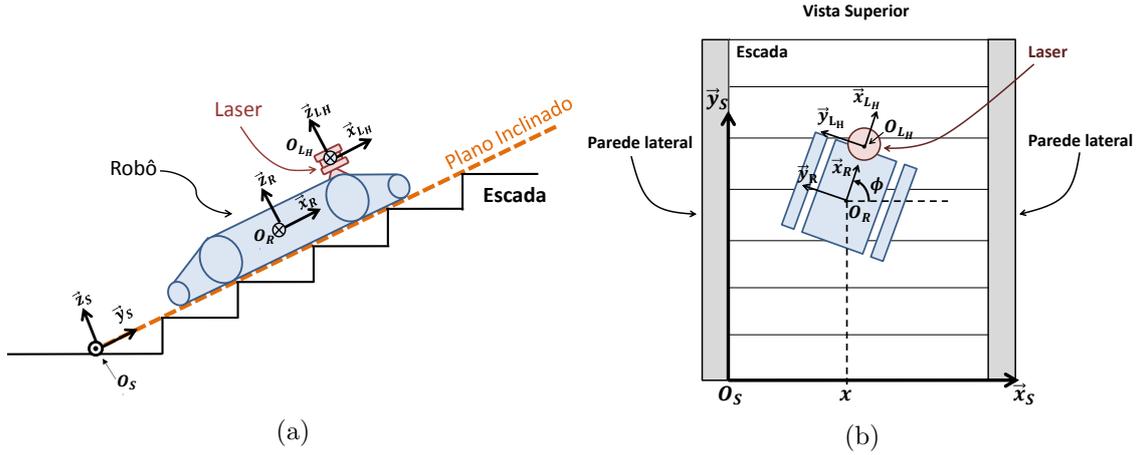


Figura 4.9: Vista lateral (a) e superior (b) do robô realizando o deslocamento sobre a escada e a representação dos sistemas de coordenadas  $O_S$ ,  $O_R$  e  $O_{LH}$

Como já mencionado, a hipótese da existência de pelo menos uma parede lateral à escada deve ser satisfeita, pois a detecção da mesma através do *laser* permite calcular as estimativas de posição e orientação do robô. Ressalta-se que o posicionamento do sensor *laser* em relação ao robô é o mesmo utilizado na primeira etapa de localização durante o deslocamento até a escada.

Durante a localização sobre a escada, o método realiza o mesmo procedimento de extração de linhas descrito na seção 4.2.1, entretanto nesta etapa as linhas extraídas são referentes às paredes laterais da escada. Após extrair os *landmarks*, é definida então a abordagem para associá-los aos *landmarks* do mapa e assim estimar a posição  $x$  e orientação  $\phi$  do robô sobre a escada.

Ao realizar o deslocamento sobre a escada, as rotações e translações realizadas pelo robô provocam as mesmas transformações nos *landmarks* extraídos do *laser*. Portanto, novamente são definidas duas funções custo que devem ser minimizadas para realizar o *matching* dos *landmarks* e, conseqüentemente, calcular a rotação e translação ótimas do robô. A diferença básica desta abordagem de *matching* para aquela utilizada na primeira etapa de localização, consiste no fato da translação ótima, neste caso, apresentar apenas uma coordenada em  $x_S$ .

A etapa de *matching* de dados para estimar a rotação ótima  $R_l$  foi definida exatamente da mesma maneira como na primeira etapa da localização, sendo descrita na subseção 4.2.2. Após obter o ângulo  $\phi_l$  ótimo da rotação realizada pelo robô, a rotação em questão é aplicada às linhas extraídas para orientar suas direções com as linhas do mapa. Em seguida, é realizado o procedimento para estimar a translação ótima que alinha os *landmarks*.

A translação  $T_l$  ótima é calculada de modo semelhante à abordagem realizada na primeira etapa de localização, descrita na subseção 4.2.2.

Primeiramente é definida a função de associação entre as linhas extraídas e linhas

do mapa:

$$e_{tij} = \eta_{trans} |(v_i^{dT} v_j^m) \rho_{li}^d - \rho_{lj}^m| + \eta_{rot} |\arccos(u_{li}^{dT} u_{lj}^m)| \quad (4.61)$$

$$\text{onde } v_i^d = \begin{bmatrix} \cos(\theta_{li}) \\ \sin(\theta_{li}) \end{bmatrix}, \quad v_i^m = \begin{bmatrix} \cos(\theta_{lj}) \\ \sin(\theta_{lj}) \end{bmatrix}$$

onde  $v_i^d$  e  $v_j^m$  são vetores unitários ortogonais às direções das linhas extraídas e linhas do mapa, respectivamente. Já os parâmetros  $\eta_{trans}$  e  $\eta_{rot}$  definem os pesos atribuídos aos erros de distância e orientação entre as linhas. Desta maneira, a associação é realizada entre as linhas com o menor erro.

Após realizar a associação entre as linhas, a função custo  $E_{T_{lx}}(T_{lx})$  a ser minimizada, a fim de estimar a translação que alinha os *landmarks* associados, é definida da seguinte maneira:

$$E_{T_{lx}}(T_{lx}) = \sum_{i=1}^{n_l} [(T_{lx} v_{xi}^d + \rho_{li}^d) v_i^{dT} v_i^m - \rho_{li}^m]^2 \quad (4.62)$$

onde  $v_{xi}^d$  representa a coordenada  $x$  do vetor  $v_i^d$ . Definindo o produto escalar  $\gamma_i = v_i^{dT} v_i^m$  e substituindo na equação (4.62), temos

$$E_{T_{lx}}(T_{lx}) = \sum_{i=1}^{n_l} [(T_{lx} v_{xi}^d + \rho_{li}^d) \gamma_i - \rho_{li}^m]^2 \quad (4.63)$$

O extremo da função custo ocorre quando  $dE_{T_{lx}}/dT_{lx} = 0$ , assim:

$$\begin{aligned} \frac{dE_{T_{lx}}}{dT_{lx}} &= \sum_{i=1}^{n_l} 2[(T_{lx} v_{xi}^d + \rho_{li}^d) \gamma_i - \rho_{li}^m] v_{xi}^d \gamma_i = 0 \\ &= \sum_{i=1}^{n_l} 2[T_{lx} (v_{xi}^d \gamma_i)^2 + (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{xi}^d \gamma_i] = 0 \end{aligned} \quad (4.64)$$

atribuindo o somatório às duas expressões da equação (4.64) e, em seguida, isolando o parâmetro  $T_{lx}$ , temos:

$$\begin{aligned} T_{lx} \sum_{i=1}^{n_l} (v_{xi}^d \gamma_i)^2 + \sum_{i=1}^{n_l} (\rho_{li}^d \gamma_i - \rho_{li}^m) v_{xi}^d \gamma_i &= 0 \\ T_{lx} &= \frac{\sum_{i=1}^{n_l} (\rho_{li}^m - \rho_{li}^d \gamma_i) v_{xi}^d \gamma_i}{\sum_{i=1}^{n_l} (v_{xi}^d \gamma_i)^2}, \quad \sum_{i=1}^{n_l} (v_{xi}^d \gamma_i)^2 \neq 0 \end{aligned} \quad (4.65)$$

Desta maneira, após calcular o ângulo de rotação  $\phi_l$  e a translação  $T_{lx}$  através das informações do *laser*, podemos estimar a posição  $x$  e orientação  $\phi$  atuais, adicionando  $T_{lx}$  e  $\phi_l$  aos valores de posição e orientação anteriores ao deslocamento realizado.

## 4.4 Simulações e experimentos

Para validar a aplicação do EKF na solução do problema de localização em mapa conhecido, inicialmente foram realizadas simulações, utilizando o modelo de odometria de um robô móvel com esteiras e direção diferencial. O modelo e as equações do filtro foram implementados no software Matlab, em seguida os resultados obtidos foram analisados.

A primeira simulação realizada considerou o robô em uma posição inicial  $(x_0, y_0) = (0, 0)$  m e orientação  $\phi_0 = 45^\circ$ , as velocidades linear e angular do robô foram definidas como  $v_k = 0.08$  m/s e  $\omega_k = 0.14$  rad/s ( $8^\circ$ /s), respectivamente. A posição final do robô foi  $(x_f, y_f) = (-0.22, 0.847)$  m e orientação final  $\phi_f = 165.3^\circ$ . A taxa de amostragem utilizada foi de  $\Delta t = 0.1$  s. A matriz de covariância do ruído do sistema foi definido com parâmetros  $\sigma_x = \sigma_y = 0.005$  m e  $\sigma_\phi = 0.1^\circ$ , já os ruídos das medidas do sistema foram definidos com parâmetros  $\sigma_{Lx} = \sigma_{Ly} = 0.06$  m e  $\sigma_{L\phi} = 1^\circ$ .

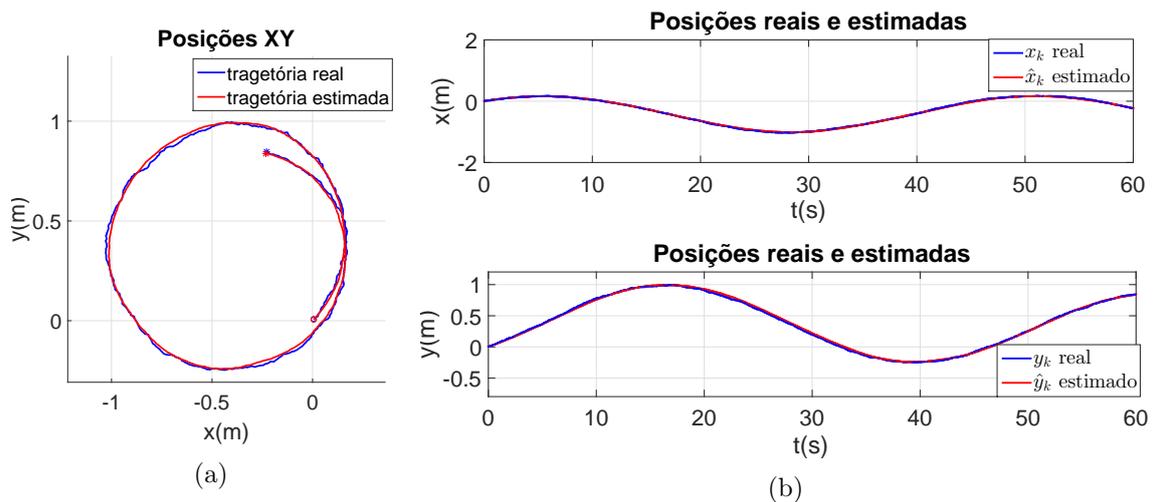


Figura 4.10: Simulação do método de localização. O gráfico (a) apresenta as trajetórias real e estimada no plano xy e os gráficos em (b) apresentam as posições x e y reais e estimadas, para os parâmetros de simulação:  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.1^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.06$  m e  $\sigma_{L\phi} = 1^\circ$ .

Como pode ser observado nos gráficos da figura 4.10, o filtro EKF é capaz de estimar a posição do robô com valores próximos da posição real.

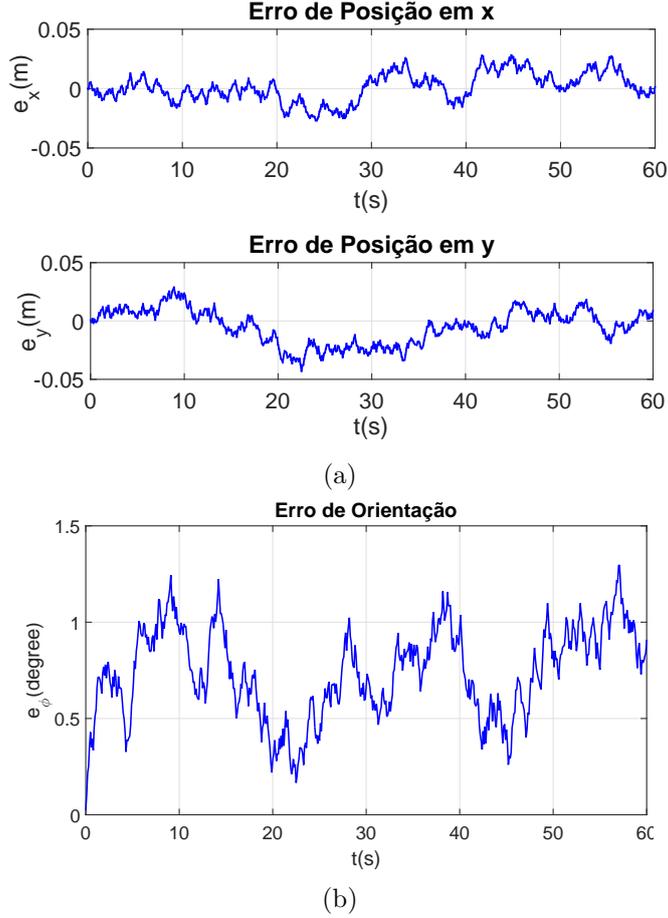


Figura 4.11: Simulação do método de localização. O gráfico (a) apresenta o erro de estimativa em  $x$  e  $y$ , já o gráfico (b) apresenta o erro de orientação, para os parâmetros de simulação  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.1^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.06$  m e  $\sigma_{L\phi} = 1^\circ$ .

Os erros da estimativa realizada pelo EKF são apresentados na figura 4.11, indicando uma variação em módulo menor que 0.05 m entre as posições reais e estimadas. Já o erro de orientação apresentou variação em torno de  $1^\circ$ . Tendo em vista os ruídos do modelo e medidas do sistema, os resultados confirmam a boa estimativa realizada pelo método EKF aplicado.

Para verificar a robustez do método de localização, uma nova simulação foi realizada adicionando-se fatores de escorregamento para cada esteira, sendo denotado  $i_l$  o fator da esteira esquerda e  $i_r$  o fator da esteira direita. O escorregamento foi definido como uma distribuição aleatória, podendo apresentar dois valores [0.8, 1]. Desta maneira, para inserir o escorregamento no modelo da odometria, as equações das velocidades linear (4.39) e angular (4.40) do robô se tornam:

$$v_k = \frac{r_e}{2}(i_r\omega_{rk} + i_l\omega_{lk}) \quad (4.66)$$

$$\omega_k = \frac{r_e}{d_R}(i_r\omega_{rk} - i_l\omega_{lk}) \quad (4.67)$$

Para a nova simulação, a posição e orientação iniciais do robô foram mantidas em

$(x_0, y_0) = (0, 0)$  m e  $\phi_0 = 45^\circ$ , as velocidades angulares das esteiras foram definidas como  $\omega_{lk} = 0.38$  rad/s e  $\omega_{rk} = 1.22$  rad/s, o escorregamento  $i_l = i_r = [0.8, 1]$ , já os ruídos do modelo, ruídos das medidas do sistema e taxa de amostragem foram mantidos como na primeira simulação.

Os resultados podem ser observados nos gráficos da figura 4.12, de onde é possível concluir que mesmo após adicionar o escorregamento, o filtro foi capaz de estimar a posição e orientação do robô.

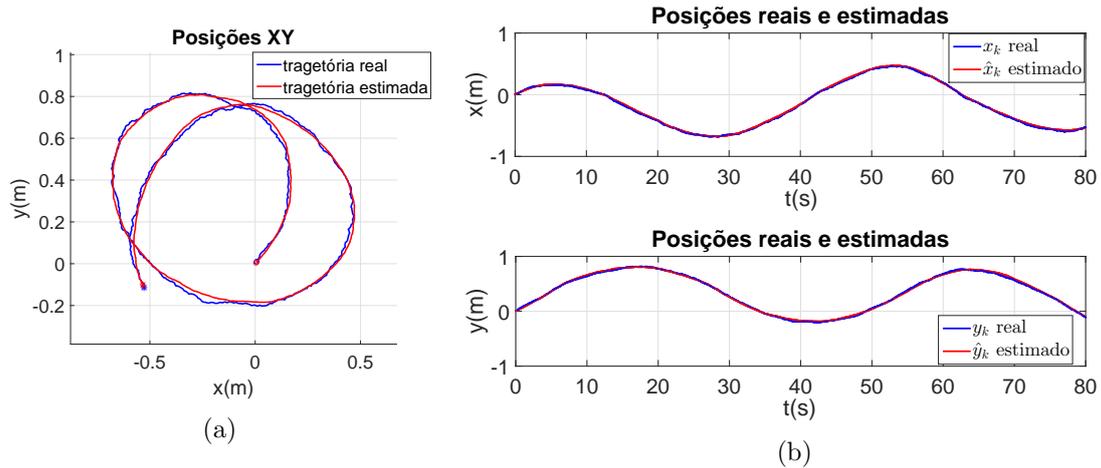


Figura 4.12: Simulação do método de localização. O gráfico (a) apresenta as trajetórias real e estimada no plano xy e os gráficos em (b) apresentam as posições x e y reais e estimadas, para os parâmetros de simulação:  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.1^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.06$  m,  $\sigma_{L\phi} = 1^\circ$ ,  $i_l = i_r = [0.8, 1]$ .

Os gráficos da figura 4.13 demonstram que o erro de estimativa do filtro EKF para as coordenadas x e y se mantiveram inferior a 0.05 m em módulo. Já o erro da estimativa de orientação apresentou variação menor que  $1^\circ$  na maior parte do tempo de simulação. Portanto, mesmo adicionando o fator de escorregamento, o filtro EKF foi capaz de estimar a posição e orientação do robô com valores dentro do esperado.

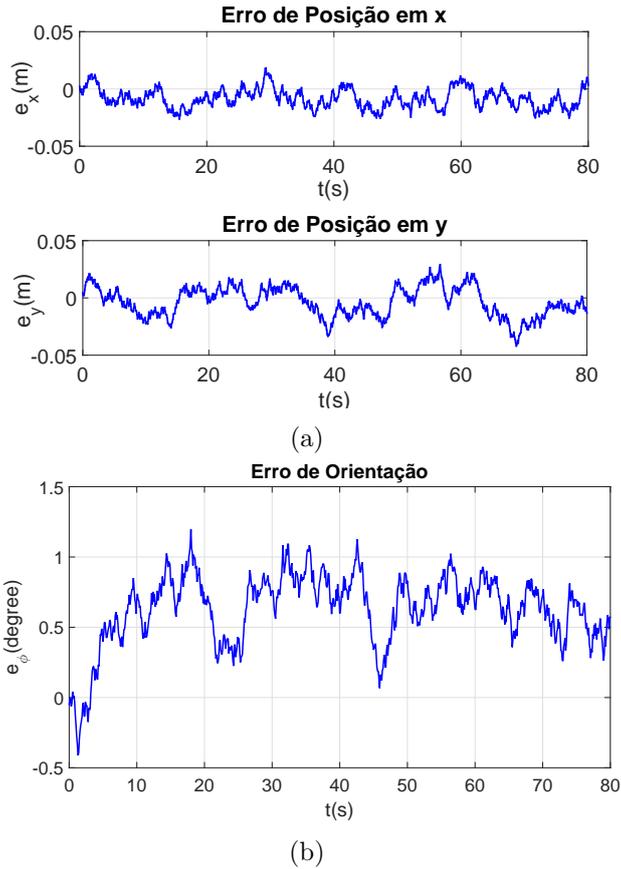


Figura 4.13: Simulação do método de localização. O gráfico (a) apresenta o erro de estimação em x e y, já o gráfico (b) apresenta o erro de orientação, para os parâmetros de simulação:  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.1^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.06$  m,  $\sigma_{L\phi} = 1^\circ$ ,  $i_l = i_r = [0.8, 1]$ .

#### 4.4.1 Resultados experimentais

Neste subseção são apresentados os experimentos realizados para validar as duas etapas do método: a localização durante a aproximação até a escada e a localização durante o deslocamento sobre a escada.

##### Localização durante aproximação até a escada

Após as simulações, o método de localização foi avaliado através de experimentos reais utilizando o robô móvel *Roomba* e o *laser scan* Sick LMS111 posicionado horizontalmente.

O *Roomba* é um robô móvel que apresenta duas rodas com direção diferencial, de modo que sensores internos podem fornecer dados da odometria como as velocidades angular e linear, além da posição e orientação do robô. A figura 4.14 ilustra o robô em questão.

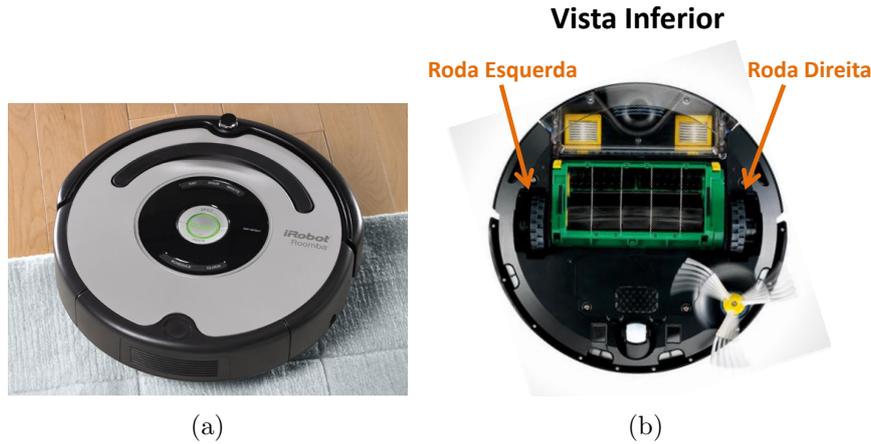


Figura 4.14: Vistas em perspectiva (a) e inferior (b) do robô Roomba

A operação do robô, bem como a aquisição dos dados da odometria e do *laser* foram realizadas através do *framework* ROS instalado no sistema operacional Linux/Ubuntu. Uma estrutura foi acoplada ao robô para a instalação do *laser scan* posicionado horizontalmente.

A posição real inicial do robô foi definida como  $(x_0, y_0) = (2.48, 2.18)$  m e a posição real final após o experimento  $(x_f, y_f) = (1.28, 4.97)$  m, ambas medidas em relação a um referencial inercial  $O_B$ . A seguir são apresentados os resultados gráficos.

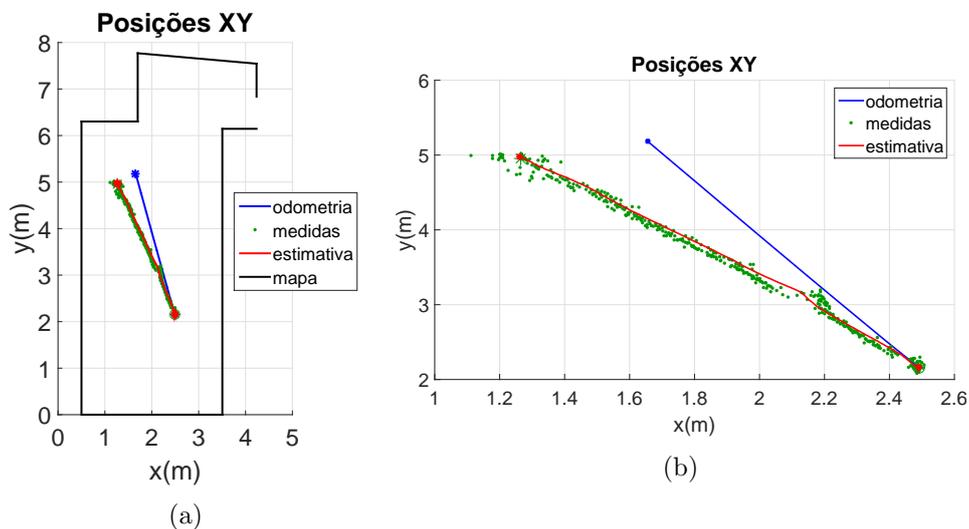


Figura 4.15: Experimento do método de localização utilizando robô móvel. O gráfico (a) apresenta as trajetórias estimada, odométrica e medidas (método de matching - *laser*) no plano xy e o gráfico (b) apresenta uma ampliação das mesmas trajetórias, para os parâmetros:  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.5^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.05$  m,  $\sigma_{L\phi} = 1^\circ$ .

Como pode ser observado nos gráficos apresentados na figura 4.15, as posições do robô estimadas a partir do método de localização (EKF) e as medidas de posição obtidas a partir do *laser* apresentaram valores próximos, enquanto que a trajetória

odométrica diferiu da trajetória estimada em maior nível.

A posição final estimada foi  $(\hat{x}_f, \hat{y}_f) = (1.26, 4.973)$ , de modo que o erro entre as posições finais estimada e real foi de  $(e_x, e_y) = (-0.0195, 0.001)$  m.

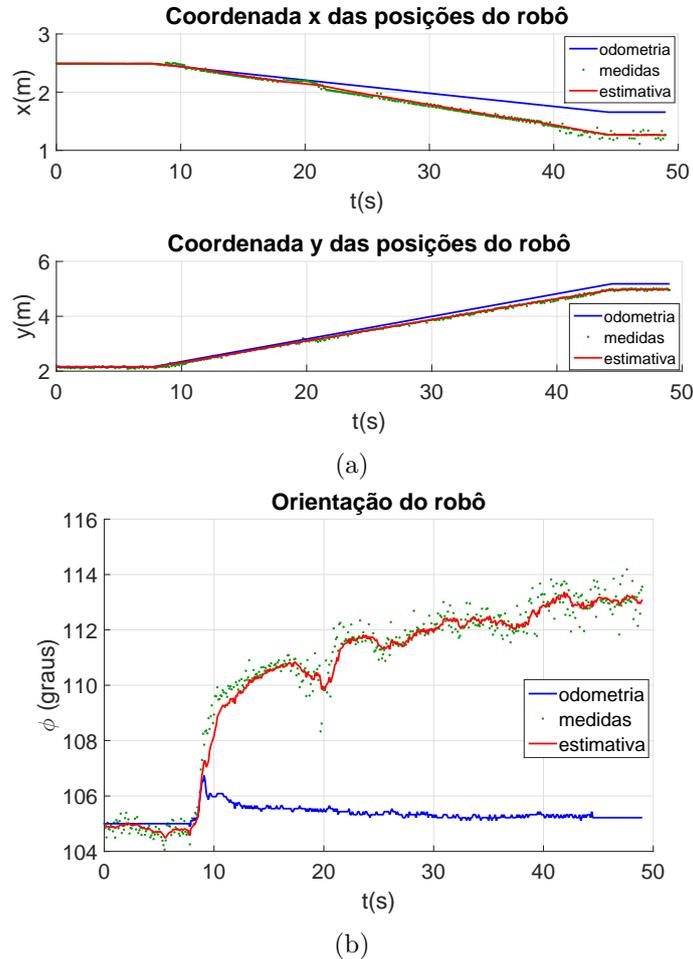


Figura 4.16: Experimento do método de localização utilizando robô móvel. O gráfico (a) apresenta as trajetórias do robô em x e y, já o gráfico (b) apresenta os dados estimados, odométricos e medidos da orientação do robô, para os parâmetros:  $\sigma_x = \sigma_y = 0.005$  m,  $\sigma_\phi = 0.5^\circ$ ,  $\sigma_{Lx} = \sigma_{Ly} = 0.05$  m,  $\sigma_{L\phi} = 1^\circ$ .

A partir do gráfico da orientação do robô (figura 4.16(b)), observa-se novamente que as estimativas do EKF e as medidas obtidas através do *laser* foram próximas, enquanto que a orientação final fornecida pela odometria diferiu em aproximadamente  $7^\circ$  da estimativa.

Portanto, tendo em vista o erro de medida do *laser* e considerando os dados da odometria discrepantes em relação à posição real, conclui-se que o método de localização adotado foi capaz de estimar a trajetória real, apresentando um erro final de aproximadamente 0.02 m (magnitude) em relação à posição real final. O que valida o método proposto.

## Localização durante a locomoção sobre a escada

Os experimentos referentes a segunda etapa da localização foram realizados utilizando um *laser* posicionado paralelamente ao plano da escada, o framework ROS e o software Matlab.

Os sistemas de coordenadas da escada  $O_S$ , do robô  $O_R$  e do sensor *laser* horizontal  $O_{LH}$  adotados na etapa de localização durante a locomoção sobre a escada são novamente apresentados na figura 4.17.

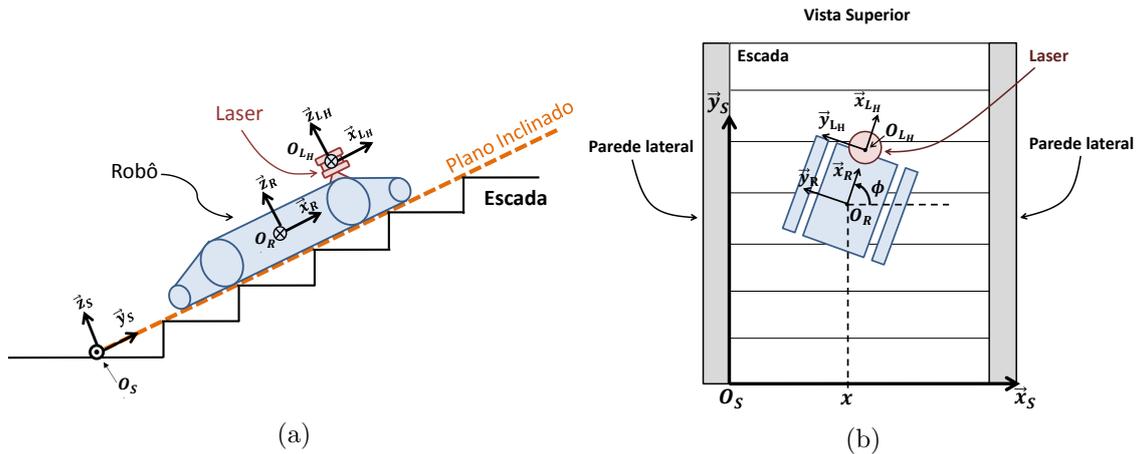


Figura 4.17: Vista lateral (a) e superior (b) do robô e sistemas de coordenadas  $O_S$ ,  $O_R$  e  $O_{LH}$  adotados na etapa de localização durante a locomoção sobre a escada

Foram definidos três pontos de referência em relação ao sistema de coordenadas da escada  $O_S$ , em torno dos quais a posição do *laser* foi variada. Os referidos pontos foram 0.2 m, 0.6 m e 1.0 m, todos medidos sobre o eixo  $x_S$  do sistema  $O_S$ . As paredes laterais à escada utilizada no experimento são separadas por uma distância  $W_S = 1.2$  m.

Como o método de localização calcula apenas a coordenada  $x$  da posição do robô e a sua orientação  $\phi$ , a coordenada  $y$  foi incrementada virtualmente para que a representação do gráfico  $XY$  permitisse distinguir todos os pontos medidos através do método. Desta maneira, o gráfico 4.18 apresenta o plano  $XY$  com os pontos medidos e as posições de referência 0.2 m, 0.6 m e 1.0 m.

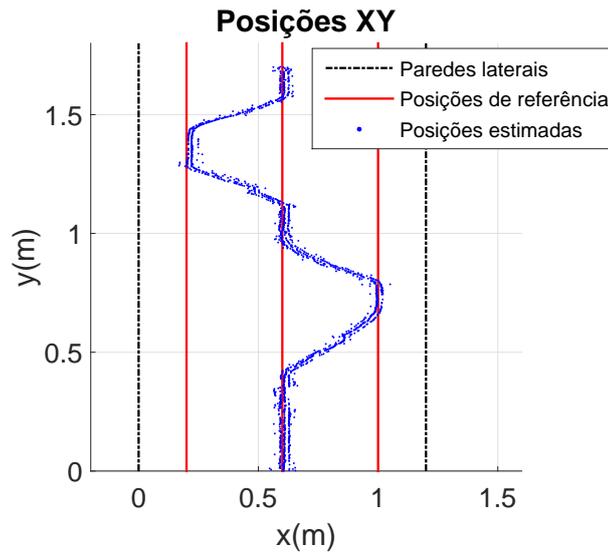


Figura 4.18: Experimento do método de localização sobre a escada. Gráfico da estimativa da posição no plano  $XY$  (sistema  $O_S$ ), utilizando as posições de referência 0.2 m, 0.6 m e 1.0 m

Os gráficos da figura 4.19 apresentam as estimativas de posição e orientação obtidas através do método de localização aplicado.

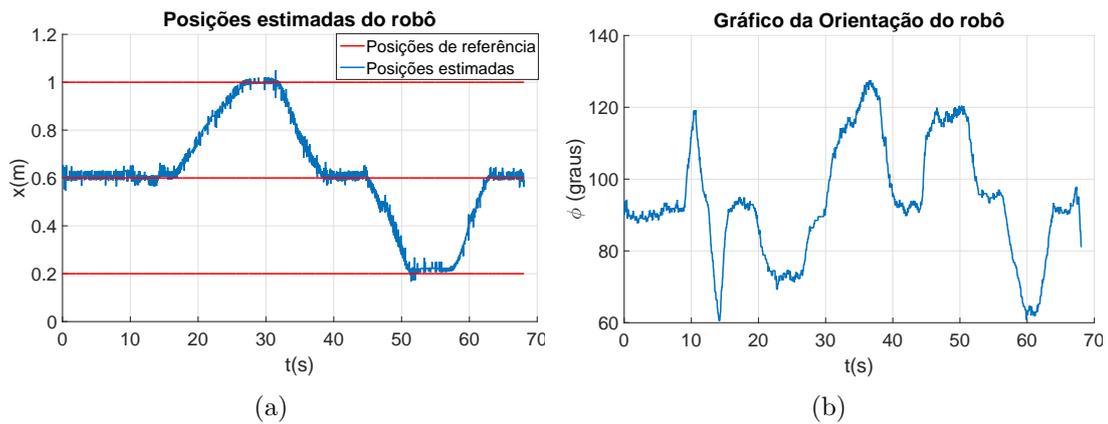


Figura 4.19: Experimento do método de localização sobre a escada. O gráfico (a) apresenta a estimativa da posição  $x$  e o gráfico (b) a estimativa da orientação  $\phi$ , utilizando as posições de referência 0.2 m, 0.6 m e 1.0 m

Como pode ser observado nos gráficos obtidos após o experimento, o método de localização aplicado foi capaz de estimar a coordenada  $x$  da posição do robô e sua orientação  $\phi$ , baseando-se nas paredes laterais. Ressalta-se ainda que o erro máximo de posição em relação aos pontos de referência foi de 0.02 m, aproximadamente. Desta maneira, os resultados obtidos validam o método proposto e são considerados boas estimativas da posição e orientação do robô, levando em consideração os ruídos do sensor.

## 4.5 Conclusões

Neste capítulo foram propostas duas abordagens para localização, uma durante o deslocamento até a escada e outra durante a locomoção sobre a mesma. Os métodos em questão foram descritos e avaliados através de simulações e experimentos.

Para a primeira etapa da localização, o modelo da odometria do robô foi apresentado, bem como o filtro EKF e a sua descrição aplicada ao problema proposto. O método de *matching* de *landmarks*, que calcula a medida da posição do robô através do *laser*, também foi apresentado. A implementação do EKF, utilizando os dados da odometria e do sensor *laser*, apresentou bons resultados durante os testes experimentais, visto que a diferença entre a posição final real e a estimada foi da ordem de 0.02 m.

A segunda etapa da localização, durante a locomoção sobre a escada, também foi descrita neste capítulo. O experimento realizado para avaliar a etapa em questão demonstrou a robustez do método aos ruídos do sensor, de modo que o erro de posição entre os valores estimados e os pontos de referência também foi da ordem de 0.02 m.

## Capítulo 5

# Metodologia para locomoção em escadas durante a subida

A definição da metodologia para locomoção em escadas é de suma importância, em virtude da complexidade da tarefa devido à possibilidade de escorregamentos, colisões e até tombamentos do robô, durante todo o deslocamento da posição próxima à escada até o pavimento superior.

Após realizar as tarefas de detecção e modelagem de escadas, além da localização durante a aproximação do robô até a mesma, é necessário definir a metodologia para a locomoção propriamente dita sobre a escada durante a subida. Deve ser ressaltado que o controle utilizado para conduzir o robô da posição inicial até a posição desejada próxima à escada é apresentado com detalhes no capítulo 6.

Como já mencionado anteriormente, os robôs móveis com esteiras do tipo EOD apresentam usualmente pernas articuladas por juntas de revolução e esteiras laterais, que auxiliam o deslocamento no ambiente e alteram a configuração da postura do robô quando necessário.

As esteiras laterais são responsáveis por realizar os movimentos de rotação e translação do robô através da rotação diferencial das mesmas. As pernas, que podem movimentar-se simultaneamente em pares ou independentemente entre si, são utilizadas para superar diferentes tipos de obstáculos, apresentando ainda esteiras que auxiliam durante o deslocamento.

O robô EOD considerado neste trabalho apresenta quatro pernas articuladas e duas esteiras laterais, de modo que as pernas também apresentam esteiras. As pernas frontais realizam movimentos simultâneos, assim como as pernas traseiras, entretanto o par de pernas frontal se movimenta independentemente em relação ao par traseiro.

A figura 5.1 ilustra os componentes do robô EOD mencionado e o movimento das pernas.

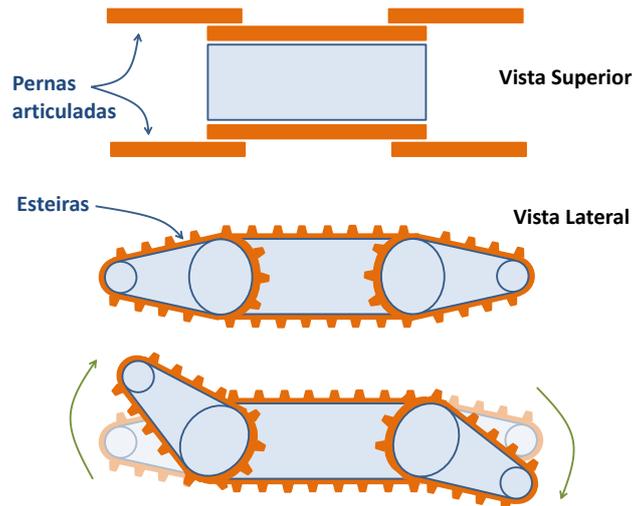


Figura 5.1: Vistas superior e lateral do robô móvel com esteiras e quatro pernas articuladas

Tendo em vista a estrutura dos robôs EOD com esteiras, a locomoção em escadas exige que sejam definidas as configurações das pernas e o controle das esteiras para conduzir o robô com segurança até o final da escada.

O procedimento para locomoção em escadas durante a subida pode ser dividido em três etapas: transição do piso inferior para a escada, deslocamento na escada durante a subida e aterrissagem suave no piso superior. Em Wang et al. (2012) e Mihankhah et al. (2009) são definidas três configurações principais para as pernas robóticas, com o objetivo de realizar cada uma das etapas mencionadas.

Assim como nos trabalhos mencionados, aqui são definidas três configurações das pernas para executar cada uma das etapas, entretanto a primeira e terceira configurações diferem das outras propostas. A primeira e terceira etapas apresentam variações de movimento em torno da configuração principal, seja para transição do piso inferior para escada, seja para realizar a aterrissagem suave no piso superior.

A definição de cada configuração foi realizada com base na modelagem do movimento das pernas através da cinemática direta. Deste modo, a partir dos ângulos das juntas é possível definir as posições das pernas, além dos pontos de contato entre escada e pernas, permitindo estabelecer as configurações necessárias à execução das etapas.

Para auxiliar a compreensão da metodologia utilizada, este capítulo é dividido como segue: a seção 5.1 define a cinemática direta do robô móvel com esteiras; as seções 5.2, 5.3 e 5.4 descrevem as três etapas que compõem o problema de reconfiguração da postura para locomoção em escadas durante a subida e as respectivas configurações das pernas robóticas utilizadas em cada etapa; por fim a seção 5.5 descreve as simulações realizadas para validar a metodologia.

## 5.1 Cinemática direta do robô móvel com esteiras do tipo EOD

A cinemática direta tem por objetivo calcular a posição e orientação do efetuador dos manipuladores robóticos em função das variáveis das juntas. Desta maneira, considera-se o manipulador como um conjunto de corpos rígidos (links), unidos por juntas (revolução, prismáticas etc), em que o efetuador é localizado na extremidade final da cadeia de corpos rígidos.

Para aplicar os conceitos da cinemática direta, o robô EOD com esteiras foi considerado como três links unidos por duas juntas de revolução. As pernas frontais apresentam movimento simultâneo, podendo ser representadas por um único corpo rígido, ocorrendo da mesma maneira para as pernas traseiras. Lembrando que o par de pernas dianteiro se movimenta independentemente do par traseiro.

A modelagem da cinemática direta do robô foi realizada de modo semelhante à convenção de Denavit-Hartenberg (DH) (Siciliano et al. 2009). O método DH permite encontrar as posições e orientações relativas entre links consecutivos, através das transformações de coordenadas correspondentes. A figura 5.2 ilustra os sistemas de coordenadas  $O_0$ ,  $O_1$  e  $O_2$  referentes a cada corpo rígido que compõe o robô: corpo 0 (chassi), corpo 1 (pernas traseiras) e corpo 2 (pernas frontais); além do sistema de coordenadas  $O_R$  do robô utilizado nas etapas de localização e controle.

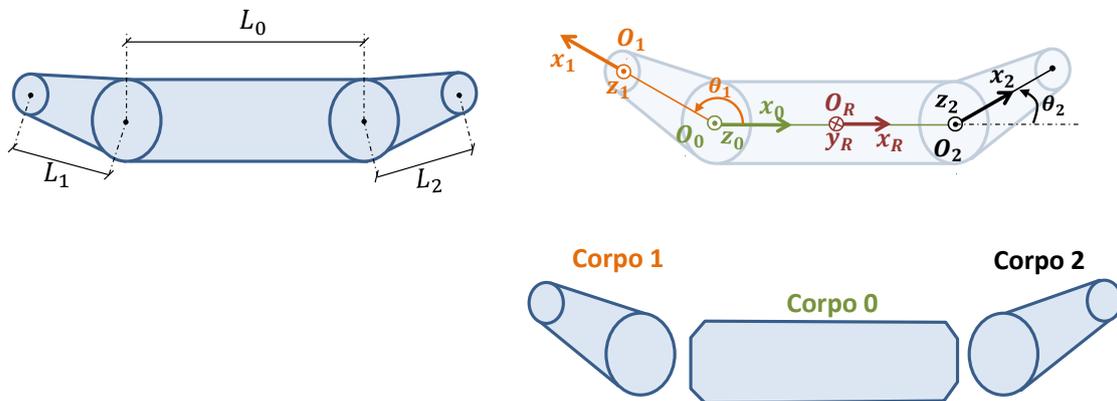


Figura 5.2: Sistemas de coordenadas e dimensões utilizados na cinemática direta do robô do tipo EOD

A partir da figura 5.2, observa-se que a origem do sistema de coordenadas  $O_1$  é fixado à perna traseira (corpo 1), a origem  $O_0$  é fixada ao chassi do robô (corpo 0) e a origem  $O_2$  é fixada à perna frontal (corpo 2), de forma que os movimentos de cada um destes corpos provocam os respectivos movimentos dos sistemas de coordenadas.

A relação entre os sistemas de coordenadas dos corpos pode ser expressa através

de matrizes de transformações homogêneas. As transformações homogêneas são representadas, desta maneira, por operações de rotação e translação.

Deste modo, a cinemática direta do robô EOD pode ser obtida a partir das seguintes transformações homogêneas:

$$T_{10} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & -L_1 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$T_{02} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L_0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

onde

- a)  $T_{10}$ : é a matriz de transformação homogênea do sistema de coordenadas  $O_0$  para o sistema  $O_1$ ;
- b)  $T_{02}$ : é a matriz de transformação homogênea do sistema de coordenadas  $O_2$  para o sistema  $O_0$ ;
- c)  $L_1, L_2$ : representam o comprimento das pernas traseiras e frontais respectivamente;
- d)  $L_0$ : representa a distância entre os sistemas de coordenadas  $O_0$  e  $O_2$ ;
- e)  $\theta_1, \theta_2$ : são os ângulos das juntas das pernas traseiras e frontais, respectivamente.

Após definir as transformações homogêneas que relacionam os sistemas de coordenadas fixados aos corpos rígidos que compõem o robô, são adquiridas as ferramentas necessárias para obter as configurações das pernas utilizadas em cada uma das etapas do problema de locomoção em escadas durante a subida.

## 5.2 Transição do piso inferior para a escada

A transição do piso inferior para a escada é a primeira etapa do processo de locomoção sobre a mesma. Neste ponto da missão, o robô está localizado próximo à escada, de modo que as orientações de ambos estão alinhadas.

Antes de iniciar a descrição do método, na figura 5.3 são apresentados os sistemas de coordenadas  $O_S$  da escada,  $O_R$  do robô, além de variáveis importantes como os ângulos de inclinação  $\alpha_S$  da escada e  $\alpha_R$  do robô medidos em relação ao plano do solo.

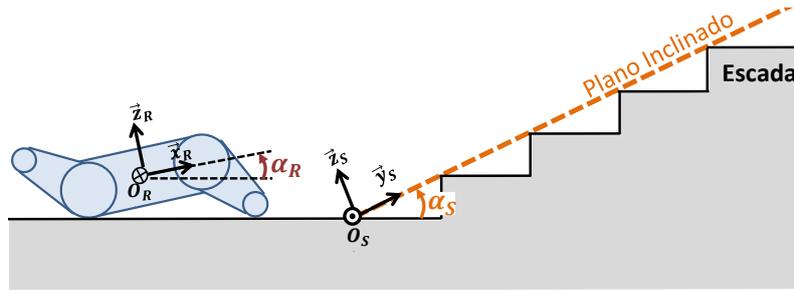


Figura 5.3: Representação dos ângulos de inclinação  $\alpha_S$  e  $\alpha_R$  da escada e robô, respectivamente.

O objetivo desta etapa é fazer com que o robô inicie o movimento em direção à escada, realizando o deslocamento sobre os primeiros degraus até que o ângulo de inclinação  $\alpha_R$  do robô se torne equivalente ao ângulo de inclinação da escada  $\alpha_S$  e as pernas traseiras do robô não mais estejam em contato com o solo.

A configuração inicial das pernas escolhida para realizar esta etapa é aquela em que as pernas frontais e traseiras estão afastadas do solo, de modo que ao movimentar as esteiras em direção à escada, as pernas frontais guiam o robô, forçando-o a realizar o movimento de elevação sobre os primeiros degraus. A figura 5.4 ilustra a posição inicial das pernas para a configuração em questão:

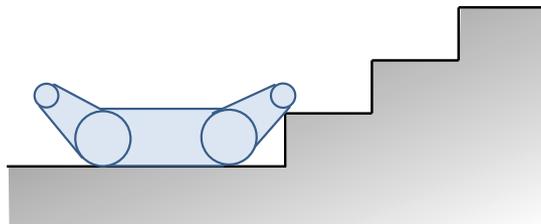


Figura 5.4: Representação do posicionamento das pernas robóticas para iniciar o movimento sobre os primeiros degraus

Para definir o ângulo das pernas frontais adotado na configuração inicial mencionada, é necessário estabelecer uma relação geométrica entre o ângulo  $\theta_2$  das pernas em questão e a altura  $h_S$  dos degraus da escada. O resultado desta relação é o cálculo do ângulo mínimo  $\theta_{2,min}$  das pernas frontais que deve ser adotado para superar os primeiros degraus da escada.

A figura 5.5 ilustra os parâmetros do problema e a posição correta da pernas robóticas para iniciar o deslocamento:

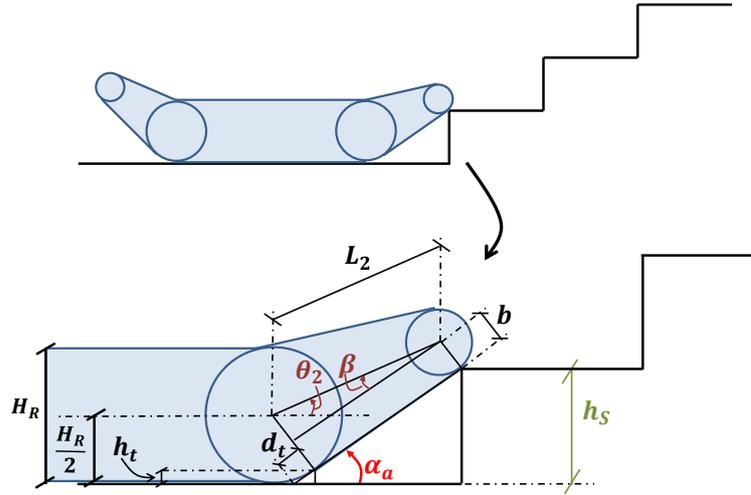


Figura 5.5: Representação da perna robótica em contato com o primeiro degrau da escada e os parâmetros necessários ao cálculo de  $\theta_{2,min}$

onde,

- a)  $\theta_2$ : Ângulo de rotação das juntas das pernas frontais;
- b)  $\beta$ : Ângulo entre o eixo principal da perna e a reta formada pela esteira da mesma;
- c)  $\alpha_a$ : Ângulo entre a esteira da perna frontal e o plano do solo;
- d)  $h_t, d_t$ : Parâmetros temporários para auxiliar os cálculos;
- e)  $L_2$ : Comprimento das pernas frontais;
- f)  $H_R$ : Altura do robô;
- g)  $b$ : Raio da circunferência menor que compõe a perna robótica.

A partir das relações geométricas representadas na ilustração 5.5, temos:

$$\sin(\beta) = \frac{\frac{H_R}{2} - b}{L_2}$$

$$\beta = \arcsin\left(\frac{\frac{H_R}{2} - b}{L_2}\right) \quad (5.3)$$

$$\alpha_a = \theta_2 + \beta \quad (5.4)$$

A partir dos resultados anteriores,

$$h_t = \frac{H_R}{2}(1 - \cos(\alpha_a)) \quad (5.5)$$

$$\frac{h_t}{h_S} = \frac{d_t}{d_t + L_2 \cos(\beta)}$$

$$d_t = \frac{h_t L_2 \cos(\beta)}{h_S - h_t} \quad (5.6)$$

Temos que o seno do ângulo  $\alpha_a$  é definido por:

$$\sin(\alpha_a) = \frac{h_t}{d_t} \quad (5.7)$$

substituindo (5.6) em (5.7):

$$\sin(\alpha_a) = \frac{h_t}{\frac{h_t L_2 \cos(\beta)}{h_S - h_t}} = \frac{h_S - h_t}{L_2 \cos(\beta)}$$

$$h_S = L_2 \sin(\alpha_a) \cos(\beta) + h_t \quad (5.8)$$

substituindo (5.5) e (5.4) na equação anterior, temos:

$$h_S = L_2 \sin(\theta_2 + \beta) \cos(\beta) + \frac{H_R}{2}(1 - \cos(\theta_2 + \beta)) \quad (5.9)$$

Desta maneira, a solução da equação (5.9) nos permite obter o ângulo mínimo  $\theta_{2,min}$  das pernas frontais necessário para superar os primeiros degraus com altura  $h_S$ . Como a equação em questão não apresenta uma solução analítica, o cálculo do ângulo  $\theta_{2,min}$  pode ser realizado através de métodos numéricos de solução.

Antes de apresentar o método para realizar a transição do piso inferior para a escada, é necessário definir ainda qual a altura máxima possível do degrau em que o robô é capaz de subir.

As definições da altura máxima do degrau  $h_{s,max}$  e do ângulo máximo das pernas  $\theta_{s,max}$  para executar a tarefa, foram realizadas novamente através de uma análise geométrica do problema considerando um único degrau como obstáculo. A figura 5.6 ilustra a posição da perna em relação ao obstáculo definido como um único degrau,

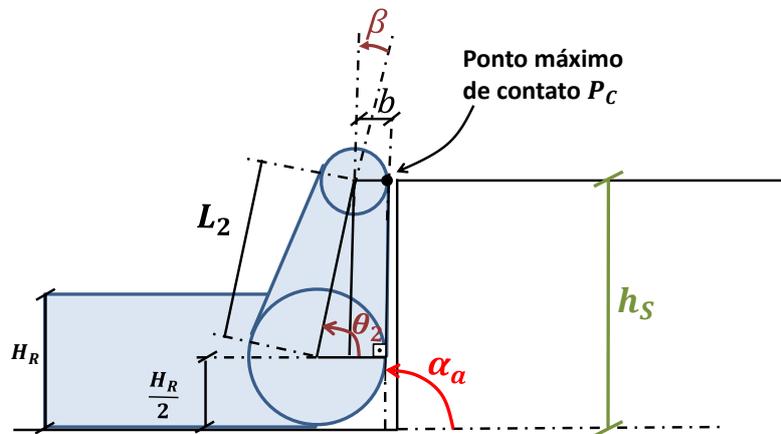


Figura 5.6: Representação da perna robótica em contato com o obstáculo definido como um único degrau

onde  $P_C$  é o ponto máximo de contato entre as pernas frontais e o primeiro degrau, ponto este que possibilita a elevação do robô sobre o referido degrau. Os outros parâmetros são definidos da mesma maneira como na figura 5.5.

Considerando que há atrito suficiente entre esteiras e o solo, bem como entre esteiras e o degrau, a altura máxima possível de um degrau o qual o robô consegue superar, é definida pela altura do ponto  $P_C$  quando o ângulo  $\alpha_a$  entre a parte inferior das pernas e o plano do solo for próximo de  $\pi/2$  rad ( $90^\circ$ ).

Desta maneira, a partir da figura 5.6 e retomando as equações (5.3) e (5.4) definidas anteriormente, temos:

$$\sin(\beta) = \frac{\frac{H_R}{2} - b}{L_2}$$

$$\beta = \arcsin\left(\frac{\frac{H_R}{2} - b}{L_2}\right) \quad (5.3)$$

$$\alpha_a = \theta_2 + \beta \quad (5.4)$$

Com  $\alpha_a \approx \pi/2$ , temos a partir de (5.4):

$$\theta_{2,max} = \frac{\pi}{2} - \beta \quad (5.10)$$

Para determinar a altura máxima  $h_{S,max}$  quando o ponto  $P_C$  está em contato com o degrau e o ângulo das pernas frontais é o máximo  $\theta_{2,max}$ , basta analisar as relações geométricas da figura 5.6, assim concluímos:

$$h_{S,max} = L_2 \cos(\beta) + \frac{H_R}{2} \quad (5.11)$$

Se substituirmos o ângulo  $\theta_{2,max}$  definido em (5.10) na equação (5.9), encontraremos o mesmo resultado anterior.

Desta maneira, a partir da medida do comprimento das pernas frontais  $L_2$ , do ângulo máximo  $\theta_{2,max}$  e da altura dos degraus  $h_S$  obtida através do método de detecção e modelagem da escada, é possível definir se o robô é capaz de iniciar a elevação sobre os primeiros degraus, comparando  $h_S$  com a altura máxima  $h_{S,max}$  definida pela equação (5.11).

Após definir o ângulo mínimo  $\theta_{2,min}$  para iniciar a locomoção sobre a escada e a altura máxima  $h_{S,max}$  do degrau que o robô é capaz de superar, é iniciado o método para realizar a transição do piso inferior para a escada.

### 5.2.1 Fases da transição

A etapa de transição do piso inferior para escada foi dividida em 3 fases denominadas: S1-A, S1-B e S1-C. As fases em questão podem ser consideradas como estados nos quais o robô executa um conjunto de ações, de modo que ao final das fases é iniciada a etapa de deslocamento propriamente dito sobre a escada.

Ressalta-se que em cada uma das fases, as entradas do sistema são os valores de referência para os ângulos das juntas das pernas robóticas e os valores de referência para as velocidades das esteiras, já as saídas são as medidas das respectivas variáveis (ângulos das juntas e velocidades das esteiras).

#### Fase S1-A

Na fase S1-A as pernas robóticas estão previamente posicionadas com ângulos  $\theta_2 \geq \theta_{2,min}$  e  $\theta_1 \leq \pi - \theta_{2,min}$ . O robô realiza o deslocamento na direção da escada, ao encontro da mesma, até que as pernas frontais entrem em contato com o primeiro degrau e o robô inicie a elevação.

Em virtude da posição das pernas e o contato com a escada, o avanço do robô e sua elevação fazem com que o chassi entre em contato com o degrau. O contínuo avanço das esteiras gera o aumento do ângulo de inclinação  $\alpha_R$  do robô e no momento em que os sensores inerciais do robô detectam que o seu ângulo é maior ou igual ao ângulo da escada  $\alpha_S$ , o avanço com as esteiras é interrompido e se inicia a fase S1-B. A figura 5.7 ilustra a realização da fase S1-A.



Figura 5.7: Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-A.

#### Fase S1-B

Caso o robô continue o avanço progressivo, o aumento de  $\alpha_R$  pode fazer com que haja uma queda brusca do robô sobre a escada, no momento em que seu centro de gravidade ultrapassar o ponto de contato entre o primeiro degrau e o chassi. Para evitar este problema, é necessário reduzir a inclinação  $\alpha_R$  do robô rotacionando as pernas traseiras até que a parte inferior das pernas esteja no mesmo plano inferior do chassi do robô, ou seja,  $\theta_1 = \pi + \beta$ . A figura 5.8 ilustra as posições das pernas descritas na fase S1-B.

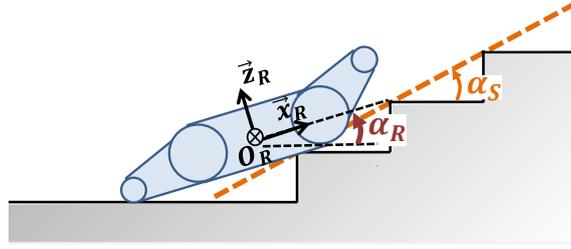


Figura 5.8: Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-B.

### Fase S1-C

Após reduzir a inclinação do robô, é iniciado novamente o movimento das esteiras impulsionando o robô sobre a escada até que o seu centro de massa ultrapasse o primeiro degrau e o seu ângulo de inclinação  $\alpha_R$  seja igual ao ângulo de inclinação  $\alpha_S$  da escada. Neste momento, o robô deixa de estar em contato com o solo posicionando-se no plano da escada, em contato apenas com os degraus.

Para aumentar a superfície de contato entre o robô e os degraus, as pernas frontais são rotacionadas para o ângulo  $\theta_2 = -\beta$ , de modo que a parte inferior das mesmas esteja alinhada ao plano do chassi. A figura 5.9 ilustra a realização da fase S1-C.

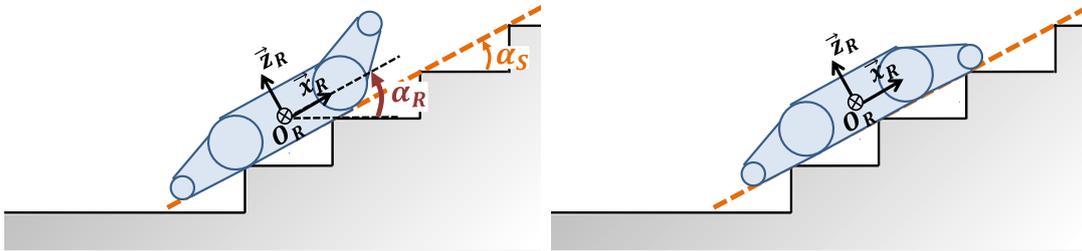


Figura 5.9: Exemplo das posições das pernas e robô durante a transição do piso inferior para a escada, fase S1-C.

**Observação 5.1** *Existem outras abordagens que podem ser utilizadas para apresentar a tarefa de transição do piso inferior para escada.*

*Uma outra abordagem possível seria considerar o sistema a eventos discretos em que uma máquina de estados coordena as ações do robô, podendo definir em cada estado, em um nível mais baixo, malhas fechadas de controle contínuas para controlar a orientação do robô atuando nas pernas robóticas. As malhas de controle podem utilizar as medidas da orientação do robô, de modo que o objetivo é manter seu ângulo de inclinação ( $\alpha_R$ ) equivalente ao ângulo de inclinação da escada ( $\alpha_S$ ).*

*Desta maneira, em uma representação conceitual, poderiam ser utilizados três estados S1-A, S1-B e S1-C para realizar a tarefa de transição do piso inferior para*

a escada, semelhantes às fases apresentadas anteriormente.

No estado S1-A, as pernas estão elevadas em relação ao plano do solo e as esteiras são tracionadas conduzindo o robô em direção à escada. Quando as pernas frontais entram em contato com o primeiro degrau, o robô começa a ser elevado sobre o mesmo, de modo que o movimento das esteiras continua até que o ângulo de inclinação do robô se torna equivalente ao da escada. Neste instante, o movimento das esteiras é interrompido e se inicia o estado S1-B.

No estado S1-B, as pernas frontais são posicionadas de modo que sua parte inferior esteja no mesmo plano da base do chassi ( $\theta_2 = -\beta$ ), e as pernas traseiras são rotacionadas até entrarem em contato com o solo, em ambos os casos as pernas são controladas em malha aberta. No momento do contato, o ângulo de inclinação do robô ( $\alpha_R$ ) é reduzido, assim é possível determinar o instante do contato através da medida da variação da inclinação do robô. Após realizar as ações descritas, a detecção do contato das pernas traseiras com o solo provoca a transição para o estado S1-C.

No estado S1-C, entra em ação o controle de orientação do robô em malha fechada, atuando nas pernas traseiras em contato com o solo. O controle de orientação utiliza como sinal de referência o ângulo de inclinação desejada do robô equivalente ao ângulo de inclinação da escada ( $\alpha_S$ ). A realimentação do sistema é realizada através das medidas da inclinação do robô utilizando os sensores inerciais. Assim, baseando-se nesta medida e no valor de referência, o controle atua no ângulo ( $\theta_1$ ) de rotação das pernas traseiras, alterando indiretamente a inclinação ( $\alpha_R$ ) do robô. Simultaneamente ao controle de orientação, as esteiras são tracionadas em direção à subida da escada. A figura 5.10 ilustra um diagrama de blocos possível para o controle de orientação em malha fechada.

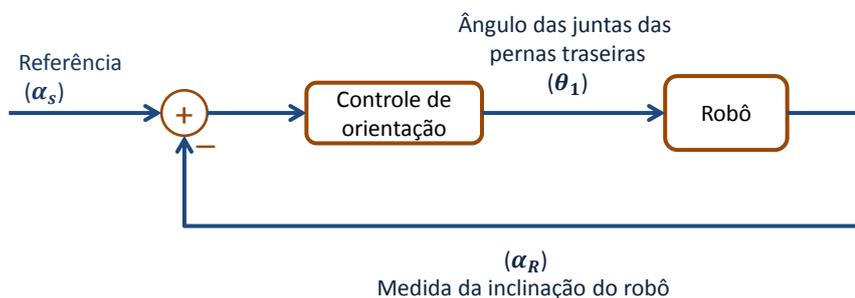


Figura 5.10: Diagrama de blocos do controle de orientação do robô em malha fechada, durante a transição do piso inferior para a escada.

Ressalta-se que, durante todo o período em que o controle atua nas pernas traseiras, as mesmas permanecem em contato com o solo, garantindo assim a ação do controle.

*Através do laser vertical ou do sensor Kinect, é possível identificar o final da escada e conseqüentemente inferir a posição ( $y$ ) do robô em relação ao sistema de coordenadas da escada  $O_S$ . Através desta medida, identificamos o instante em que o robô está completamente posicionado sobre os degraus, caracterizando o fim da tarefa de transição do piso inferior para a escada.*

*Para uma descrição mais formal do sistema a eventos discretos que pode representar a tarefa de transição do piso inferior para a escada, o leitor pode utilizar como referência *Cassandras & Lafortune (2008)*.*

### 5.3 Deslocamento na escada durante a subida

Finalizada a etapa de transição do piso inferior para escada, o robô encontra-se em contato apenas com os degraus, de modo que o seu ângulo de inclinação  $\alpha_R$  é igual ao ângulo de inclinação da escada  $\alpha_S$ . Em consequência da geometria do robô e da escada, além da hipótese de não escorregamento e baixas velocidades (capítulo 2), é possível aproximar o movimento do robô como em um plano inclinado definido pelas bordas dos degraus.

Tendo em vista as hipóteses realizadas para a interação entre o robô e a escada, o principal objetivo desta etapa é permitir que o robô execute o deslocamento sobre a mesma, mantendo sua posição equidistante em relação às laterais e ao mesmo tempo conservando a sua orientação na direção de subida. Desta forma, evitam-se colisões com as laterais da escada, escorregamentos em virtude de variações da orientação do robô, além de reduzir os riscos de tombamentos.

Foi avaliada uma estratégia de controle para permitir que o robô siga uma trajetória equidistante em relação às laterais da escada, mantendo assim a orientação desejada. A abordagem em questão é baseada em rastreamento de trajetórias, sendo descrita com detalhes no capítulo 6.

Além da estratégia de controle, é necessário definir a configuração das pernas robóticas a fim de auxiliar o deslocamento sobre a escada. Durante a travessia, o contato entre robô e escada é limitado aos locais onde as bordas dos degraus tocam o chassi e esteiras do robô, portanto para aumentar a superfície de contato, define-se o posicionamento das pernas de modo que a parte inferior das mesmas esteja alinhada com o plano referente a base do chassi do robô. A figura 5.11 ilustra o posicionamento das pernas descrito.

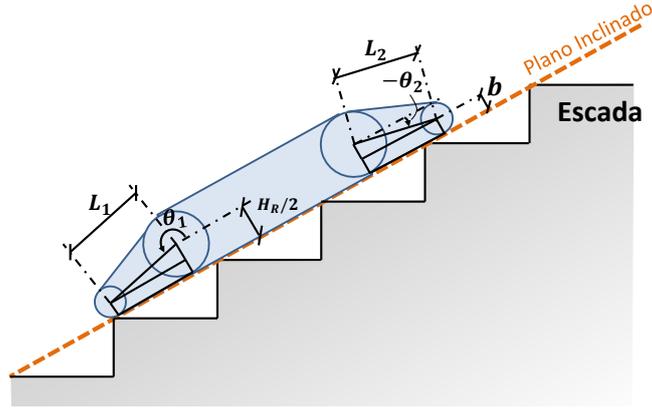


Figura 5.11: Configuração de posicionamento das pernas robóticas durante o deslocamento sobre a escada.

Portanto, com base na etapa anterior, a transição do piso inferior para a escada, os ângulos  $\theta_1$  e  $\theta_2$  devem ser mantidos os mesmos definidos ao final da fase S1-C. Desta maneira, com base na figura 5.11 e retomando as equações (5.3) e (5.4), os ângulos das juntas são definidos:

$$\begin{aligned}\sin(\beta) &= \frac{\frac{H_R}{2} - b}{L_2} \\ \beta &= \arcsin\left(\frac{\frac{H_R}{2} - b}{L_2}\right) \\ \theta_2 &= -\beta\end{aligned}\tag{5.12}$$

da mesma maneira para  $\theta_1$ , temos

$$\theta_1 = \pi + \beta\tag{5.13}$$

onde  $L_1 = L_2$ .

## 5.4 Aterrissagem suave sobre o piso superior

Após estabelecer as posições das pernas durante o deslocamento sobre a escada e conduzir o robô até os degraus finais, torna-se necessário definir qual deve ser a configuração das pernas para realizar uma aterrissagem suave no piso superior.

Durante a saída da escada, o robô que anteriormente deslocava-se no mesmo plano da escada, realiza uma abrupta mudança de ângulo de inclinação, alterando seu posicionamento para o mesmo plano horizontal do pavimento superior ( $\alpha_R = 0$ ). Esta queda abrupta no piso superior ocorre devido ao surgimento do momento das forças que atuam no robô.

No instante em que a posição horizontal do centro de gravidade do robô ultrapassa a posição horizontal do ponto de contato entre o robô e o último degrau da escada, o momento de forças originado cria uma aceleração angular que rotaciona o robô, impulsionando-o em direção ao plano horizontal do pavimento superior.

A colisão ocasionada pela mudança do plano da escada para o plano horizontal pode provocar danos ao chassi e pernas do robô, além de prejuízos aos componentes eletrônicos embarcados, falhas ou até mesmo o comprometimento total do sistema em determinados casos. Portanto, para evitar os prejuízos mencionados, foi definida uma estratégia que utiliza as pernas frontais do robô para sua aterrissagem suave.

Antes de descrever o procedimento adotado, é possível estimar o ponto de contato no robô em que ocorre o surgimento do momento das forças que atuam no mesmo e, conseqüentemente, provocam a mudança do ângulo de inclinação. Esta análise é realizada através do cálculo do centro de massa do robô e da geometria do problema. A figura 5.12 ilustra as posições dos sistemas de coordenadas  $O_0$ ,  $O_1$  e  $O_2$  para auxiliar o cálculo do centro de massa do robô.

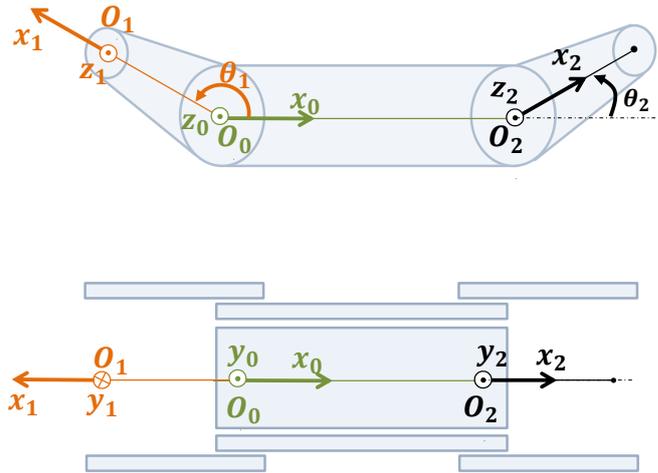


Figura 5.12: Vista lateral e superior dos sistemas de coordenadas em cada um dos três corpos rígidos que compõem o robô.

Novamente o robô é dividido em três corpos rígidos, pernas traseiras, chassi e pernas frontais, de modo que cada um apresenta seu próprio centro de massa. Assim,

- $(\tilde{l}_{c1})_1 = [-l_{c1x} \ 0 \ 0 \ 1]$ : Posição do centro de massa das pernas traseiras representada no sistema de coordenadas  $O_1$ ;
- $(\tilde{l}_{c0})_0 = [l_{c0x} \ l_{c0y} \ l_{c0z} \ 1]$ : Posição do centro de massa do chassi representada no sistema de coordenadas  $O_0$ ;
- $(\tilde{l}_{c2})_2 = [l_{c2x} \ 0 \ 0 \ 1]$ : Posição do centro de massa das pernas frontais representada no sistema de coordenadas  $O_2$ ;
- $m_c = m_{c1} + m_{c0} + m_{c2}$ : Massa total do robô;

e)  $m_{c1}, m_{c0}, m_{c2}$ : Massas das pernas traseiras, chassi e pernas frontais, respectivamente;

**Observação 5.2** *Os centros de massa  $(\tilde{l}_{c0})_0, (\tilde{l}_{c1})_1, (\tilde{l}_{c2})_2$  pertencem ao  $\mathbb{R}^4$ , pois estão sendo representados na forma homogênea. Desta maneira, para um ponto genérico  $\tilde{p} = [* \ * \ * \ 1]$  representado na forma homogênea, os três primeiros elementos indicam as coordenadas  $x, y$  e  $z$  do respectivo ponto, e o último elemento unitário não apresenta nenhum significado físico, sendo utilizado apenas para tornar consistentes as transformações de coordenadas realizadas através das matrizes de transformações homogêneas.*

*Portanto, para simplificar a notação, os vetores e pontos a seguir representados na forma homogênea apresentam um til ( $\tilde{\ })$  na parte superior da variável, e todos os outros pontos ou vetores sem marcação, estão representados na forma cartesiana.*

Deve ser ressaltado que os subscritos  $(\ )_0, (\ )_1$  e  $(\ )_2$  indicam que a variável está sendo representada no sistema de coordenadas com o mesmo número respectivo ( $O_0, O_1$  e  $O_2$ ).

Para representar os centros de massa em um único sistema de coordenadas é necessário retomar as transformações homogêneas,  $T_{10}$  e  $T_{02}$ :

$$T_{10} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & -L_1 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

$$T_{02} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & L_0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

Por simplificação, todos os centros de massa são representados no sistema de coordenadas  $O_0$ , por conseguinte torna-se necessário obter a transformação  $T_{01}$  inversa à transformação  $T_{10}$ :

$$\text{definindo } R_{10} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad p_{10} = \begin{bmatrix} -L_1 \\ 0 \\ 0 \end{bmatrix}$$

a transformação  $T_{01}$  é obtida a partir da seguinte operação:

$$T_{01} = \begin{bmatrix} (R_{10})^T & -(R_{10})^T p_{10} \\ 0 & 1 \end{bmatrix}$$

$$T_{01} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & L_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & L_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.14)$$

Os centros de massa  $(\tilde{l}_{c1})_0$  e  $(\tilde{l}_{c2})_0$  representados no sistema de coordenadas  $O_0$ :

$$(\tilde{l}_{c1})_0 = T_{01}(\tilde{l}_{c1})_1 = \begin{bmatrix} (L_1 - l_{c1x}) \cos(\theta_1) \\ (L_1 - l_{c1x}) \sin(\theta_1) \\ 0 \\ 1 \end{bmatrix} \quad (5.15)$$

da mesma maneira,

$$(\tilde{l}_{c2})_0 = T_{02}(\tilde{l}_{c2})_2 = \begin{bmatrix} L_0 + l_{c2x} \cos(\theta_2) \\ l_{c2x} \sin(\theta_2) \\ 0 \\ 1 \end{bmatrix} \quad (5.16)$$

Portanto, a posição do centro de massa  $(\tilde{p}_{CM})_0$  do robô em relação ao sistema de coordenadas  $O_0$  é dada por:

$$(\tilde{p}_{CM})_0 = \frac{1}{m_c} [(\tilde{l}_{c1})_0 m_{c1} + (\tilde{l}_{c0})_0 m_{c0} + (\tilde{l}_{c2})_0 m_{c2}] \quad (5.17)$$

colocando (5.17) em termos das transformações homogêneas  $T_{01}$  e  $T_{02}$ , temos:

$$(\tilde{p}_{CM})_0 = \frac{1}{m_c} [T_{01}(\tilde{l}_{c1})_1 m_{c1} + (\tilde{l}_{c0})_0 m_{c0} + T_{02}(\tilde{l}_{c2})_2 m_{c2}] \quad (5.18)$$

Em seguida, é realizado o cálculo das coordenadas do ponto de contato  $(\tilde{p}_N)_0$  entre o último degrau da escada e a parte inferior do chassi do robô. Sendo  $(\tilde{p}_N)_0$  também representado em relação à  $O_0$ . A figura 5.13 ilustra o ponto em questão.

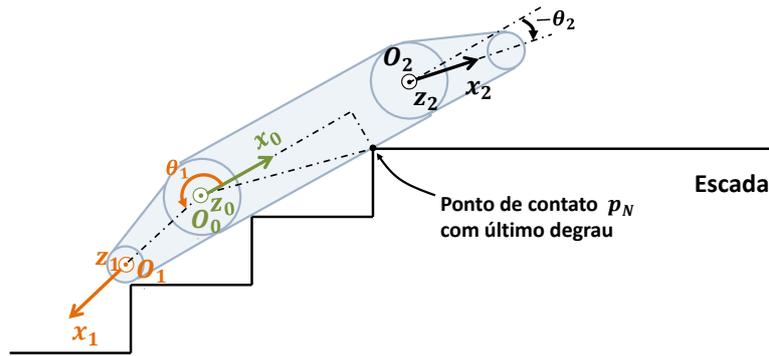


Figura 5.13: Exemplo do robô próximo ao final da escada e indicação do ponto de contato entre o chassi e o último degrau

Como pode ser observado na ilustração 5.13, o ponto de contato  $(\tilde{p}_N)_0$  pode ser representado por:

$$(\tilde{p}_N)_0 = \begin{bmatrix} p_{Nx} \\ -\frac{H_R}{2} \\ 0 \\ 1 \end{bmatrix} \quad (5.19)$$

Quando o robô estiver na iminência de realizar a mudança do plano da escada para o plano do piso superior, as únicas forças que atuam no robô são: a força da gravidade  $F_g$  aplicada ao centro de massa (CM) e a força  $N$  que a borda do último degrau exerce sobre o chassi e esteiras. Portanto, imediatamente antes de surgir o momento das forças que impulsiona o robô para o plano horizontal, as coordenadas horizontais do centro de massa do robô e do ponto de contato com o último degrau se tornam equivalentes quando projetadas no plano horizontal. A figura 5.14 ilustra o limite mencionado.

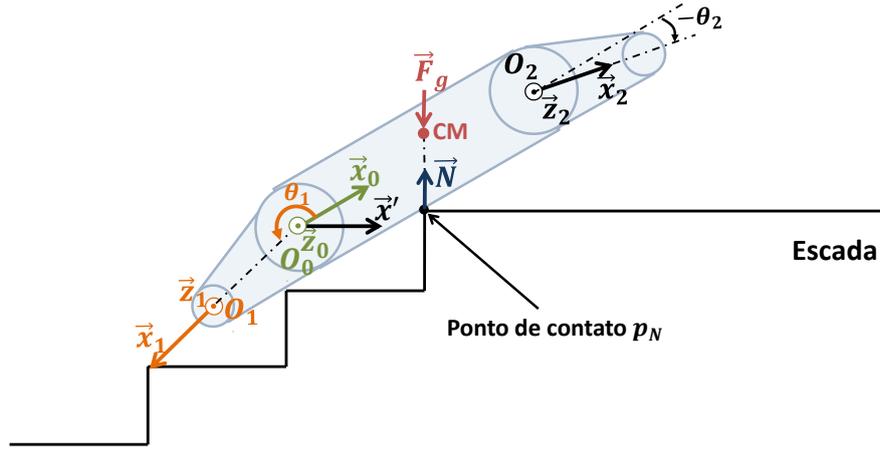


Figura 5.14: Robô posicionado no ponto limite antes da mudança do plano da escada para o plano horizontal do piso superior

Como pode ser observado na figura anterior, para definir a relação limite entre o ponto de contato  $(\tilde{p}_N)_0$  e o centro de massa  $(\tilde{p}_{CM})_0$  basta projetar ambos os pontos sobre o eixo horizontal  $x'$ , comparando em seguida os valores encontrados.

A projeção é definida por uma rotação elementar  $R_p$  ao redor do eixo  $z_0$  em que o ângulo de rotação deve ser o ângulo de inclinação da escada  $\alpha_S$ . Para não sobrecarregar a nomenclatura o subscrito  $( )_0$  foi suprimido das variáveis  $\tilde{p}_N$  e  $\tilde{p}_{CM}$  nos próximos passos.

A rotação  $R_p$  é definida por:

$$R_p = \begin{bmatrix} \cos(\alpha_S) & -\sin(\alpha_S) & 0 \\ \sin(\alpha_S) & \cos(\alpha_S) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.20)$$

assim, a transformação homogênea  $T_{R_p}$  que define a rotação  $R_p$  é dada por:

$$T_{R_p} = \begin{bmatrix} \cos(\alpha_S) & -\sin(\alpha_S) & 0 & 0 \\ \sin(\alpha_S) & \cos(\alpha_S) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.21)$$

consequentemente a projeção do ponto  $\tilde{p}_{CM}$  no eixo horizontal  $x'$  é definida por:

$$T_{R_p} \tilde{p}_{CM} = \begin{bmatrix} (T_{R_p} \tilde{p}_{CM})_{x'} \\ * \\ * \\ 1 \end{bmatrix} \quad (5.22)$$

substituindo  $\tilde{p}_{CM}$  pela equação (5.18) temos,

$$T_{R_p} \tilde{p}_{CM} = \frac{1}{m_c} T_{R_p} [T_{01}(\tilde{l}_{c1})_1 m_{c1} + (\tilde{l}_{c0})_0 m_{c0} + T_{02}(\tilde{l}_{c2})_2 m_{c2}] \quad (5.23)$$

Como apenas a coordenada horizontal interessa para definir o momento em que o centro de gravidade ultrapassa o ponto de contato com o último degrau da escada, o subscrito  $( )_{x'}$  indica que apenas o primeiro elemento do vetor foi utilizado.

Da mesma maneira, para calcular a coordenada horizontal do ponto  $\tilde{p}_N$ , temos:

$$T_{R_p} \tilde{p}_N = \begin{bmatrix} (T_{R_p} \tilde{p}_N)_{x'} \\ * \\ * \\ 1 \end{bmatrix} \quad (5.24)$$

substituindo  $\tilde{p}_N$  apresentado em (5.19) na equação anterior, teremos:

$$(R_p p_N)_{x'} = p_{Nx} \cos(\alpha_S) + \frac{H_R}{2} \sin(\alpha_S) \quad (5.25)$$

Igualando os pontos  $(T_{R_p} \tilde{p}_N)_{x'}$  e  $(T_{R_p} \tilde{p}_{CM})_{x'}$  apresentados, é possível chegar a seguinte equação:

$$p_{Nx} \cos(\alpha_S) + \frac{H_R}{2} \sin(\alpha_S) = [(L_1 - l_{c1x}) \cos(\alpha_S + \theta_1) m_{c1} + l_{c2x} \cos(\alpha_S + \theta_2) m_{c2} + L_0 \cos(\alpha_S) m_{c2} - l_{c0y} \sin(\alpha_S) m_{c0} + l_{c0x} \cos(\alpha_S) m_{c0}] \frac{1}{m_c} \quad (5.26)$$

Isolando o termo  $p_{Nx}$ , temos:

$$p_{Nx} = -\frac{H_R}{2} \tan(\alpha_S) + [(L_1 - l_{c1x}) \cos(\alpha_S + \theta_1) m_{c1} + l_{c2x} \cos(\alpha_S + \theta_2) m_{c2} + L_0 \cos(\alpha_S) m_{c2} - l_{c0y} \sin(\alpha_S) m_{c0} + l_{c0x} \cos(\alpha_S) m_{c0}] \left( \frac{1}{\cos(\alpha_S) m_c} \right) \quad (5.27)$$

Portanto, quando  $p_{Nx}$  for menor que a expressão encontrada na equação (5.27), o momento gerado pela força da gravidade provoca a rotação do robô em direção ao plano do piso superior. Lembrando que  $p_{Nx}$  representa a coordenada  $x$  do ponto de

contato  $\tilde{p}_N$  entre o último degrau da escada e o robô, sendo representado no sistema de coordenadas  $O_0$ .

Durante o deslocamento sobre a escada, não são realizadas medidas da posição do robô na direção de subida  $y_S$ , fato que impede a obtenção da medida direta da coordenada  $p_{Nx}$ . Desta forma, não é possível definir o momento exato em que o centro de massa do robô ultrapassa o ponto de contato  $p_N$ .

Para solucionar o problema da aterrissagem suave sobre o piso superior sem a medida direta da posição  $y$  do robô, foi proposta uma abordagem de avanço gradual através das esteiras e medições das mudanças de orientação do robô em função do contato das pernas frontais com o pavimento superior.

Assim como na etapa de transição do piso inferior para a escada, as entradas do sistema são os valores de referência para os ângulos das juntas das pernas robóticas e valores de referência para as velocidades das esteiras, e as saídas são as medidas destas variáveis.

Como pode ser observado na figura 5.15, quando as pernas frontais entram em contato com o piso superior e continuam o movimento de rotação no mesmo sentido, a inclinação do robô, que era equivalente à do plano da escada, é elevada.

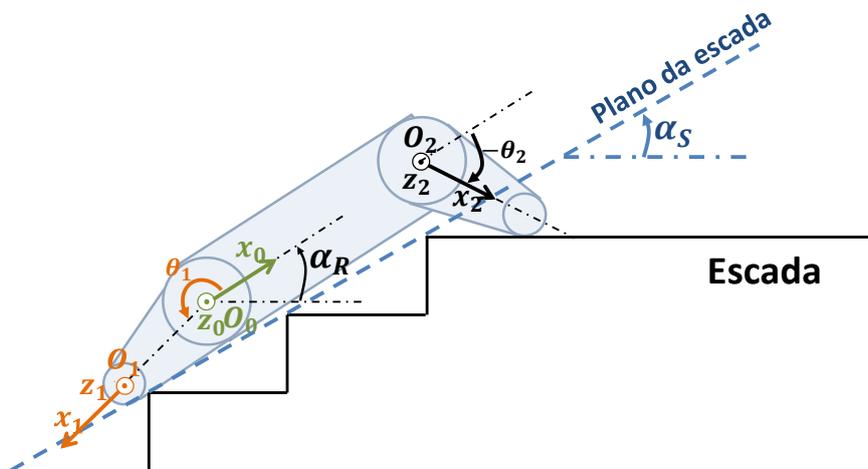


Figura 5.15: Robô com pernas frontais em contato com pavimento superior

Para executar a abordagem proposta, o robô realiza movimentos graduais com as esteiras na direção de avanço sobre a escada. As distâncias de deslocamento são sempre iguais, de modo que após realizar cada um desses avanços, o robô inicia o movimento de rotação das pernas frontais na direção de contato com o piso superior.

Assim que as pernas entram em contato com o solo, o sensor inercial detecta o aumento da inclinação do robô ( $\alpha_R$ ) que se torna maior que o ângulo de inclinação da escada ( $\alpha_S$ ). Após detectar a variação de inclinação do robô, as pernas frontais são movimentadas no sentido contrário por um ângulo constante  $\Delta\theta_2$  a ser definido.

Retomada a inclinação inicial do robô equivalente a da escada ( $\alpha_R = \alpha_S$ ), o robô realiza novamente um avanço com as esteiras e todo o procedimento é realizado novamente. Os passos descritos são repetidos até que o sensor inercial do robô detecte uma inclinação menor que a inclinação da escada, fato este que indica a mudança do robô do plano da escada para o plano do piso superior.

Durante a execução dos passos descritos, caso o ângulo das pernas frontais atinja o valor de  $\theta_2 = -\pi/2$  ( $-90^\circ$ ), a repetição do procedimento é interrompida e o robô apenas realiza o movimento contínuo de avanço com as esteiras até que a inclinação do robô se torne inferior à inclinação da escada.

O pseudocódigo 6 do método descrito é apresentado a seguir:

---

**Pseudocódigo 6** Aterrissagem suave no piso superior

---

```

1: while  $\alpha_R \geq \alpha_S$  do
2:   Mover esteiras na direção de subida por uma distância ( $d_{sub}$ ) fixa
3:   if  $\theta_2 > -\frac{\pi}{2}$  then
4:     repeat mover pernas frontais em direção ao solo (rotação negativa)
5:     until ( $\alpha_R > \alpha_S$  or  $\theta_2 \leq -\frac{\pi}{2}$ )
6:     if  $\alpha_R > \alpha_S$  then
7:       Elevar pernas frontais ( $\theta_2 = \theta_2 + \Delta\theta_2$ )
8:     end if
9:   end if
10: end while
11: Robô aterrissou no piso superior

```

---

**Observação 5.3** *Antes de executar o pseudocódigo 6 apresentado para realizar a aterrissagem suave no piso superior, é necessário definir o instante em que o algoritmo deve ser iniciado. Para tal, utiliza-se o laser posicionado verticalmente à frente do robô, para detectar o final da escada e indicar o início do método. Durante o deslocamento sobre a escada, as varreduras verticais do laser detectam apenas os degraus da escada. Quando o robô alcança os últimos degraus, o laser passa a detectar uma linha contínua e longa que corresponde ao piso superior. Desta maneira, para identificar o final da escada, basta realizar a extração de linhas dos dados do laser utilizando o método Ransac apresentado na seção 4.2.1 e verificar o momento em que é detectada uma linha horizontal nas varreduras com dimensão maior que as dimensões dos degraus, indicando, assim, o piso superior.*

**Observação 5.4** *Ressalta-se que assim como na etapa de transição do piso inferior para a escada, a aterrissagem suave no piso superior também pode ser apresentada em uma abordagem utilizando sistemas a eventos discretos. É possível utilizar novamente uma máquina de estados para coordenar as ações do robô, e em cada estado pode ser utilizada uma malha de controle de orientação do robô fazendo uso do contato das pernas frontais com piso superior, semelhante à abordagem apresentada na*

observação 5.1.

**Observação 5.5** *O trabalhos desenvolvidos em Kalantari et al. (2009), Wang et al. (2012), Zhang et al. (2011) apresentam configurações fixas das pernas robóticas para realizar as tarefas de transição do piso inferior para a escada e aterrissagem no piso superior. Diferentemente dos trabalhos mencionados, o método proposto nesta dissertação utiliza configurações variáveis das pernas robóticas baseando-se no ângulo de inclinação do robô. Desta maneira, durante a transição do piso inferior para a escada, evita-se o tombamento do robô no início da subida sobre os primeiros degraus, e na transição da escada para o piso superior, o algoritmo proposto permite realizar a aterrissagem suave, procurando manter as pernas frontais em contato com o solo sem utilizar sensores de proximidade adicionais para indicar o momento da transição.*

## 5.5 Simulações

A abordagem utilizada para realizar cada uma das etapas da locomoção sobre a escada foi avaliada através de simulações utilizando o software Matlab.

A primeira etapa representada pela transição do piso inferior para a escada e a terceira etapa para aterrissagem suave no piso superior foram simuladas e apresentadas nesta seção, de modo que a segunda etapa que consiste no deslocamento sobre a escada durante a subida foi simulada no capítulo 6.

Os parâmetros do robô e da escada utilizados nas simulações foram definidos da seguinte maneira:

- a) Comprimento das pernas ( $L_1 = L_2 = 0.255$  m);
- b) Distância entre  $O_0$  e  $O_2$  ( $L_0 = 0.46$  m);
- c) Distância entre as esteiras ( $d_R = 0.6$  m);
- d) Altura do robô ( $H_R = 0.2$  m);
- e) Massa das pernas traseiras ( $m_{c1} = 8$ kg), pernas frontais ( $m_{c2} = 8$ kg) e chassi ( $m_{c0} = 46$ kg);
- f) Centro de massa das pernas traseiras ( $(l_{c1})_1 = [-0.17 \ 0 \ 0]$  m), pernas frontais ( $(l_{c2})_2 = [0.085 \ 0 \ 0]$  m) e chassi ( $(l_{c0})_0 = [0.23 \ 0 \ 0]$  m);
- g) Altura dos degraus da escada ( $h_S = 0.18$  m);
- h) Profundidade dos degraus ( $d_S = 0.30$  m);
- i) Raio da circunferência menor que compõe a perna robótica ( $b = 0.02$  m);

### 5.5.1 Simulação da primeira etapa: Transição do piso inferior para a escada

Para definir qual a altura máxima dos degraus que o robô é capaz de subir, basta utilizar a equação (5.11) apresentada anteriormente. Portanto, utilizando os parâmetros da simulação, obtêm-se a altura máxima do degrau  $h_{S,max} = 0.34$  m.

Após solucionar a equação (5.9) e utilizar os parâmetros de simulação, o ângulo mínimo para que o robô seja capaz de iniciar a subida sobre a escada é definido por  $\theta_{2,min} = 21.9^\circ$

Portanto, com  $h_{S,max} > h_S$  o robô é capaz de iniciar a locomoção sobre os primeiros degraus.

Os ângulos das juntas  $\theta_1$  e  $\theta_2$ , utilizados em cada uma das fases S1-A, S1-B e S1-C realizadas na primeira etapa, são apresentados na tabela 5.1 abaixo:

	$\theta_1$	$\theta_2$
<b>S1-A</b>	$140^\circ$	$40^\circ$
<b>S1-B</b>	$198.28^\circ$	$40^\circ$
<b>S1-C</b>	$198.28^\circ$	$-18.28^\circ$

Tabela 5.1: Ângulos das juntas utilizados em cada fase S1-A, S1-B e S1-C.

Os gráficos apresentados na figura 5.16 ilustram os movimentos das pernas e a elevação do robô sobre os primeiros degraus durante a simulação da fase S1-A.

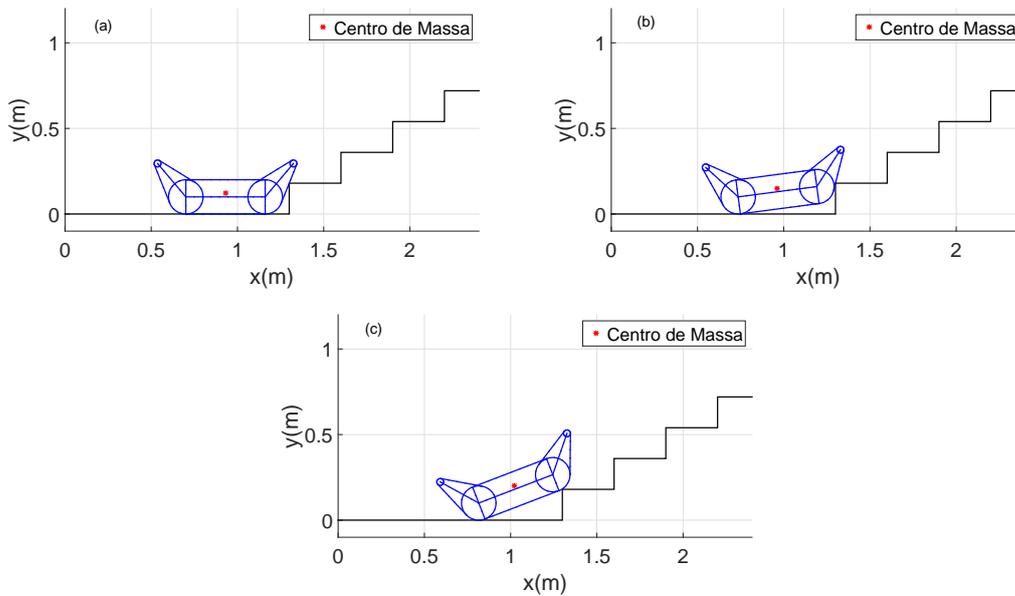


Figura 5.16: Simulação da primeira etapa da locomoção sobre a escada, fase S1-A.

Em seguida, são demonstrados na figura 5.17 a locomoção do robô realizando a fase S1-B.

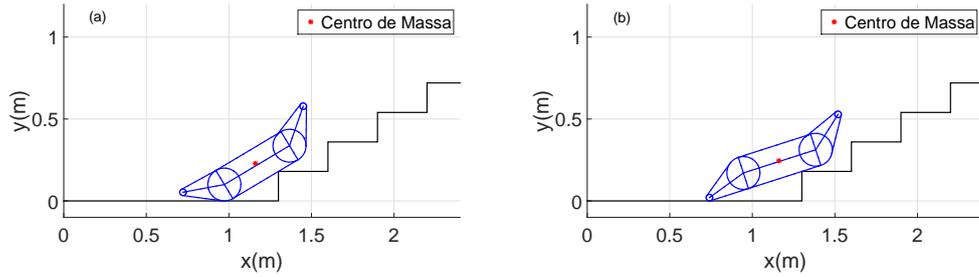


Figura 5.17: Simulação da primeira etapa da locomoção sobre a escada, fase S1-B.

Por fim, são apresentados na figura 5.18 os gráficos referentes à execução da fase S1-C.

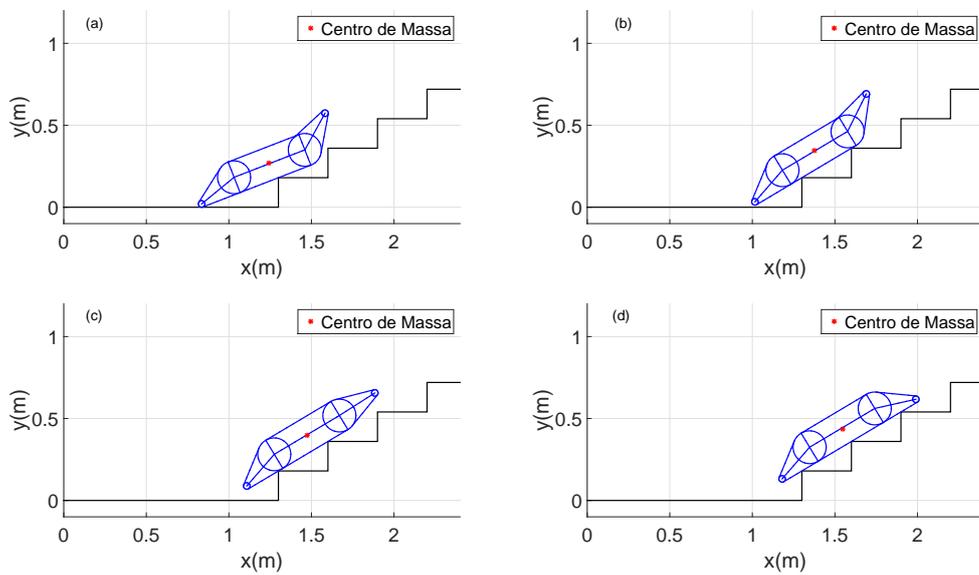


Figura 5.18: Simulação da primeira etapa da locomoção sobre a escada, fase S1-C.

A figura 5.19 ilustra a variação do ângulo de inclinação  $\alpha_R$  do robô, durante a primeira etapa da locomoção sobre a escada.

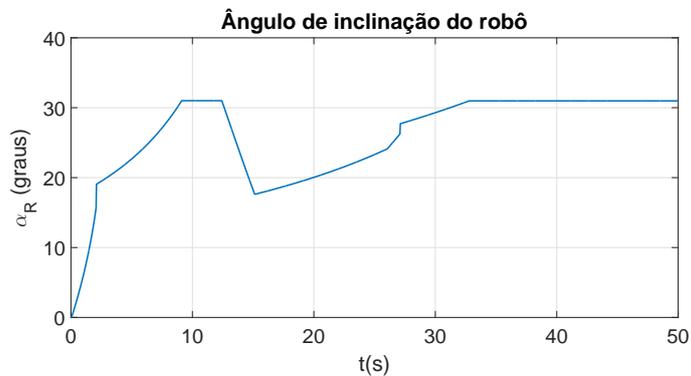


Figura 5.19: Ângulo de inclinação do robô obtido durante a simulação da primeira etapa de transição do piso inferior para a escada.

Como pode ser observado nos gráficos anteriores, as configurações propostas para

as pernas robóticas e a metodologia para realizar a transição do piso inferior para a escada permitem que o robô inicie o movimento sobre os primeiros degraus, estando completamente apoiado sobre escada, atingindo, portanto, o objetivo desta etapa.

### 5.5.2 Simulação da terceira etapa: Aterrissagem suave no piso superior

A terceira etapa, que consiste na aterrissagem suave sobre o piso superior, também foi simulada utilizando os mesmos parâmetros do robô e escada definidos anteriormente.

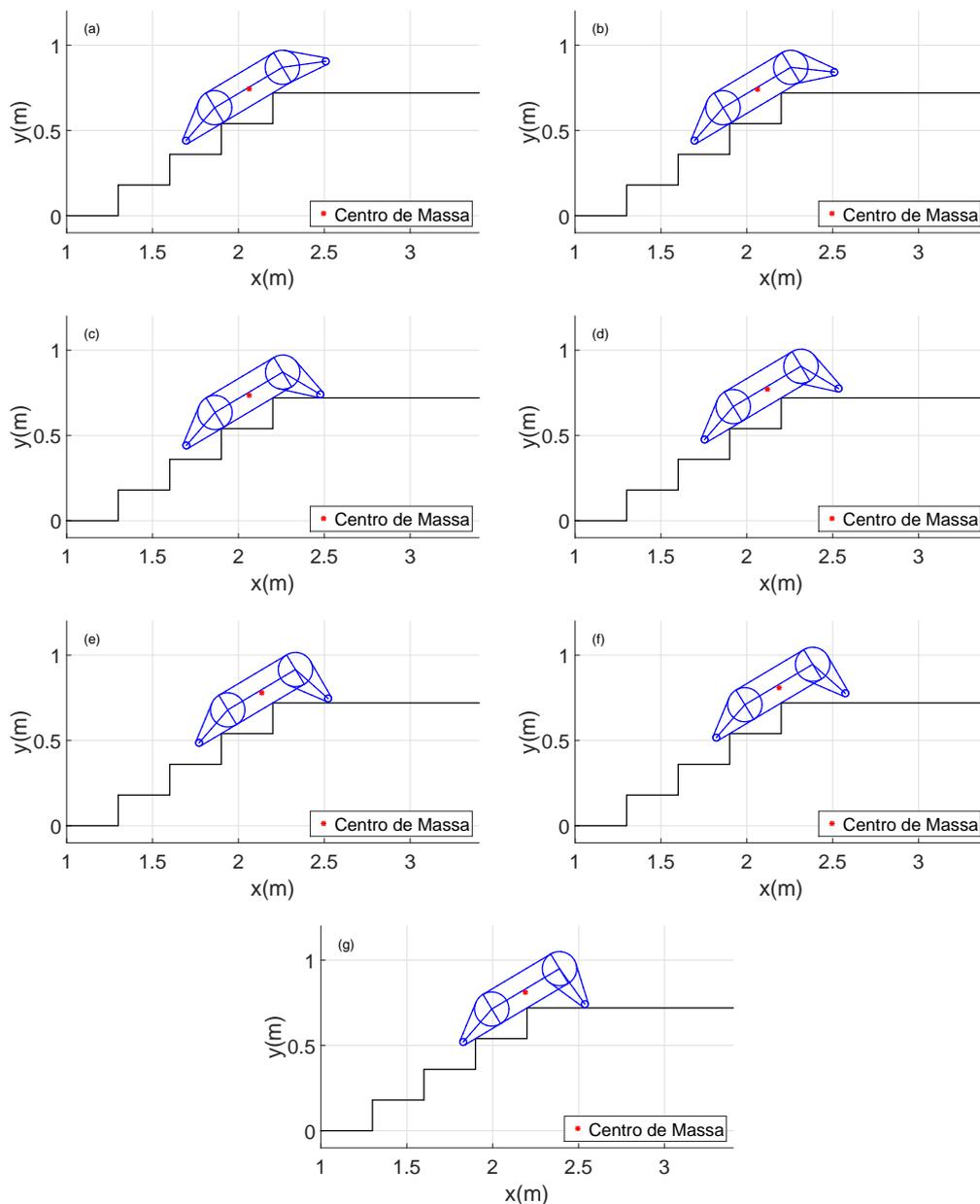


Figura 5.20: Gráficos referentes a simulação da etapa de aterrissagem suave sobre o piso superior.

Os gráficos 5.20 (a)-(g), apresentam os resultados da simulação do pseudocódigo 6 utilizado para solucionar o problema de aterrissagem suave no piso superior. Como pode ser observado no gráfico (g), no instante em que a posição horizontal do centro de massa do robô está alinhada com a posição horizontal do ponto de contato entre o último degrau e o chassi, a perna frontal está em contato com o pavimento superior, garantindo uma transição suave da escada para o solo. Desta maneira, o método proposto é capaz de realizar a aterrissagem suave sobre o piso superior como é demonstrado pelos resultados obtidos.

## 5.6 Conclusões

Este capítulo apresentou a metodologia para realizar a locomoção em escadas durante a subida, sendo dividida em três etapas: transição do piso inferior para a escada, deslocamento propriamente dito sobre a escada e aterrissagem suave no piso superior.

As etapas foram detalhadas e para cada uma delas foi definida a configuração das pernas robóticas necessária à solução do problema.

A cinemática direta do robô com esteiras do tipo EOD também foi descrita, a fim de auxiliar a definição das configurações das pernas robóticas mencionadas.

A primeira etapa, a transição do piso inferior para a escada, foi dividida nas fases S1-A, S1-B e S1-C, sendo cada uma delas responsável por realizar diferentes configurações das pernas robóticas descritas no texto.

Para a última etapa, a aterrissagem suave no piso superior, foi definido um método baseado no avanço gradual do robô através das esteiras e nas medidas das variações da inclinação do mesmo, ocasionadas pelo contato entre as pernas frontais e o pavimento superior.

A primeira etapa e a terceira foram simuladas para avaliar a abordagem proposta, de modo que ambas apresentaram resultados satisfatórios. Já as simulações da segunda etapa são apresentadas no capítulo 6.

# Capítulo 6

## Controle para locomoção em escadas durante a subida

A tarefa principal considerada no problema de locomoção em escadas é o deslocamento propriamente dito do robô sobre a mesma durante a subida, iniciando nos degraus iniciais até a posição final antes do piso superior.

Desta maneira, o deslocamento semiautônomo em escadas requer a definição de uma estratégia de controle que permita realizar a tarefa com segurança.

Grande parte das abordagens existentes (Helmick et al. 2002, Kalantari et al. 2009, Zhang et al. 2011) buscam manter o robô em posição equidistante das laterais da escada, além de assegurar que sua orientação não seja desviada da direção de subida da mesma. Desta maneira, evitam-se colisões com paredes laterais ou balaustres, além de reduzir a possibilidade de tombamentos causados pelo desvio de orientação do robô em relação à direção de subida da escada.

Nesta dissertação, o objetivo da abordagem de controle é definido da mesma maneira: realizar o deslocamento do robô durante a subida, mantendo sua posição equidistante em relação às laterais da escada e a orientação no valor desejado. Deste modo, uma estratégia de controle cinemático é aplicada para a solução do problema, sendo baseada em rastreamento de trajetórias.

Utilizando a hipótese de não escorregamento das esteiras em contato com os degraus, além da hipótese de velocidades baixas, foi considerado o deslocamento do robô sobre a escada como o movimento em um plano inclinado. A figura 6.1 ilustra o robô deslocando-se sobre a escada.

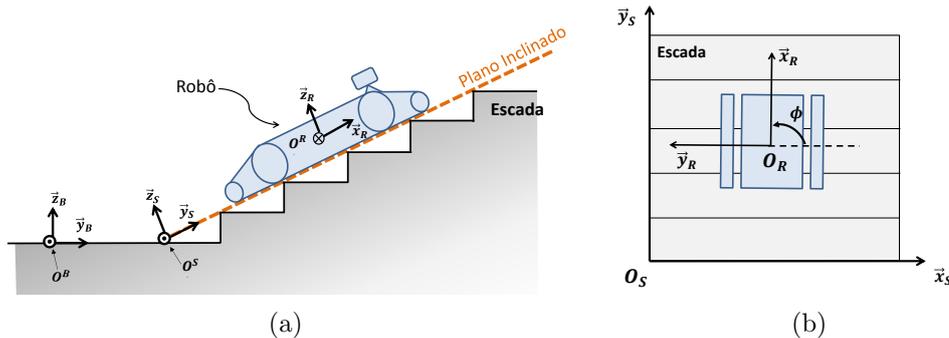


Figura 6.1: Vista lateral (a) e vista superior (b) representando a locomoção do robô sobre a escada modelada como um plano inclinado

Na figura 6.1,  $O_B$  representa a origem do sistema de coordenadas inercial fixado ao plano do solo. Já o termo  $O_S$  representa a origem do sistema de coordenadas da escada, de modo que o plano  $XY$  do sistema  $O_S$  está fixado ao plano inclinado que modela a escada, ou seja, são coplanares.

Para a locomoção sobre a escada, o sistema de coordenadas  $O_S$  é definido como a referência. O sistema de coordenadas móvel  $O_R$ , por sua vez, é fixado ao centro do chassi do robô, movendo-se junto ao sistema robótico em questão. Desta maneira, o plano  $XY$  do sistema  $O_R$  se mantém paralelo ao plano  $XY$  da sistema  $O_S$ , durante o deslocamento sobre a escada.

Definidas as considerações referentes ao movimento do robô sobre a escada, a seção 6.1 descreve o modelo cinemático do robô móvel e a seção 6.2, por sua vez, descreve a estratégia de controle baseada em rastreamento de trajetórias para a locomoção em escadas durante a subida.

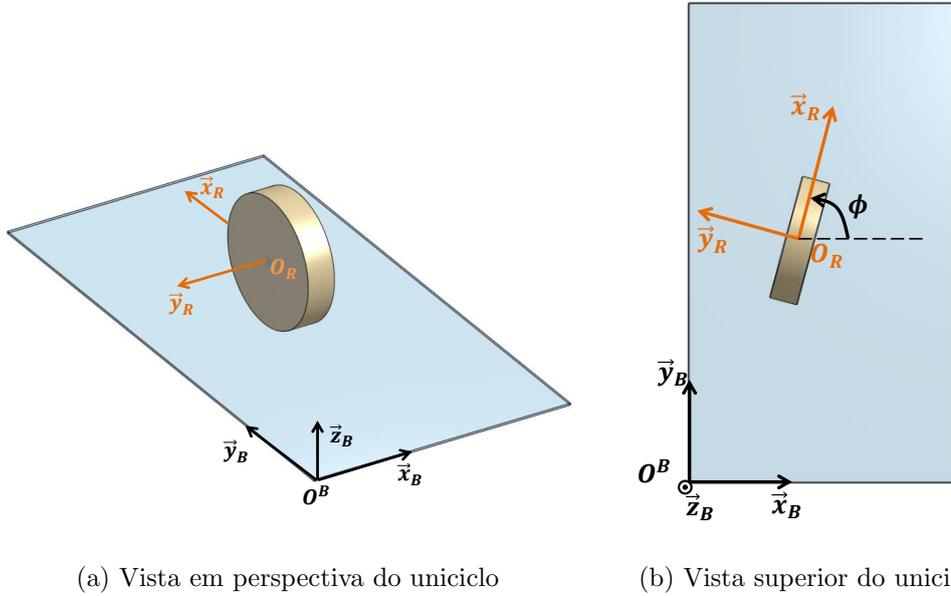
## 6.1 Modelo cinemático do robô móvel

O deslocamento dos robôs móveis do tipo EOD é realizado usualmente através de duas esteiras laterais, de forma que as trajetórias lineares e angulares são executadas através da rotação diferencial das referidas esteiras. Deste modo, a semelhança entre os movimentos dos robôs móveis com rodas e direção diferencial e robôs móveis com esteiras, nos permite modelar ambos os sistemas da mesma maneira.

O modelo cinemático dos robôs móveis com direção diferencial pode ser obtido a partir do caso mais simples que é o unicyclo. O unicyclo consiste em um veículo com apenas uma roda direcionável. Usualmente, os veículos com rodas são caracterizados por apresentar restrições cinemáticas holonômicas e não-holonômicas. Sendo desta maneira, caracterizados como sistemas não-holonômicos por natureza.

A configuração de um unicyclo pode ser definida completamente pelo vetor de coordenadas generalizadas  $\mathbf{q} = [x \ y \ \phi] \in \mathcal{C}$ , onde  $\mathcal{C}$  é o espaço de configuração que define todas as configurações possíveis do veículo, o par  $(x, y)$  representa as

coordenadas cartesianas do ponto de contato da roda com o solo, e  $\phi$  representa a orientação da roda em relação ao eixo  $x_B$  do sistema de coordenadas inercial  $O_B$ . Assim, para o caso do unicyclo  $\mathcal{C} = \mathbb{R}^2 \times SO(2)$ , onde  $SO(2)$  é o grupo especial ortogonal das rotações em duas dimensões. A figura 6.2 ilustra o unicyclo durante o movimento e as variáveis que definem sua configuração.



(a) Vista em perspectiva do unicyclo

(b) Vista superior do unicyclo

Figura 6.2: Exemplo do unicyclo em movimento

Utilizando o vetor de coordenadas generalizadas  $\mathbf{q}$ , as restrições holonômicas são aquelas que podem ser expressas na forma:

$$h_i(\mathbf{q}) = 0 \quad , \quad \forall i = 1, 2, 3, \dots, k \quad (6.1)$$

onde  $h_i(\mathbf{q})$  é uma função que mapeia  $\mathcal{C} \rightarrow \mathbb{R}$ .

Já as restrições não-holonômicas são aquelas não integráveis, podendo apenas ser expressas sob a forma:

$$a_i(\mathbf{q}, \dot{\mathbf{q}}) = 0 \quad , \quad \forall i = 1, 2, 3, \dots, k \quad (6.2)$$

Ao considerarmos o unicyclo como um disco com movimento restrito ao plano do solo, de modo que não há escorregamentos, a restrição não-holonômica de rotação pura pode ser expressa da seguinte maneira (Siciliano et al. 2009):

$$\dot{x} \sin(\phi) - \dot{y} \cos(\phi) = [\sin(\phi) \quad -\cos(\phi) \quad 0] \dot{\mathbf{q}} = 0 \quad (6.3)$$

implicando que, na ausência de escorregamento, a componente de velocidade do unicyclo na direção  $y_R$  (vide figura 6.2) do seu próprio sistema de coordenadas  $O_R$  é

zero.

A restrições não-holonômicas como (6.3) podem ser expressas na forma de *Pi-faffian*:

$$\mathbf{a}^T(\mathbf{q})\dot{\mathbf{q}} = 0 \quad (6.4)$$

portanto,

$$\mathbf{a}^T(\mathbf{q}) = [\sin(\phi) \quad -\cos(\phi) \quad 0] \quad (6.5)$$

Desta maneira, é possível concluir que as velocidades generalizadas  $\dot{\mathbf{q}}$  são restritas ao espaço nulo de  $\mathbf{a}^T(\mathbf{q})$ .

Escolhendo duas bases para o espaço nulo da matriz  $\mathbf{a}^T(\mathbf{q})$  do unicycle (6.5):

$$g_1(\mathbf{q}) = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ 0 \end{bmatrix}, \quad g_2(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (6.6)$$

Podemos, então, escolher qualquer combinação linear dos vetores  $g_1(\mathbf{q})$  e  $g_2(\mathbf{q})$  para produzir o vetor de velocidades generalizadas  $\dot{\mathbf{q}}$ . Escolhendo aquela que fornece uma interpretação física para o caso do unicycle, teremos:

$$\dot{\mathbf{q}} = \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ 0 \end{bmatrix} v_c + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega_c \quad (6.7)$$

onde  $v_c$  é a velocidade linear da roda na direção  $x_R$  e  $\omega_c$  é a velocidade angular da roda em torno do eixo  $z_B$ . Portanto, (6.7) define o modelo cinemático do unicycle, a partir da restrição não-holonômica de não escorregamento.

Colocando o modelo cinemático do unicycle (6.7) na forma matricial:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \quad (6.8)$$

Desta maneira,  $v_c$  e  $\omega_c$  são as entradas do sistema responsáveis por alterar a configuração do unicycle.

O modelo cinemático do unicycle (6.8) é equivalente ao modelo dos robô móveis com duas rodas e direção diferencial, bem como o modelo do robô móvel com esteiras. Entretanto, é possível reescrevê-lo em termos das velocidades das rodas ou esteiras.

Exprimindo as velocidades linear  $v_c$  e angular  $\omega_c$  em termos das velocidades

angulares das esteiras esquerda  $\omega_l$  e direita  $\omega_r$ , temos:

$$\begin{aligned} v_c &= \frac{r_e}{2} (\omega_r + \omega_l) \\ \omega_c &= \frac{r_e}{d_R} (\omega_r - \omega_l) \end{aligned} \quad (6.9)$$

onde a)  $r_e$ : Raio da engrenagem principal que traciona as esteiras; e b)  $d_R$ : Distância entre o centro das duas esteiras. A figura 6.3 ilustra o movimento de um robô com esteiras e direção diferencial.

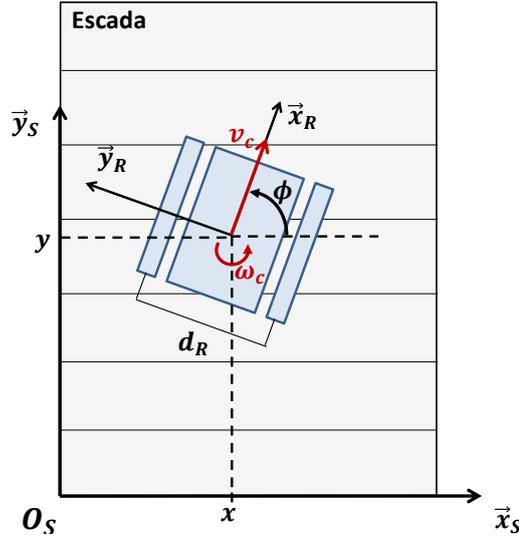


Figura 6.3: Robô móvel com esteiras e direção diferencial

Substituindo a relação entre as velocidades definida em (6.9) no modelo cinemático do uniciclo (6.8), temos:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\phi) & 0 \\ \sin(\phi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r_e}{2} (\omega_r + \omega_l) \\ \frac{r_e}{d_R} (\omega_r - \omega_l) \end{bmatrix} \quad (6.10)$$

Reordenando a equação (6.10):

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r_e}{2} \cos(\phi) & \frac{r_e}{2} \cos(\phi) \\ \frac{r_e}{2} \sin(\phi) & \frac{r_e}{2} \sin(\phi) \\ \frac{r_e}{d_R} & -\frac{r_e}{d_R} \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (6.11)$$

Deste modo, o modelo cinemático do robô móvel com esteiras é definido em (6.11). A partir do modelo, é possível propor uma estratégia de controle cinemático para guiar robô sobre a escada.

## 6.2 Controle cinemático baseado em rastreamento de trajetória

A estratégia de controle cinemático estudada se baseia no rastreamento de trajetórias. Este método, proposto em Canudas de Wit et al. (1996), determina uma trajetória a qual o robô deve seguir e através da lei de controle proposta, os erros de orientação e posição em relação à trajetória desejada são minimizados para zero.

A figura (6.4) ilustra o rastreamento de trajetórias e as variáveis de interesse para a resolução do problema.

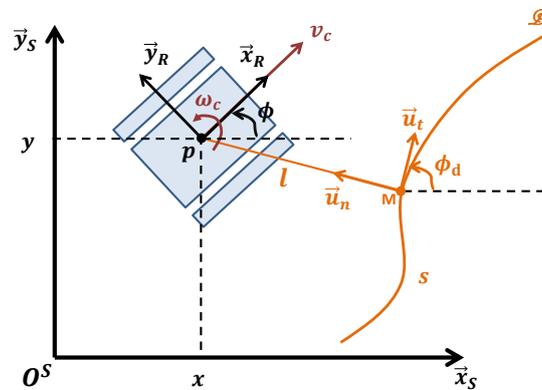


Figura 6.4: Exemplo do caso geral para o rastreamento de trajetórias

Temos que  $\mathcal{P}$  representa o caminho a ser seguido e o ponto  $M$  é a projeção ortogonal da posição  $p = (x, y)$  do robô no caminho  $\mathcal{P}$ . Os termos  $u_t$  e  $u_n$  representam os vetores tangente e normal ao caminho  $\mathcal{P}$  no ponto  $M$ , respectivamente. A variável  $l$  representa a distância com sinal entre  $M$  e  $p$ . Já o termo  $s$  é a distância com sinal ao longo da curva  $\mathcal{P}$  entre um ponto arbitrário na mesma e o ponto  $M$ . O ângulo  $\phi$  representa a orientação do robô em relação ao eixo  $x_S$  e o  $\phi_d$  é o ângulo entre o vetor tangente  $u_t$  e o eixo  $x_S$  do sistema de coordenadas fixo  $O_S$ . Por fim  $c(s)$  representa a curvatura do trajetória no ponto  $M$ , assumindo ser uniformemente limitada e diferenciável.

A configuração do robô móvel com esteiras em movimento planar é completamente determinada pelas coordenadas de posição  $p = [x \ y]$  e orientação  $\phi$ . Entretanto, no desenvolvimento do controle utilizam-se as variáveis de estado  $s, l, e_\phi$  para definir a configuração do robô, substituindo a escolha usual  $(x, y, \phi)$ . Ressalta-se que a variável  $e_\phi = \phi - \phi_d$ , define o erro de orientação entre o robô e o caminho  $\mathcal{P}$ .

Desta maneira, a dinâmica do novo sistema é dada por:

$$\begin{aligned}\dot{s} &= v_c \cos(e_\phi) \left( \frac{1}{1 - c(s)l} \right) \\ \dot{l} &= v_c \sin(e_\phi) \\ \dot{e}_\phi &= \omega_c - v_c \cos(e_\phi) \left( \frac{c(s)}{1 - c(s)l} \right)\end{aligned}\tag{6.12}$$

onde  $v_c$  e  $\omega_c$  representam as velocidades linear e angular do robô, respectivamente.

Desta maneira, a partir do controle proposto em (Canudas de Wit et al. 1996) é possível minimizar as variáveis  $l$  e  $e_\phi$  regidas pela dinâmica (6.12).

O método de controle é capaz ainda de rastrear trajetórias em um caso particular, as trajetórias retilíneas.

Como o problema proposto nesta dissertação considera os tipos de escadas retas, o modelo do sistema foi particularizado para o rastreamento de trajetórias retilíneas, aplicado ao problema de deslocamento em escadas.

### **Caso particular: Trajetórias retilíneas**

Para o problema proposto nesta dissertação, o rastreamento de trajetórias foi particularizado para o caso em que as trajetórias são retilíneas, visto que o principal objetivo do controle, durante a locomoção sobre a escada, é manter o robô em posição equidistante das laterais e com a orientação na direção  $y_S$  de subida.

Portanto, ao tratar as trajetórias retilíneas, a curvatura se torna zero ( $c(s) = 0$ ) independentemente da distância  $s$  ao longo da trajetória, fato que exclui a necessidade de obter a dinâmica  $\dot{s}$ .

A figura 6.5 ilustra o caso particular do rastreamento de trajetórias retilíneas aplicado à locomoção sobre escadas.

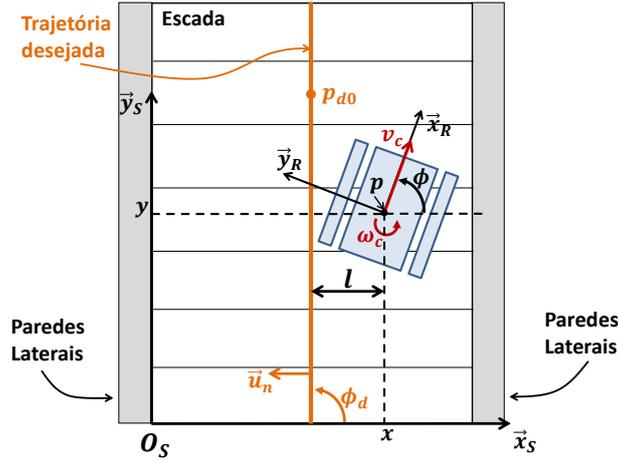


Figura 6.5: Vista superior de um robô do tipo EOD realizando rastreamento de trajetória retilíneas em escadas

Desta maneira, o novo conjunto de equações do sistema, excluindo-se a curvatura  $c(s)$  e a dinâmica  $\dot{s}$ , é definido por:

$$\dot{l} = v_c \sin(e_\phi) \quad (6.13)$$

$$\dot{e}_\phi = \omega_c$$

Para calcular a dinâmica definida em (6.13), é necessário obter as equações correspondentes às variáveis  $l$  e  $e_\phi$ , assim:

$$l = u_n^T (p - p_{d0}) \quad (6.14)$$

$$e_\phi = \phi - \phi_d \quad (6.15)$$

onde

- a)  $l$ : Distância ortogonal com sinal entre o robô e a reta que define a trajetória desejada;
- b)  $u_n$ : Vetor normal à trajetória retilínea desejada;
- c)  $p = [x \ y]$ : Posição do robô em relação ao sistema de coordenadas da escada  $O_S$ ;
- d)  $p_{d0} = [x_{d0} \ y_{d0}]$ : Representa um ponto qualquer pertencente à reta que define a trajetória desejada;
- e)  $\phi$ : Orientação do robô em relação ao eixo  $x_S$ ;
- f)  $\phi_d$ : Ângulo da trajetória desejada, medido em relação ao eixo  $x_S$ ;
- g)  $e_\phi$ : Erro de orientação entre o robô e a trajetória retilínea desejada.

Substituindo  $u_n = [\cos(\phi_d + \pi/2) \ \sin(\phi_d + \pi/2)]$  nas equações dos estados do sistema (6.14) e colocando  $p$  e  $p_{d0}$  em termos das suas coordenadas cartesianas,

temos:

$$\begin{aligned}
l &= \begin{bmatrix} \cos(\phi_d + \frac{\pi}{2}) & \sin(\phi_d + \frac{\pi}{2}) \end{bmatrix} \begin{bmatrix} x - x_{d0} \\ y - y_{d0} \end{bmatrix} \\
l &= \begin{bmatrix} -\sin(\phi_d) & \cos(\phi_d) \end{bmatrix} \begin{bmatrix} x - x_{d0} \\ y - y_{d0} \end{bmatrix} \\
l &= -\sin(\phi_d)(x - x_{d0}) + \cos(\phi_d)(y - y_{d0})
\end{aligned} \tag{6.16}$$

derivando o estado  $l$  (6.16) em relação ao tempo,

$$\dot{l} = -\sin(\phi_d)\dot{x} + \cos(\phi_d)\dot{y} \tag{6.17}$$

substituindo  $\dot{x}$  e  $\dot{y}$  pela equação do modelo cinemático do robô móvel (6.8) em termos das velocidades  $v_c$  e  $\omega_c$ ,

$$\begin{aligned}
\dot{l} &= -\sin(\phi_d)v_c \cos(\phi) + \cos(\phi_d)v_c \sin(\phi) \\
&= v_c (\sin(\phi) \cos(\phi_d) - \cos(\phi) \sin(\phi_d)) \\
&= v_c \sin(\phi - \phi_d) \\
\dot{l} &= v_c \sin(e_\phi)
\end{aligned} \tag{6.18}$$

Já a dinâmica do erro de orientação é obtida diretamente derivando-se  $e_\phi$ :

$$\begin{aligned}
\frac{d}{dt}(e_\phi) &= \frac{d}{dt}(\phi - \phi_d) \ , \quad \phi_d = cte \\
\dot{e}_\phi &= \dot{\phi} = \omega_c
\end{aligned} \tag{6.19}$$

Portanto, a partir da definição das variáveis  $l$  e  $e_\phi$ , obtêm-se a dinâmica do sistema (6.13) para o rastreamento de trajetórias retilíneas.

Embora o caso particular do rastreamento de trajetórias tenha sido exemplificado apenas para a locomoção sobre a escada, a mesma estratégia de controle foi aplicada também para a etapa de aproximação do robô do ponto inicial até a posição próxima à escada. Deste modo, a subseção 6.2.1 apresenta o controle aplicado à locomoção sobre escadas e a subseção 6.2.2 apresenta o mesmo controle, porém aplicado durante a etapa de aproximação até a mesma.

### 6.2.1 Controle para deslocamento em escadas durante a subida

O principal objetivo do controle é permitir que o robô siga uma trajetória retilínea equidistante das laterais da escada com a orientação definida pela direção  $y_S$ , de

modo que o robô deve ser guiado até a parte superior da escada.

Durante o deslocamento, o operador pode alterar a velocidade linear  $v_c(t)$  do robô, sendo, desta maneira, um sinal externo de entrada no sistema.

A figura 6.6 ilustra o diagrama de blocos da estratégia de controle em malha fechada aplicada para a locomoção em escadas.

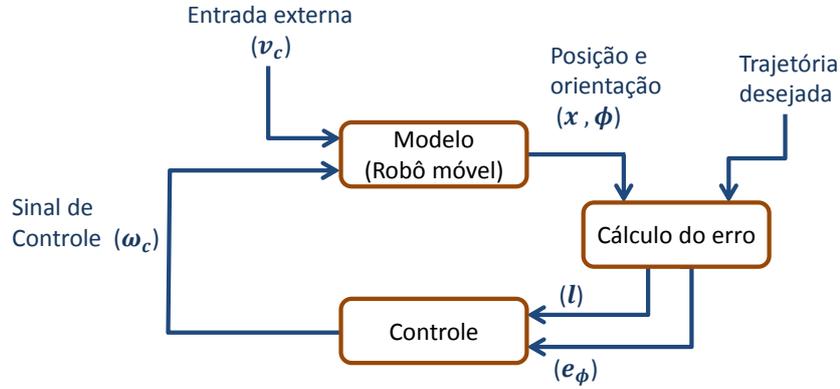


Figura 6.6: Diagrama de blocos da estratégia de controle aplicada durante a locomoção em escadas

Portanto, é necessário encontrar uma lei de controle

$$\omega_c = g(l, e_\phi, v_c(t)) \quad (6.20)$$

que permita ao robô seguir a trajetória determinada, de modo a minimizar as variáveis de estado  $l$  e  $e_\phi$ . Deste modo:

$$\lim_{t \rightarrow \infty} l(t) = 0 \quad ; \quad \lim_{t \rightarrow \infty} e_\phi(t) = 0 \quad (6.21)$$

Considerando a dinâmica do sistema (6.13):

$$\begin{aligned} \dot{l} &= v_c \sin(e_\phi) \\ \dot{e}_\phi &= \omega_c \end{aligned}$$

A lei de controle proposta é definida como segue:

$$\omega_c = -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \quad (6.22)$$

onde  $k$  e  $k_2$  são constantes maiores que zero.

Substituindo a lei de controle (6.22) em (6.13), o sistema em malha fechada se

torna:

$$\begin{aligned} \dot{l} &= v_c \sin(e_\phi) \\ \dot{e}_\phi &= -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \end{aligned} \quad (6.23)$$

Para provar a estabilidade assintótica do ponto de equilíbrio do sistema em malha fechada (6.23), foi considerada a seguinte hipótese:

**Hipótese 6.1** *A velocidade linear  $v_c(t)$  é uma função contínua limitada com derivadas primeira e segunda limitadas, e  $v_c(t) > 0$ , tal que:*

$$\lim_{t \rightarrow \infty} v_c(t) \neq 0$$

**Lema 6.1** *Seja  $D = \{(l, e_\phi) \in \mathbb{R}^2 \mid \|(l, e_\phi)\| < r\}$  e seja  $(l = 0, e_\phi = 0)$  um ponto de equilíbrio do sistema em malha fechada (6.23). Seja  $V(l, e_\phi) : [0, \infty) \times D \rightarrow \mathbb{R}$  uma função continuamente diferenciável tal que*

$$2V(l, e_\phi) = k_2 l^2 + e_\phi^2 \quad (6.24)$$

*Supondo satisfeita a hipótese (6.1), então a origem  $(l = 0, e_\phi = 0)$  do sistema em malha fechada (6.23) é globalmente assintoticamente estável.*

**Prova 6.1** *O primeiro passo é provar a convergência para zero da derivada da função candidata de Lyapunov:*

$$\dot{V}(l, e_\phi) \rightarrow 0 \quad (6.25)$$

*Definindo a função  $f(l, e_\phi)$  a partir do sistema em malha fechada (6.23), temos*

$$f(l, e_\phi) = \begin{bmatrix} v_c \sin(e_\phi) \\ -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \end{bmatrix} \quad (6.26)$$

*Utilizando o Teorema 4.4 apresentado em (Khalil 1996) página 192, temos que a função  $f(l, e_\phi)$  deve ser localmente Lipschitz em  $(l, e_\phi)$  e contínua por partes em  $t$ . A função candidata de Lyapunov  $V(l, e_\phi)$  deve ainda ser tal que:*

$$W_1(l, e_\phi) \leq V(l, e_\phi) \leq W_2(l, e_\phi) \quad (6.27)$$

$$\dot{V}(l, e_\phi) \leq -W(l, e_\phi) \quad (6.28)$$

*sendo  $W_1(l, e_\phi)$  e  $W_2(l, e_\phi)$  funções contínuas positivas definidas e  $W(l, e_\phi)$  positiva semidefinida em  $D$ .*

Para simplificar a notação as funções  $W(l, e_\phi)$ ,  $W_1(l, e_\phi)$ ,  $W_2(l, e_\phi)$  e  $V(l, e_\phi)$  serão designadas apenas por  $W$ ,  $W_1$ ,  $W_2$  e  $V$ .

Derivando a função candidata de Lyapunov (6.24) em relação às trajetórias do sistema (6.23),

$$2\dot{V} = 2k_2 l \dot{l} + 2e_\phi \dot{e}_\phi$$

Substituindo na equação anterior os termos  $\dot{l}$  e  $\dot{e}_\phi$  por (6.23), temos:

$$\begin{aligned}\dot{V} &= k_2 v_c l \sin(e_\phi) + e_\phi \left( -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \right) \\ \dot{V} &= -k e_\phi^2 \leq 0\end{aligned}\tag{6.29}$$

Com  $V > 0$  e  $\dot{V} \leq 0$ , conclui-se que  $V$  é limitada inferiormente por zero e monotonicamente decrescente, convergindo para um limite quando  $t \rightarrow \infty$ .

Com  $V$  limitada, conseqüentemente  $e_\phi$  e  $l$  são limitados. Ao observar o sistema em malha fechada (6.23), sendo  $k_2$  e  $k$  constantes,  $v_c$  limitada (hipótese 6.1), conclui-se que  $\dot{e}_\phi$  e  $\dot{l}$  são limitados.

A função  $f(l, e_\phi)$  definida em (6.26) não apresenta pontos de singularidades em  $D$  e a função  $v_c$  é contínua (hipótese 6.1), logo a condição de continuidade por partes de  $f(l, e_\phi)$  é satisfeita.

A condição de Lipschitz é utilizada no Teorema 4.4 para garantir que  $W$  é uniformemente contínua. A partir de (Slotine & Li 1991) temos que para provar que uma função é uniformemente contínua, basta garantir que a derivada da função é limitada. Assim, a condição de Lipschitz será substituída pela hipótese da derivada  $\dot{W}$  ser limitada.

Definindo ( $W = k e_\phi^2$ ), a hipótese ( $\dot{V} \leq -W$ ) é satisfeita e derivando  $W$  em relação ao tempo, temos:

$$\begin{aligned}\dot{W} &= -2k e_\phi \dot{e}_\phi \\ &= -2k e_\phi \left( -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \right) \\ &= 2k k_2 v_c l \sin(e_\phi) + 2k^2 e_\phi^2\end{aligned}\tag{6.30}$$

com  $l$ ,  $e_\phi$ ,  $k$ ,  $k_2$  e  $v_c$  limitados, conclui-se que  $\dot{W}$  é limitada, logo a função  $W$  é uniformemente contínua.

Escolhendo  $W_1$  e  $W_2$  como,

$$W_1 = \frac{1}{4}(k_2 l^2 + e_\phi^2) \quad (6.31)$$

$$W_2 = 2(k_2 l^2 + e_\phi^2) \quad (6.32)$$

a condição  $W_1 \leq V \leq W_2$  é satisfeita e com a hipótese de  $W$  uniformemente contínua, todas as hipóteses exigidas no Teorema 4.4 são satisfeitas, conseqüentemente,  $W \rightarrow 0$  quando  $t \rightarrow \infty$ . Como  $W_1$  é radialmente ilimitada, visto que pela equação (6.31)  $W_1 \rightarrow \infty$  se  $\|(l, e_\phi)\| \rightarrow \infty$ , e as hipóteses são satisfeitas globalmente, conclui-se que a convergência de  $W \rightarrow 0$  é válida para toda condição inicial  $(l_0, e_{\phi 0}) \in \mathbb{R}^2$ .

Com  $W \rightarrow 0$  temos que  $\dot{V} \rightarrow 0$ , portanto a partir de (6.29) temos que  $e_\phi \rightarrow 0$ , pois  $k$  é constante.

Portanto, após concluir a convergência do erro de orientação  $e_\phi$  para zero agora temos que determinar se distância  $l$  tende a zero.

Para determinar a convergência de  $l$ , utilizaremos a convergência de  $e_\phi$  e a função limitada  $\dot{v}_c$  (hipótese 6.1), de modo que a partir da convergência de  $d(v_c^2 e_\phi)/dt$  para zero, conseguiremos chegar à convergência de  $l$ .

A convergência de  $d(v_c^2 e_\phi)/dt$  para zero é realizada através do lema de Barbalat B.1(Khalil 1996).

Derivando o termo  $v_c^2 e_\phi$  em relação ao tempo, temos:

$$\frac{d}{dt}(v_c^2 e_\phi) = 2 v_c \dot{v}_c e_\phi - v_c^2 k e_\phi - k_2 v_c^3 l \frac{\sin(e_\phi)}{e_\phi} \quad (6.33)$$

Para provar que  $d(v_c^2 e_\phi)/dt$  é uniformemente contínuo derivamos uma segunda vez a expressão  $v_c^2 e_\phi$ ,

$$\frac{d^2}{dt^2}(v_c^2 e_\phi) = \underbrace{2 \dot{v}_c^2 e_\phi + 2 v_c (\ddot{v}_c e_\phi + \dot{v}_c \dot{e}_\phi) - 2 v_c \dot{v}_c k e_\phi - v_c^2 k \dot{e}_\phi}_{f_1} - \underbrace{k_2 \frac{d}{dt}(v_c^3 l \frac{\sin(e_\phi)}{e_\phi})}_{f_2} \quad (6.34)$$

com as variáveis  $v_c$ ,  $\dot{v}_c$ ,  $\ddot{v}_c$ ,  $e_\phi$ ,  $l$ ,  $k$  e  $k_2$  limitadas, temos que a função  $f_1$  é limitada, já a prova de  $f_2$  limitada é apresentada no apêndice B.2. Conseqüentemente, com  $d^2(v_c^2 e_\phi)/dt^2$  limitada temos que  $d(v_c^2 e_\phi)/dt$  é uniformemente contínua e como  $v_c^2 e_\phi$  é limitado, conclui-se que

$$\frac{d}{dt}(v_c^2 e_\phi) \rightarrow 0 \quad (6.35)$$

Retomando (6.33) temos que os dois primeiros termos em função de  $e_\phi$  tendem a

zero, restando apenas a última parcela dependente de  $l$ . Como a expressão  $d(v_c^2 e_\phi)/dt$  tende a zero, então:

$$-k_2 v_c^3 l \frac{\sin(e_\phi)}{e_\phi} \rightarrow 0 \quad (6.36)$$

Consequentemente, como  $v_c \neq 0$  e  $(\sin(e_\phi)/e_\phi) \rightarrow 1$  quando  $t \rightarrow \infty$ , então necessariamente  $l \rightarrow 0$ .

Desta maneira, para qualquer condição inicial  $(e_{\phi_0}, l_0) \in \mathbb{R}^2$  o controle (6.22) torna a origem ( $l = 0, e_\phi = 0$ ) do sistema globalmente assintoticamente estável.

## 6.2.2 Controle para a aproximação até a escada

Antes de realizar o deslocamento propriamente dito sobre a escada, é necessário definir o controle que deve guiar o robô do ponto inicial distante da escada até a posição próxima à mesma. Portanto, após identificar a posição da escada e detectar seus parâmetros, propõe-se que seja utilizada a mesma lei de controle (6.22) apresentada na subseção 6.2.1 para realizar a aproximação.

Assim, é definida uma trajetória retilínea que conecta a posição inicial do robô ao ponto próximo à escada, de modo que o controle permite ao robô rastrear a trajetória desejada, minimizando a distância  $l$  do robô até a mesma, além do erro de orientação  $e_\phi$ .

A figura 6.7 ilustra a trajetória desejada e os sistemas de coordenadas adotados.

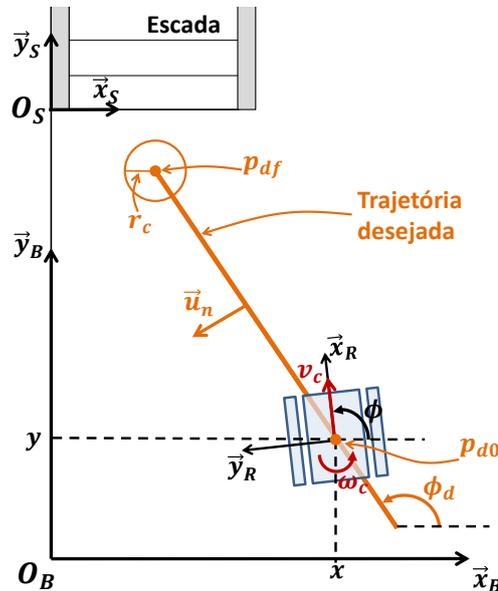


Figura 6.7: Exemplo do robô realizando o rastreamento de trajetória do ponto inicial  $p_{do}$  até o ponto final  $p_{df}$  próximo à escada

Desta maneira, a nova definição das variáveis e parâmetros é realizada como segue:

- a)  $u_n$ : Vetor ortogonal à trajetória retilínea desejada;
- b)  $p_{d0} = [x_{d0} \ y_{d0}]$ : Posição inicial do robô pertencente à reta que define a trajetória desejada, sendo medida em relação ao sistema de coordenadas da base  $O_B$ ;
- c)  $p_{df} = [x_{df} \ y_{df}]$ : Posição final desejada próxima à escada medida em relação à  $O_B$ , também pertencente à trajetória desejada;
- d)  $\phi$ : Orientação do robô em relação ao eixo  $x_B$ ;
- e)  $\phi_d$ : Ângulo da trajetória desejada, medido em relação ao eixo  $x_B$ ;
- f)  $r_c$ : Raio da circunferência ao redor do ponto  $p_{df}$  que delimita a área onde se considera o alcance da posição final desejada;
- g)  $l$ : Distância com sinal do robô à trajetória desejada, medida ortogonalmente;
- h)  $e_\phi$ : Erro de orientação do robô em relação à trajetória desejada.

Como pode ser observado na figura 6.7, para o controle durante a aproximação até a escada, os ângulos e posições do robô e trajetória desejada são representados em relação ao sistema de coordenadas da base  $O_B$ , diferindo portanto do controle para locomoção sobre a escada. Ressalta-se ainda que apesar da distância  $l$  do robô à reta ser zero inicialmente, os desvios do robô em relação à trajetória provocam alterações da referida variável de estado que é minimizada pela ação do controle (6.22).

O raio  $r_c$ , que delimita a região da posição final a ser alcançada pelo robô, foi definido em virtude das incertezas do método de localização e incertezas paramétricas do robô. As incertezas mencionadas podem provocar pequenos desvios da posição final do robô em relação a trajetória desejada, o que poderia originar o não alcance do alvo final. Portanto, no momento em que o robô ingressa na região de aceitação para a posição final desejada, é atribuído valor igual a zero para a velocidade linear  $v_c$ , interrompendo o deslocamento do robô.

Desta maneira, ao definir a trajetória desejada e aplicar o controle (6.22) descrito anteriormente, o robô é conduzido até a posição final, minimizando as variáveis de estado  $e_\phi$  e  $l$ .

### Controle de Orientação

Ao alcançar a região definida por  $r_c$ , o robô provavelmente estará com a sua orientação defasada em relação à escada, por este motivo aplica-se um controle para alinhar ambas as orientações, robô e escada. Desta maneira, define-se uma nova trajetória desejada com ângulo  $\phi_d$  equivalente ao ângulo de orientação da escada e aplica-se a mesma lei de controle (6.22) já apresentada, com a particularidade de atribuir valor igual a zero para a velocidade linear ( $v_c = 0$ ).

Retomando as equações do estado do sistema (6.23) para o rastreamento de

trajetórias, temos:

$$\begin{aligned} \dot{l} &= v_c \sin(e_\phi) \\ \dot{e}_\phi &= \omega_c \\ \omega_c &= -k_2 v_c l \frac{\sin(e_\phi)}{e_\phi} - k e_\phi \end{aligned}$$

Tornando ( $v_c = 0$ ), obtêm-se:

$$\dot{l} = 0 \quad (6.37)$$

$$\dot{e}_\phi = \omega_c \quad (6.38)$$

$$\omega_c = -k e_\phi \quad (6.39)$$

Substituindo o controle  $\omega_c$  na dinâmica do erro de orientação (6.38), temos

$$\dot{e}_\phi = -k e_\phi \quad (6.40)$$

Resolvendo a equação diferencial (6.40), temos

$$e_\phi(t) = e_{\phi 0} e^{-kt} \quad (6.41)$$

onde  $e_{\phi 0}$  é o erro de orientação inicial entre a escada e o robô.

Portanto, mantendo a constante  $k > 0$ , o  $\lim_{t \rightarrow \infty} e_\phi(t) \rightarrow 0$  para qualquer condição inicial  $e_{\phi 0}$ . Provando, assim, que o controle de orientação (6.39), torna a origem  $e_\phi = 0$  do sistema (6.40) globalmente assintoticamente estável.

### 6.3 Simulações

Para validar a estratégia de controle cinemático proposta, foram realizadas simulações utilizando o modelo do robô móvel com esteiras e direção diferencial. O modelo e o controle foram implementados no software Matlab de modo que os resultados obtidos são apresentados a seguir.

Para avaliar a robustez do método, foi adicionado ao modelo fatores de escorregamento  $i_l$  e  $i_r$  para as esteiras esquerda e direita. Deste modo, o modelo cinemático do robô definido pela equação (6.11) se transforma em:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r_e}{2} \cos \phi & \frac{r_e}{2} \cos \phi \\ \frac{r_e}{2} \sin \phi & \frac{r_e}{2} \sin \phi \\ \frac{r_e}{d_R} & -\frac{r_e}{d_R} \end{bmatrix} \begin{bmatrix} i_r \omega_r \\ i_l \omega_l \end{bmatrix}, \quad 0 < (i_r, i_l) \leq 1 \quad (6.42)$$

Como pode ser observado, para os parâmetros  $i_r = i_l = 1$  não há escorregamento, visto que a velocidade angular das rodas e o valor de entrada são equivalentes. Entretanto, quando os parâmetros são alterados para  $i_r \leq 1$  e  $i_l \leq 1$ , o valor real de velocidade angular se torna inferior ao valor de entrada, introduzindo variações de orientação do robô e deslocamentos lineares inesperados.

As simulações foram divididas em duas etapas: a primeira para o controle durante o deslocamento sobre a escada e a segunda para a aproximação até a mesma.

### 6.3.1 Simulação: Controle para o deslocamento em escadas durante a subida

Para a primeira simulação realizada, foi definida uma trajetória retilínea com ângulo  $\phi_d = 90^\circ$  em relação ao eixo  $x_S$  horizontal e passando pelo ponto  $p_{d0} = [0.5 \ 0]$  m. O raio da engrenagem principal que traciona as esteiras e a distância entre as mesmas foram definidos como  $r_e = 0.1$  m e  $d_R = 0.6$  m, respectivamente. A configuração inicial do robô é definida por  $\mathbf{q}_0 = [1 \text{ m} \ 0 \text{ m} \ 70^\circ]$  e para cada fator de escorregamento  $(i_r, i_l)$  foi utilizada uma distribuição aleatória com dois valores possíveis  $[0.7 \ 0.95]$ . Os parâmetros de controle foram definidos como  $k = 0.38$  e  $k_2 = 3.34$ .

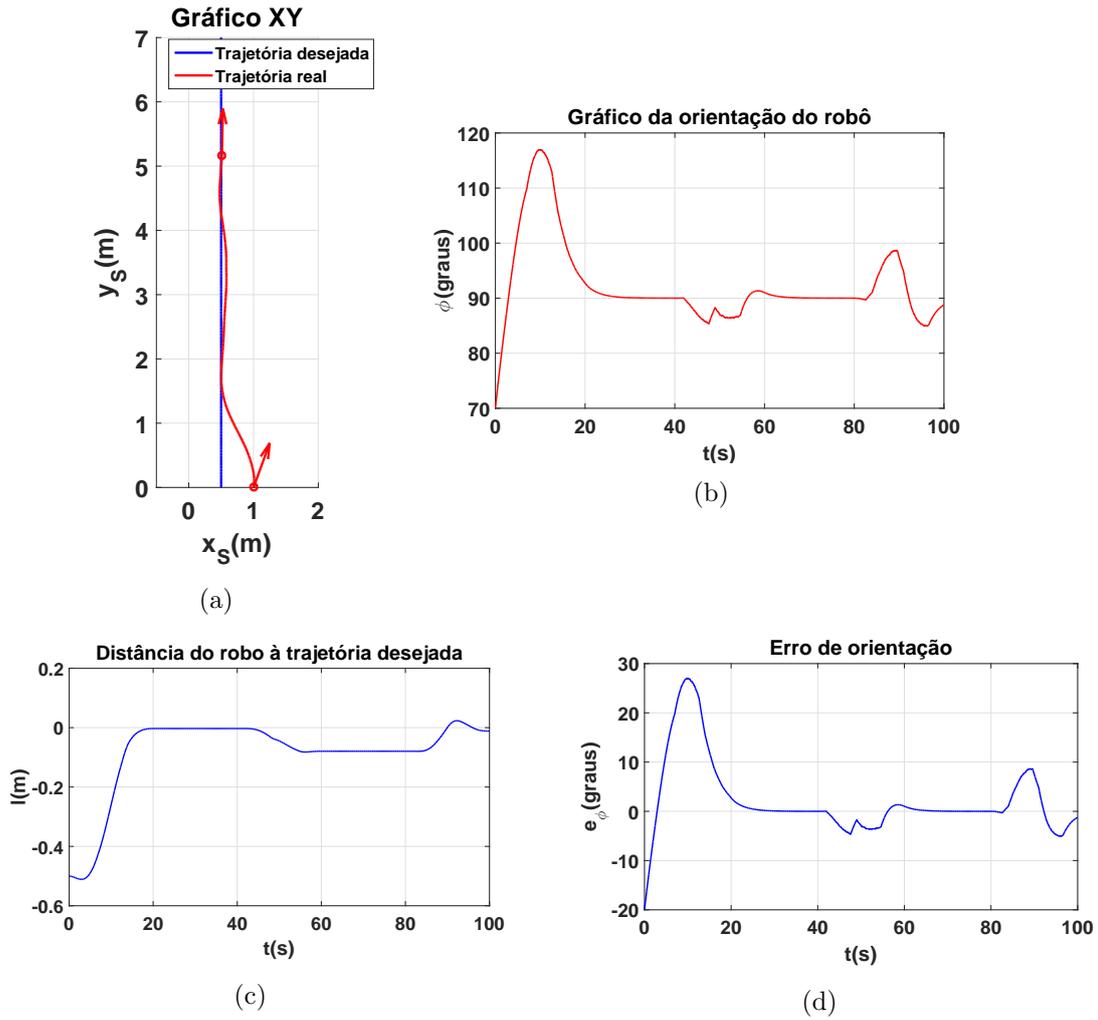


Figura 6.8: Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, utilizando os parâmetros  $\mathbf{q}_0 = [1 \ 0 \ 70^\circ]$ ,  $\phi_d = 90^\circ$ ,  $p_{d0} = [0.5 \ 0]$  m,  $(i_r, i_l) \in [0.7 \ 0.95]$ ,  $k = 0.38$  e  $k_2 = 3.34$ . Os gráficos (c) e (d) apresentam o erro de orientação  $e_\phi$  e a distância  $l$  entre robô e trajetória desejada para os mesmos parâmetros de simulação.

Como pode ser observado nos gráficos em 6.8, o controle foi capaz de rastrear a trajetória desejada apresentando um erro final de orientação  $e_\phi = -1.23^\circ$  e distância até a linha  $l = -0.011$  m, o que pode ser considerado um bom resultado tendo em vista a influência do fator de escorregamento considerado.

**Observação 6.1** *Os transientes observados no gráfico do erro de orientação do robô (figura 6.8 (d)) entre os instantes (40s a 60s) e (80s a 100s) ocorreram em virtude dos fatores de escorregamento, que nos intervalos mencionados, reduziram a velocidade das esteiras, ocasionando as variações de orientação observadas.*

A velocidade linear do robô  $v_c$  foi definida como uma senoide com máximo 0.15 m/s e ciclo negativo multiplicado por zero (vide gráfico 6.9(a)). O objetivo da função  $v_c$  utilizada é simular a ação do operador que pode variar a velocidade e até mesmo parar o robô em determinados momentos, durante o deslocamento sobre a escada.

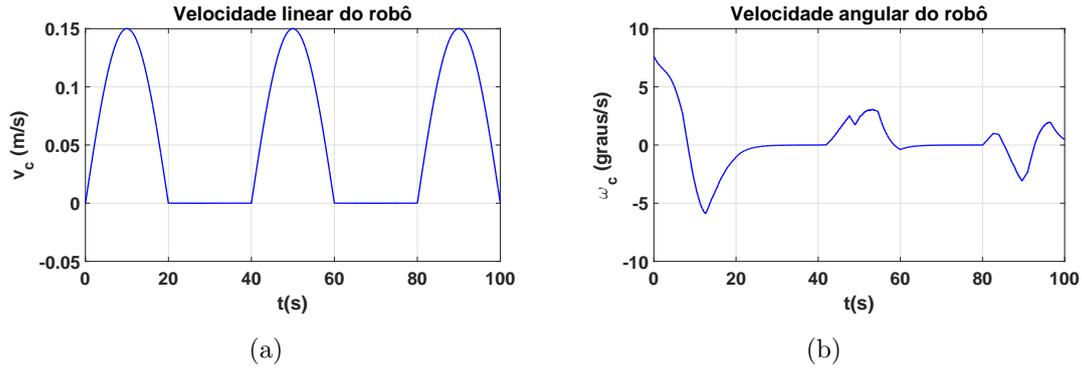


Figura 6.9: Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta a variação da velocidade linear  $v_c$  pelo operador e o gráfico (b) apresenta o sinal de controle  $\omega_c$ , obtidos utilizando os parâmetros  $\mathbf{q}_0 = [1 \ 0 \ 70^\circ]$ ,  $\phi_d = 90^\circ$ ,  $p_{d0} = [0.5 \ 0]$  m,  $(i_r, i_l) \in [0.7 \ 0.95]$ ,  $k = 0.38$  e  $k_2 = 3.34$ .

Nos instantes em que é atribuído valor zero para a velocidade linear ( $v_c = 0$ ), observa-se que o controle faz com que a orientação do robô se iguale à orientação da trajetória desejada ( $\phi_d = 90^\circ$ ). Característica esta que pode ser utilizada pelo operador para reduzir o escorregamento entre esteiras e degraus, caso a orientação real do robô se afaste da orientação desejada.

Para a segunda simulação foram inseridas incertezas paramétricas no modelo, definindo o raio da engrenagem que traciona as esteiras como  $r_e = 0.3 + 0.005$  m e a distância entre as esteiras como  $d_R = 0.6 + 0.02$  m. Utilizou-se a mesma trajetória retilínea da primeira simulação e a configuração inicial do robô foi definida por  $\mathbf{q}_0 = [-0.4 \text{ m} \ 0 \text{ m} \ 90^\circ]$ . Os fatores de escorregamento e os parâmetros de controle foram escolhidos exatamente como na primeira simulação.

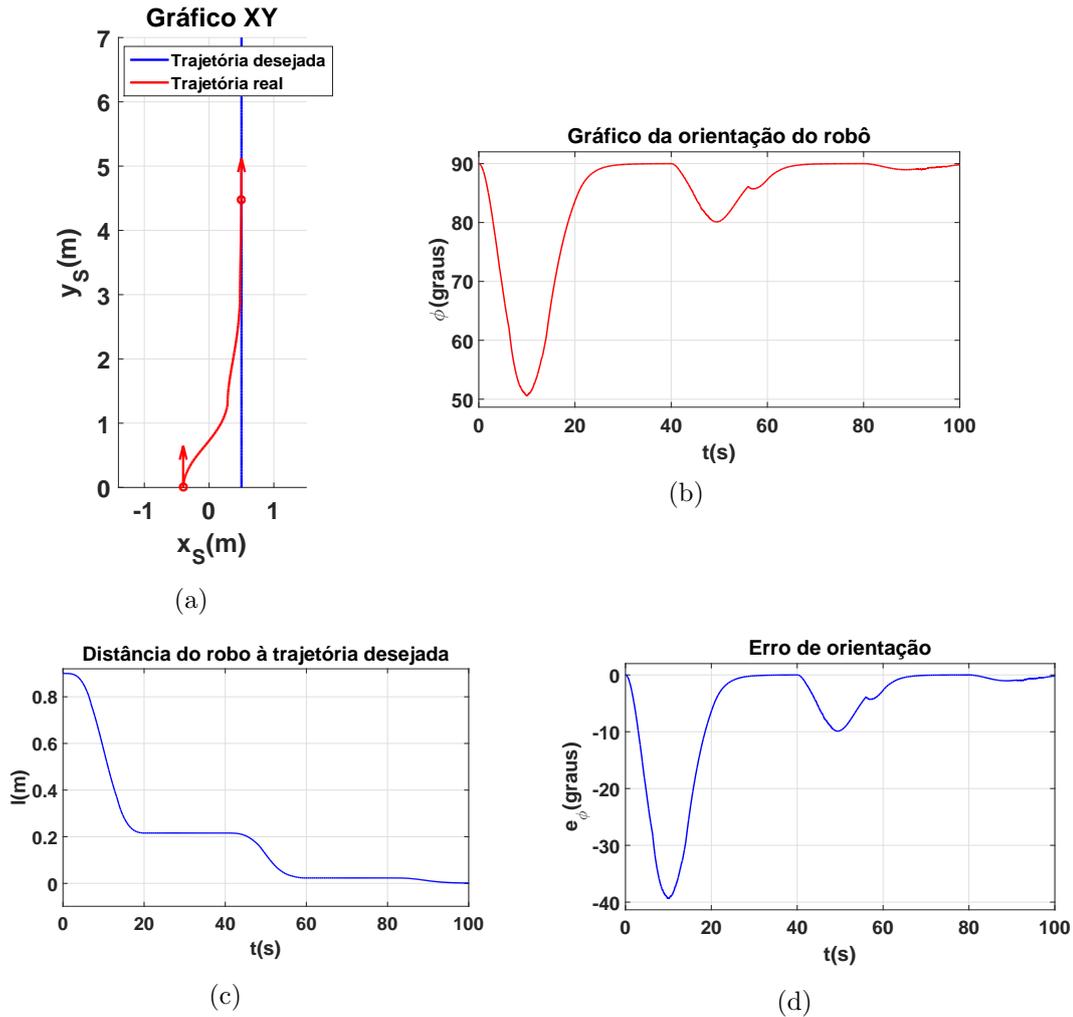


Figura 6.10: Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, utilizando os parâmetros  $\mathbf{q}_0 = [-0.4 \ 0 \ 90^\circ]$ , incertezas paramétricas  $r_e = 0.1 + 0.005$  m e  $d_R = 0.6 + 0.02$  m,  $(i_r, i_l) \in [0.7 \ 0.95]$ ,  $k = 0.38$  e  $k_2 = 3.34$ . Os gráficos (c) e (d) apresentam o erro de orientação  $e_\phi$  e a distância  $l$  entre robô e trajetória desejada para os mesmos parâmetros de simulação.

Os gráficos 6.10 demonstram que mesmo inserindo os erros paramétricos no modelo cinemático do robô, além da ação dos escorregamentos, o controle foi capaz de rastrear a trajetória desejada. O erro final de orientação foi  $e_\phi = -0.2^\circ$  e a distância do robô a trajetória foi  $l = 0.002$  m, demonstrando novamente a eficácia da estratégia de controle apresentada.

Assim como na primeira simulação, a velocidade linear  $v_c$  foi variada para avaliar o desempenho do controle e simular a ação do operador. Os gráficos da figura (6.11) apresentam os resultados obtidos.

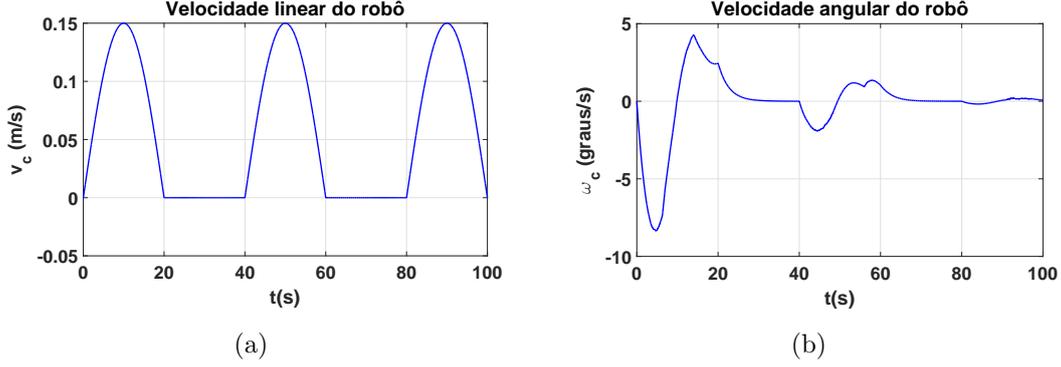


Figura 6.11: Simulação do controle: locomoção sobre a escada. O gráfico (a) apresenta a variação da velocidade linear  $v_c$  pelo operador e o gráfico (b) apresenta o sinal de controle  $\omega_c$ , obtidos utilizando os parâmetros  $q = [-0.4 \ 0 \ 90^\circ]$ ,  $r_e = 0.1 + 0.005$ ,  $d_R = 0.6 + 0.02$ ,  $i_r, i_l \in [0.7 \ 0.95]$ ,  $k = 0.38$  e  $k_2 = 3.34$ .

### 6.3.2 Simulação: Controle para a aproximação até a escada

A estratégia de controle proposta para a aproximação até a escada foi avaliada através de simulações, utilizando o software Matlab.

Para avaliar a robustez do controle, foi considerado que a posição inicial do robô difere do ponto onde a trajetória desejada se inicia. Ressalta-se ainda que foram adicionados os fatores de escorregamento  $i_l$  e  $i_r$  tal qual realizado anteriormente.

Os parâmetros de simulação foram definidos como segue: as posições inicial e final da trajetória desejada,  $p_{d0} = [2 \ 1]$  m e  $p_{df} = [1.1 \ 5.3]$  m, respectivamente, ambas medidas em relação ao sistema de coordenadas da base  $O_B$ ; o raio  $r_c = 0.15$  m da região onde se considera que a posição final foi alcançada; a configuração inicial do robô  $\mathbf{q}_0 = [3 \text{ m} \ 1 \text{ m} \ 60^\circ]$ ; os escorregamentos  $(i_l, i_r)$  definidos por distribuição aleatória com apenas dois valores  $[0.8 \ 1]$ ; a orientação da escada  $\phi_S = 90^\circ$  medida em relação ao vetor  $x_B$ ; a velocidade linear  $v_c = 0.08$  m/s; e por fim as constantes de controle  $k = 0.20$  e  $k_2 = 3.24$ .

Os gráficos em 6.12 apresentam os resultados da simulação.

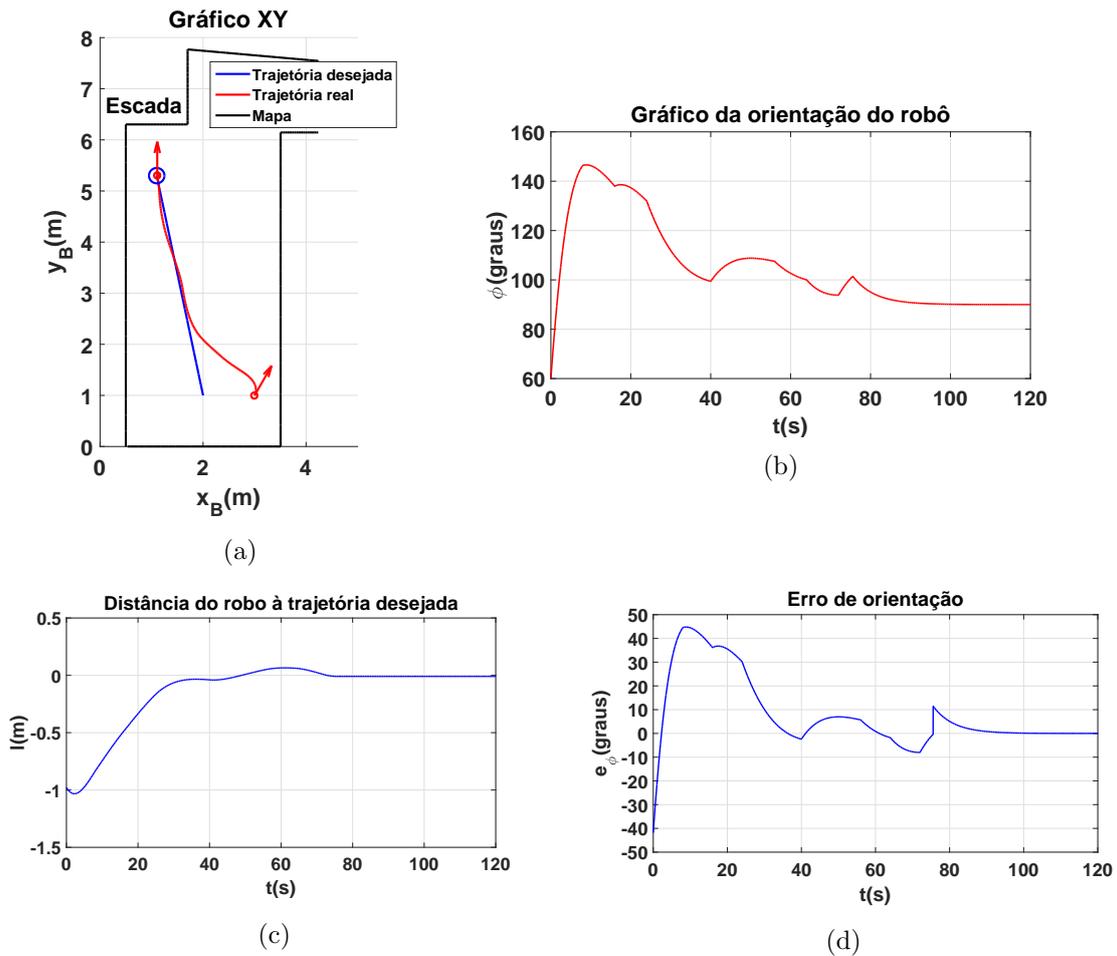


Figura 6.12: Simulação do controle: aproximação até a escada. O gráfico (a) apresenta as trajetórias real/desejada no plano xy e (b) a orientação do robô, ambos obtidos utilizando os parâmetros  $\mathbf{q}_0 = [3 \ 1 \ 60^\circ]$ ,  $p_{d0} = [2 \ 1]$ ,  $p_{df} = [1.1 \ 5.3]$ ,  $(i_r, i_l) \in [0.8 \ 1]$ ,  $k = 0.20$  e  $k_2 = 3.24$ . Os gráficos (c) e (d) apresentam o erro de orientação  $e_\phi$  e a distância  $l$  entre robô e trajetória desejada para os mesmos parâmetros de simulação.

Nos gráficos da figura 6.12 observa-se que o robô, mesmo sob efeito de escorregamentos, segue a trajetória desejada em todo o percurso, alcançando a região final desejada. Após a mudança para o controle de orientação, o robô foi capaz de alinhar sua orientação com a orientação da escada. O erro entre a posição final desejada e a real foi de 0.01 m em módulo e o erro de orientação final foi  $e_\phi = 0^\circ$ , validando assim o controle avaliado.

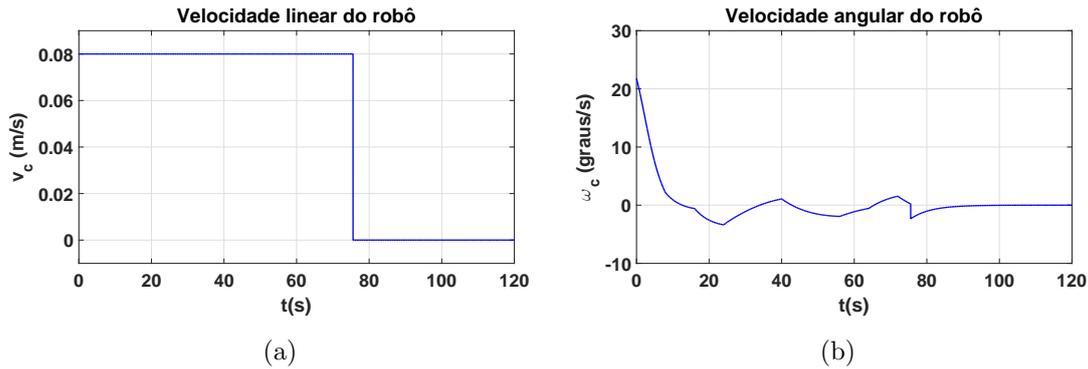


Figura 6.13: Simulação do controle: aproximação até a escada. O gráfico (a) apresenta a variação da velocidade linear  $v_c$  e o gráfico (b) apresenta o sinal de controle  $\omega_c$ , obtidos utilizando os parâmetros  $\mathbf{q}_0 = [3 \ 1 \ 60^\circ]$ ,  $p_{d0} = [2 \ 1]$ ,  $p_{df} = [1.1 \ 5.3]$ ,  $i_r, i_l \in [0.8 \ 1]$ ,  $k = 0.20$  e  $k_2 = 3.24$ .

Os gráficos 6.13(a) e (b) apresentam as velocidades linear e angular do robô, respectivamente, de modo que é possível observar exatamente o instante em que o controle em rastreamento de trajetória é chaveado para o controle de orientação.

## 6.4 Conclusões

Portanto, neste capítulo foi apresentado o modelo cinemático do robô com esteiras do tipo EOD baseado no robô móvel com direção diferencial. Em seguida, foi descrita a estratégia de controle para o rastreamento de trajetórias sendo dividida em controle para locomoção sobre a escada durante a subida e controle para a aproximação até a escada.

A estabilidade do controle cinemático apresentado foi avaliada através da teoria de Lyapunov, de modo que se concluiu a estabilidade assintótica global do sistema em malha fechada.

Para validar ambas as estratégias de controle, foram realizadas simulações e apresentados os resultados. Tanto para o controle durante a aproximação até a escada, quanto para o controle durante a locomoção sobre a mesma, as simulações demonstraram que o controle utilizado é robusto a erros paramétricos no modelo do sistema, bem como inserção de fatores de escorregamento.

# Capítulo 7

## Locomoção semiautônoma em escadas durante a descida

Durante a exploração de novos ambientes em aplicações de busca e resgate ou aplicações militares, os robôs móveis com esteiras devem ser capazes de lidar com um outro desafio, a descida da escada.

Semelhante ao problema de locomoção em escadas durante a subida, o problema de descida da escada também apresenta riscos como: tombamentos durante a transição do piso superior para a escada, escorregamentos e colisões com paredes ou balaustres laterais.

Portanto, neste capítulo é formulado o problema de descida da escada utilizando um robô móvel com esteiras, sendo proposta ainda uma solução para o problema baseada nas ferramentas já apresentadas para a locomoção em escadas durante a subida.

### 7.1 Formulação do problema

O problema de descida da escada, semelhante à subida, consiste basicamente em detectar e localizar a escada utilizando os sensores do robô, em seguida deve ser aplicada uma metodologia de reconfiguração das pernas para realizar a transição do piso superior para a escada, o deslocamento propriamente dito na escada e a transição da escada para o piso inferior. Durante o deslocamento na escada, o robô deve apresentar ainda uma estratégia de controle que o conduza sobre os degraus até o piso inferior.

As mesmas considerações relacionadas ao robô, escada, ambiente e a interação entre estes elementos, apresentadas no capítulo 2, foram utilizadas nesta etapa.

Resumidamente, foi considerado um robô móvel com esteiras e pernas articuladas, apresentando ainda um manipulador e sensores como *lasers scan*, *Kinect* e

sensor inercial (IMU). Uma característica importante é que os sensores *laser* e o *Kinect* podem ser acoplados ao manipulador para auxiliar a percepção do ambiente.

Em relação a escada, foram mantidas as considerações relacionadas ao ângulo de inclinação entre 18° e 40° graus, bem como aquelas relacionadas ao formato das escadas retas e à limitação lateral por uma ou duas paredes. A interação entre as esteiras do robô e os degraus foi definida como o contato em um plano inclinado, sendo desconsiderados novamente os efeitos de escorregamento, tal qual foi realizado no problema de locomoção durante a subida.

O ambiente foi considerado novamente como o interior de prédios com estruturas planas e ortogonais (pisos, paredes e portas), de modo que existe um mapa conhecido a priori.

Por fim, foi considerado que o robô está inicialmente localizado próximo à escada, sendo esta hipótese necessária para permitir que o robô possa detectar e modelar a escada, além de estimar seus parâmetros.

## 7.2 Solução proposta

O problema de descida da escada pode ser considerado como a inversão do problema de subida apresentado nos capítulos anteriores. Desta maneira, foi proposta uma solução baseada nas ferramentas já apresentadas e avaliadas.

A locomoção sobre a escada para a descida foi dividida em quatro etapas: detecção e modelagem da escada, metodologia de reconfiguração das pernas, estratégia de controle e localização no ambiente.

### 7.2.1 Detecção e modelagem da escada

Para detectar e modelar a escada é proposto utilizar os dois métodos avaliados no capítulo 3: a detecção e modelagem utilizando sensores *lasers* e a detecção e modelagem da escada utilizando imagem 3D densa. Desta maneira, os dois métodos foram adaptados para atender às características do problema de descida da escada.

#### Detecção e localização da escada utilizando sensores *laser*

Para adaptar o método ao problema de descida da escada, novamente foram considerados dois *lasers*, um posicionado verticalmente e outro horizontalmente. Entretanto, o *laser* vertical foi posicionado na extremidade do manipulador, permitindo uma melhor percepção do ambiente e escada. A figura 7.1 ilustra os sistemas de coordenadas do *laser* vertical  $O_{LV}$ , escada  $O_S$  e robô  $O_R$  adotados.

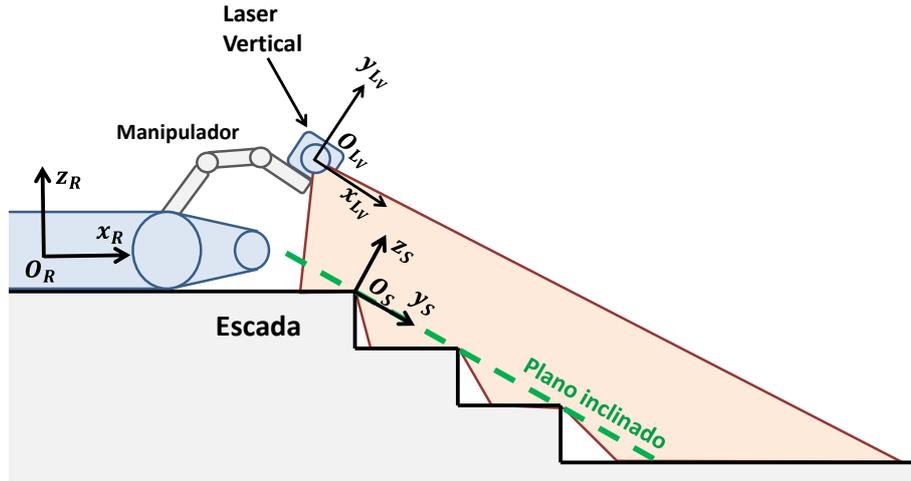


Figura 7.1: Exemplo da varredura do *laser* vertical aplicado ao problema de descida da escada.

Ao realizar a varredura do *laser* capturando os pontos 2D da escada e transformar os referidos pontos do sistema de coordenadas  $O_{LV}$  para o sistema de coordenadas do robô  $O_R$ , é possível observar que os pontos referentes aos degraus estarão dispostos como na figura 7.2.

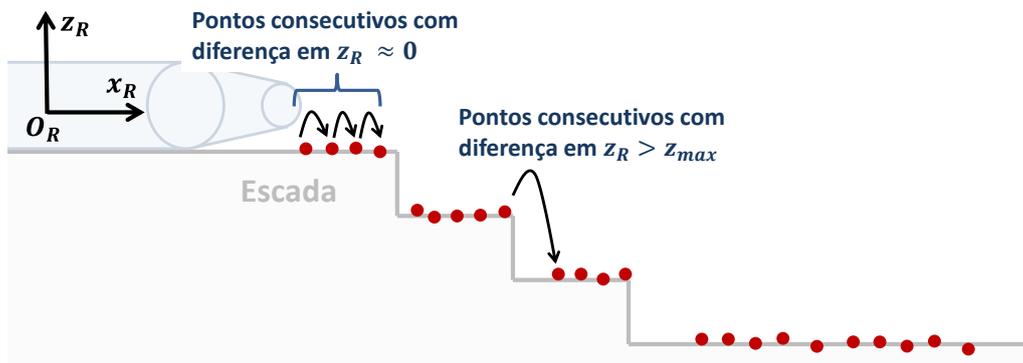


Figura 7.2: Pontos 2D da varredura do *laser* vertical representados no sistema de coordenadas  $O_R$  do robô.

Portanto, semelhante ao método apresentado na seção 3.1, os pontos consecutivos da varredura do *laser* vertical são avaliados, sendo verificado se a diferença na coordenada  $z_R$  é menor que um valor máximo ( $z_{max}$ ). Os pontos consecutivos que satisfazem à condição mencionada são agrupados, de modo que a profundidade  $d_S$  do degrau passa a ser definida pela diferença em  $x_R$  entre o primeiro e o último ponto do grupo avaliado. Após detectar novos grupos de pontos com diferença em  $z_R < z_{max}$ , a altura  $h_S$  dos degraus será definida pela diferença em  $z_R$  entre grupos de pontos consecutivos.

Para este caso, a localização da escada é calculada a partir das coordenadas  $(x_R, y_R)$  do último ponto pertencente ao grupo de pontos do primeiro degrau da

escada. Sendo conhecida a localização do robô, é possível transformar as coordenadas de posição da escada do referencial  $O_R$  para o referencial inercial  $O_B$ . Desta maneira, a localização da escada é designada pela variável  $p_S$  sendo representada no sistema de coordenadas  $O_B$ .

Já a orientação da escada pode ser obtida utilizando o *laser* horizontal. Assim como na seção 3.1, as linhas do ambiente são extraídas a partir dos pontos 2D do *laser*, utilizando o método *Ransac*, e ao encontrar as linhas paralelas referentes às paredes laterais da escada, torna-se possível determinar a sua largura e orientação. A figura 7.3 ilustra a etapa do cálculo da orientação e largura da escada.

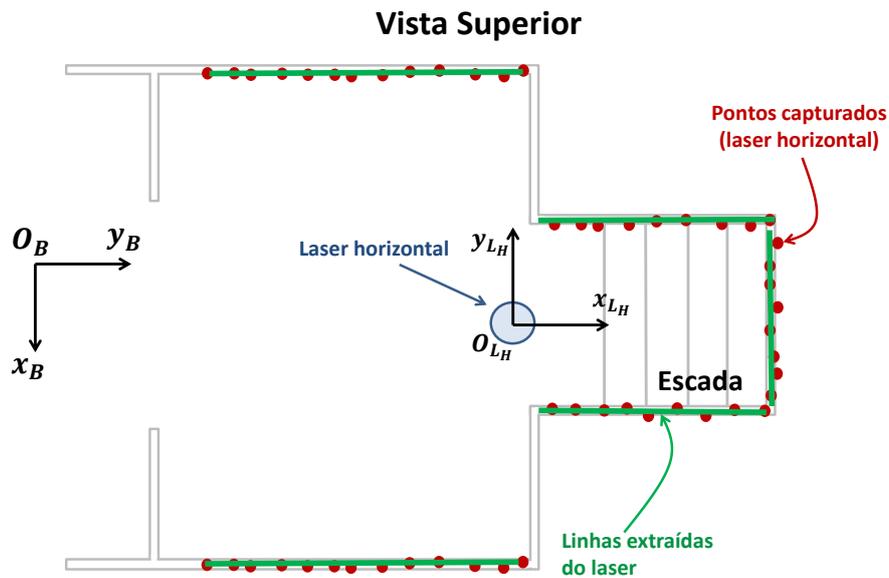


Figura 7.3: Captura das paredes laterais à escada utilizando o *laser* horizontal.

Portanto, ao encontrar as linhas paralelas do ambiente com a menor distância ao ponto  $p_S$  calculado na primeira etapa, é possível obter as linhas laterais à escada, de modo que a distância entre as respectivas linhas define a largura da escada e a direção das linhas define a orientação da escada.

### Detecção e localização da escada utilizando imagem 3D densa

Para o problema de descida da escada, o mesmo método de detecção e modelagem de escadas utilizando imagem 3D densa apresentado na seção 3.2 pode ser aplicado, realizando-se apenas algumas alterações.

Neste caso, o manipulador do robô é utilizado para posicionar o sensor *Kinect* capturando os degraus da escada verticalmente. A figura 7.4 ilustra a posição proposta para o sensor *Kinect* e os sistemas de coordenadas da escada  $O_S$ , robô  $O_R$  e *Kinect*  $O_K$ .

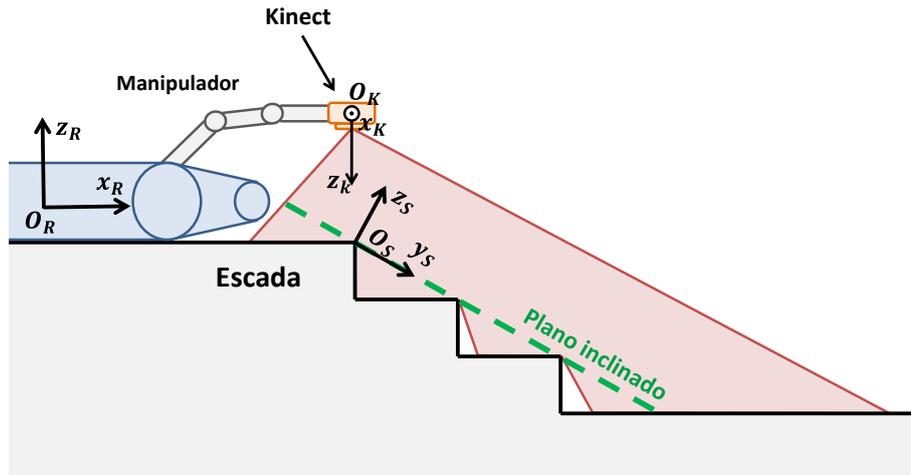


Figura 7.4: Exemplo do posicionamento do sensor *Kinect* para capturar a escada no problema de descida, e a representação dos sistemas de coordenadas adotados.

Portanto, ao posicionar o sensor *Kinect* verticalmente para realizar a captura da escada, ou seja, o eixo  $z_K$  ortogonal ao plano do solo, é possível aplicar a mesma metodologia avaliada na seção 3.2.

Novamente as bordas dos degraus são evidenciadas na imagem em profundidade (*depth image*) e, a partir da detecção das bordas (método de *Canny*) e extração de linhas (*Probabilistic Hough Transform*), são capturadas as linhas paralelas das bordas dos degraus e os correspondentes pontos em 3D. Em seguida, um plano é ajustado aos pontos 3D das bordas dos degraus, sendo verificado se o ângulo de inclinação do plano com o solo é menor que um limite máximo.

Assim como no problema de subida da escada, o modelo da escada pode ser definido como um plano inclinado  $pl$  limitado por um prisma  $B_l$ , então após realizar diversas capturas dos pontos 3D das bordas dos degraus, o plano  $pl$  é ajustado aos pontos, bem como a caixa delimitadora  $B_l$ .

O plano ortogonal transversal  $pt$  é encontrado a partir do plano  $pl$  e centroide da caixa  $B_l$ . Os pontos 3D são projetados em  $pt$ , de modo que a altura  $h_S$  dos degraus passa a ser definida pela diferença em  $z_K$  entre grupos de pontos (bordas dos degraus) consecutivos projetados em  $pt$ , e a profundidade  $d_S$  passa a ser definida pela diferença em  $y_K$  entre os mesmos grupos de pontos.

A localização da escada e a sua orientação são calculadas através do centroide  $(B_{l,cx}, B_{l,cy}, B_{l,cz})$  da caixa delimitadora e do vetor normal ao plano inclinado  $pl$  do mesmo modo como foi realizado na seção 3.2.

**Observação 7.1** *Em ambos os métodos, as informações de posição e orientação da escada devem ser representadas no sistema de coordenadas inercial  $O_B$ , para tal é*

definida a cinemática direta do manipulador utilizado para posicionar o laser vertical, bem como sensor Kinect. Desta maneira, a cinemática direta do manipulador permite realizar a transformação de coordenadas do laser ou Kinect para o sistema de coordenadas do robô  $O_R$  e em seguida para o sistema de coordenadas inercial  $O_B$ , considerando conhecida a posição do robô.

## 7.2.2 Localização baseada em mapa conhecido

O método de localização baseado em mapa conhecido proposto no capítulo 4 também pode ser utilizado no problema de descida da escada, pois as considerações relacionadas ao ambiente e escada foram mantidas as mesmas.

Portanto, durante o deslocamento no piso superior, os dados da odometria e as medidas de posição e orientação obtidas a partir do método de *matching* de *landmarks*, utilizando o *laser*, são unidas através do Filtro de Kalman Estendido (EKF). O EKF, por sua vez, calcula a melhor estimativa da posição  $(x, y)$  e orientação  $(\phi)$  do robô em relação ao sistema de coordenadas inercial  $O_B$ .

Para o deslocamento sobre a escada durante a descida, novamente propõe-se utilizar o método de *matching* de *landmarks* para obter as medidas de posição  $(x)$  e orientação  $(\phi)$  do robô em relação ao sistema de coordenadas da escada  $O_S$ . Assim, os dados do *laser* capturando as paredes laterais são utilizados no método para obter as informações desejadas, do mesmo modo apresentado na seção 4.3.

## 7.2.3 Metodologia para locomoção em escadas durante a descida

Para que o robô realize a locomoção durante a descida na escada, novamente é proposto um conjunto de configurações das pernas robóticas que permitam executar cada uma das etapas necessárias. Desta maneira, semelhante à metodologia apresentada no capítulo 5, o problema foi dividido em três etapas:

- a) Transição suave do piso superior para a escada;
- b) Deslocamento na escada; e
- c) Transição da escada para o piso inferior.

Diferentemente do problema de subida da escada em que a etapa crítica é a transição da escada para o piso superior, neste caso o ponto crítico é a transição do piso superior para a escada.

Durante a transição do piso superior para a escada, caso não haja um correto posicionamento das pernas, no instante em que a coordenada  $x$  do centro de massa do robô ultrapassa a posição da borda do primeiro degrau em contato com as esteiras, o momento das forças que atuam no robô fazem com que ele realize uma queda

brusca na escada, havendo grandes possibilidades de ocorrer um tombamento sobre os degraus.

Portanto, novamente podemos utilizar a cinemática direta aplicada ao robô definido como três corpos rígidos unidos por juntas de revolução, sendo o corpo 1 as pernas traseiras, o corpo 0 o chassi e o corpo 2 as pernas frontais. Após determinar a cinemática direta do robô é possível então definir o centro de massa do robô e a configuração das pernas necessária a cada uma das três etapas.

### Transição do piso superior para a escada

Semelhante à abordagem de transição da escada para o piso superior apresentada no capítulo 5, é proposto separar esta etapa em duas fases: SD-A e SD-B.

Na fase SD-A as pernas frontais e traseiras devem ser posicionadas em ângulos  $\theta_1 = 225^\circ$  e  $\theta_2 = -45^\circ$ , como representado na figura 7.5.

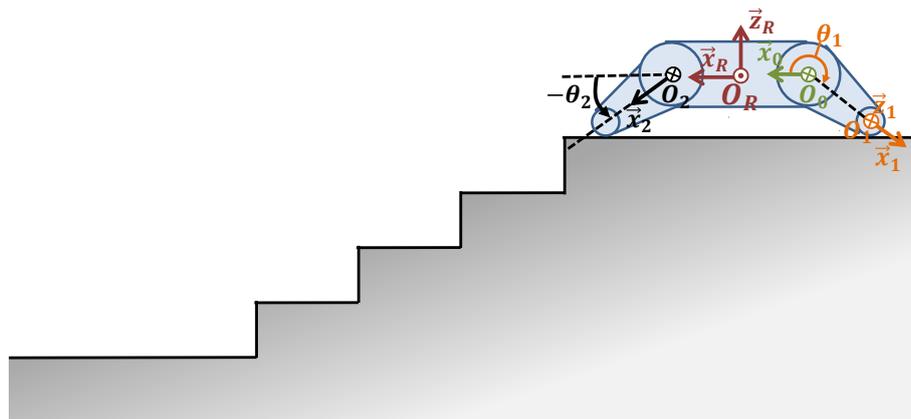


Figura 7.5: Exemplo do posicionamento das pernas robóticas durante a fase SD-A.

Com as pernas na posição indicada, as esteiras são tracionadas conduzindo o robô na direção de descida da escada. No momento em que a posição horizontal do centro de massa do robô ultrapassa o ponto de contato entre o primeiro degrau da escada e o chassi, as forças de interação e a gravidade geram um momento fazendo com que o robô realize a transição do piso superior para a escada. A posição das pernas abaixo da linha horizontal do chassi permite que o robô realize uma transição suave, evitando assim um possível tombamento sobre os degraus.

Utilizando a informação da posição ( $p_S$ ) do primeiro degrau da escada obtida na etapa de detecção e modelagem da mesma, em conjunto com os dados da odometria e sensor inercial do robô, é possível estimar o instante em que o robô realiza a transição do piso superior para a escada. Com as informações mencionadas, estima-se a distância horizontal do centro de massa do robô ao primeiro degrau da escada, de modo que ao detectar que a referida distância se torna zero, é iniciada então a

etapa SD-B.



Figura 7.6: Exemplo da transição do piso superior para escada, fase SD-A.

Na etapa SD-B, o robô já se encontra completamente posicionado sobre os primeiros degraus da escada, desta maneira o robô deve tornar seu ângulo de inclinação ( $\alpha_R$ ) equivalente ao ângulo de inclinação da escada ( $\alpha_S$ ). Para tal, as pernas são posicionadas de modo que a parte inferior de cada uma delas esteja no mesmo plano da base do chassi. A figura 7.7 ilustra a posição das pernas e robô durante a fase SD-B.



Figura 7.7: Exemplo da transição do piso superior para escada, fase SD-B.

Portanto, ao final da fase SD-B o robô já se encontra completamente posicionado sobre os primeiros degraus da escada, de modo que seu ângulo de inclinação é igual ao ângulo de inclinação da escada ( $\alpha_R = \alpha_S$ ).

## Deslocamento na escada

O problema de deslocamento na escada durante a descida é praticamente idêntico à subida. Novamente o movimento do robô sobre a escada é modelado como em um plano inclinado definido pelas bordas dos degraus.

A configuração do robô adotada é a mesma utilizada ao final da fase SD-B em que a parte inferior das pernas está no mesmo plano da base do chassi, permitindo assim uma maior superfície de contato do chassi e pernas com os degraus.

Durante o deslocamento na escada, o objetivo é manter o robô equidistante das laterais com a sua orientação na direção de descida, evitando, assim, possíveis colisões ou escorregamentos entre esteiras e degraus.

Portanto, é proposto utilizar o mesmo controle apresentado no capítulo 6 para conduzir o robô durante o deslocamento na escada. O controle em questão, como

já descrito, é baseado em rastreamento de trajetórias, de forma que é definida uma trajetória retilínea equidistante das laterais da escada e na direção de descida, permitindo que o robô siga a trajetória desejada do início da escada até os degraus finais.

Para detectar a proximidade com o final da escada, ou seja, a proximidade com o piso inferior, o robô pode utilizar tanto o sensor *laser* vertical quanto o sensor *Kinect*.

A partir dos dados do *laser* vertical, é possível realizar a extração de linhas e determinar a distância do robô à reta longa com inclinação horizontal referente ao piso inferior.

Utilizando o sensor *Kinect*, a detecção do piso inferior pode ser executada identificando o maior plano horizontal ajustado aos dados 3D (*point cloud*) fornecidos pelo sensor. Desta maneira, determinando a distância do robô ao plano em questão, é estimada a distância do robô até o piso inferior.

### Transição da escada para o piso inferior

Após detectar a proximidade do robô com o piso inferior, é iniciada a última etapa da metodologia de locomoção em escadas durante a descida. A transição da escada para o piso inferior é a etapa menos crítica em relação às anteriores, pois o robô já se encontra bem próximo ao solo, de modo que o posicionamento das pernas no mesmo plano da base do chassi auxilia a aterrissagem.

Para realizar uma transição suave para piso inferior, é proposto manter as pernas frontais alinhadas ao plano da base do chassi e as pernas traseiras em posição acima do plano do chassi ( $\theta_1 = 135^\circ$ ). A configuração em questão pode ser observada na figura 7.8.

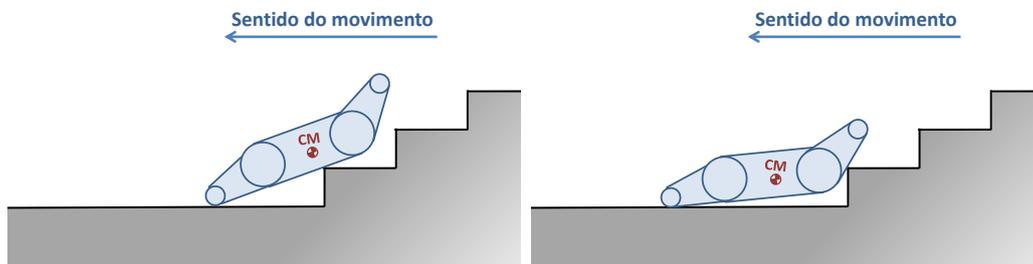


Figura 7.8: Exemplo da transição suave da escada para o piso inferior.

Desta maneira, realizando o movimento das esteiras em direção à descida da escada, a posição das pernas traseiras representada na figura 7.8 permite que o robô execute uma transição suave da escada para o piso inferior.

## 7.3 Conclusões

Neste capítulo foi proposta uma solução para o problema de locomoção em escadas durante a descida. Para tal, foram utilizadas todas as ferramentas avaliadas nos capítulos anteriores para o problema de subida da escada, sendo adaptadas caso a caso para a tarefa de descida.

A detecção e modelagem da escada foi proposta baseando-se nos métodos utilizando os sensores *laser* e o sensor *Kinect*. Ambos os métodos foram adaptados acoplando o *laser* vertical e o *Kinect* no manipulador do robô, para permitir uma melhor captura da escada. As alterações em cada método foram propostas para lidar com a mudança de orientação dos sensores em relação à escada.

Para realizar a locomoção na escada durante a descida, foi proposto um método de reconfiguração da postura do robô através das pernas, baseando-se no método utilizado para o problema de subida (capítulo 5). O problema foi dividido novamente em três etapas e para realizar cada etapa foram definidas as posições das pernas correspondentes.

Durante a descida da escada, foi proposto utilizar o mesmo controle apresentado no capítulo 6 baseado em rastreamento de trajetórias, de modo a conduzir o robô dos degraus iniciais até o piso inferior.

# Capítulo 8

## Conclusões

Esta dissertação apresentou uma abordagem para locomoção semiautônoma em escadas aplicada a robôs móveis com esteiras. O objetivo do método proposto é auxiliar a operação dos referidos sistemas robóticos em aplicações de exploração e resgate em áreas atingidas por desastres.

É crescente a utilização de robôs para auxiliar equipes de resgate em locais hostis ao ser humano, como áreas com destroços ou presença de radioatividade. Para explorar as diversas áreas é necessário que o robô supere obstáculos como escombros, escadas, dentre outros. Em virtude da larga presença de escadas em ambientes urbanos, este obstáculo se tornou o principal foco do estudo desenvolvido nesta dissertação.

A teleoperação dos robôs fornece uma visão limitada do ambiente, o que incentiva a utilização de estratégias autônomas ou semiautônomas para superar diferentes obstáculos e auxiliar a operação.

Na formulação do problema de subida da escada, foram considerados ambientes interiores de construções prediais, de modo que as escadas podem apresentar inclinação entre  $18^\circ$  e  $40^\circ$  e o robô inicia o movimento de um ponto distante da escada, finalizando a operação no pavimento superior.

Os robôs considerados no problema são classificados como EOD, que são robôs móveis com esteiras e direção diferencial, apresentando ainda quatro ou duas pernas robóticas, frontais e traseiras. As pernas permitem que o robô reconfigure sua postura, auxiliando a superação dos obstáculos. Para percepção do ambiente foi considerado no trabalho que o robô apresenta dois sensores *lasers scan*, um para escaneamentos verticais e outro para escaneamentos horizontais, além de uma câmera 3D do tipo RGB-D como o *Kinect* e um sensor inercial IMU.

Para detecção e modelagem da escada foram apresentadas duas abordagens, uma utilizando os *lasers scan* e outra o *Kinect*. A primeira utiliza o escaneamento vertical da escada para detectar a geometria característica dos degraus e o *laser* horizontal estima a localização e orientação da mesma. Já o segundo método é

baseado na detecção das bordas da escada através do sensor *Kinect* e no ajuste do modelo aos dados obtidos. Ambos os métodos foram testados experimentalmente, de modo que a primeira abordagem apresentou bons resultados, entretanto é sensível a ruídos do sensor e longas distâncias entre robô e escada. Por outro lado, o método baseado no sensor 3D apresentou bons resultados, se mostrando robusto a ruídos do sensor e variação da orientação entre robô e escada. Ressalta-se ainda que a segunda estratégia foi capaz de estimar os parâmetros da escada como altura e profundidade dos degraus com boa precisão se comparados aos ruídos do *Kinect*, além de estimar a inclinação, localização e orientação da escada.

A estratégia de localização no ambiente, durante a aproximação até a escada, foi realizada com base nos dados do *laser* posicionado horizontalmente e dados da odometria do robô. O modelo da odometria foi apresentado para obter a posição e orientação a partir das velocidades das esteiras. Um método de *matching* de *landmarks* foi utilizado para obter a medida da posição e orientação do robô, através do *laser*. A fusão dos dados do sensor *laser* e odometria foi realizada através da aplicação do Filtro de Kalman Estendido (EKF) que fornece a melhor estimativa da posição e orientação do robô.

O método de *matching* de *landmarks* proposto permite obter a medida de posição e orientação do robô através do *laser scan*. A partir das coordenadas 2D do *laser*, as linhas do ambiente são extraídas e comparadas às linhas do mapa conhecido, de modo que o método minimiza as funções do erro de posição e orientação definidas a priori. Ao final, é obtida a informação de posição e orientação do robô em relação ao mapa do ambiente, informação esta posteriormente utilizada no EKF.

Para a localização durante a locomoção sobre a escada, também foi proposto um método de *matching* de *landmarks*. O *laser* posicionado horizontalmente detecta as paredes laterais da escada, de modo que os *landmarks* são extraídos e comparados aos *landmarks* do mapa, resultando na medida de orientação ( $\phi$ ) e coordenada ( $x$ ) da posição do robô em relação ao sistema de coordenadas da escada  $O_S$ .

O método de localização foi avaliado através de simulações e experimentos reais, utilizando o robô móvel com duas rodas e direção diferencial conhecido como *Romba*. O *laser* foi acoplado horizontalmente ao robô e ambos foram posicionados no corredor de um prédio, como definido na formulação do problema. As simulações bem como os experimentos apresentaram resultados semelhantes, de modo que o EKF implementado em conjunto com o método de *matching* de *landmarks* e os dados da odometria foram capazes de estimar a posição e orientação do robô com boa precisão, tendo em vista os ruídos do *laser* e odometria.

Para superar obstáculos como escadas, definiu-se um conjunto de posições das pernas robóticas que permitem a reconfiguração do robô e o auxilia na travessia da escada. Deste modo, o problema foi dividido em três etapas: transição do piso

inferior para a escada, locomoção propriamente dita sobre a mesma e aterrissagem suave no piso superior.

A primeira etapa de transição do piso inferior para a escada foi dividida em três fases, de maneira que cada fase apresenta uma configuração característica das pernas para auxiliar a elevação do robô sobre os primeiros degraus.

Para a segunda etapa, a locomoção propriamente dita sobre a escada, foi proposto um posicionamento em que a parte inferior das pernas robóticas se encontra no mesmo plano da base do chassi do robô. Esta configuração aumenta a superfície de contato entre robô e degraus, auxiliando portanto o deslocamento.

A terceira e última etapa de reconfiguração, a aterrissagem suave no piso superior, foi realizada através de uma metodologia de avanço gradual das esteiras e medições da variação da inclinação do robô decorrente do contato das pernas frontais com o piso superior.

As três etapas de reconfiguração das pernas robóticas foram validadas através da simulação cinemática dos movimentos do robô e configurações propostas.

Uma estratégia de controle cinemático baseada em rastreamento de trajetórias foi apresentada para realizar a locomoção do robô sobre a escada. Para simplificar o problema, foi aproximado o movimento do robô sobre a escada como o movimento em um plano inclinado. O objetivo do controle consiste em definir uma trajetória retilínea equidistante em relação às laterais da escada, de modo que o robô segue a trajetória, mantendo a sua orientação na direção de subida, ou seja, a direção retilínea desejada. A estabilidade do sistema é avaliada através da teoria de Lyapunov, de onde se conclui que a origem do sistema em malha fechada é globalmente assintoticamente estável.

O mesmo controle foi aplicado na etapa anterior à travessia da escada, durante a aproximação do robô do ponto inicial até a posição próxima à mesma. Para tal, a trajetória desejada é definida pela reta que une a posição inicial do robô e a posição final desejada próxima à escada. O controle é aplicado da mesma maneira como ocorre durante a locomoção sobre a escada, de modo que ao alcançar o ponto desejado, a velocidade linear do robô é interrompida, convertendo o controle de posição e orientação em um controle de orientação apenas. No momento em que ocorre o chaveamento do controle, a orientação desejada é definida como a orientação da escada, permitindo assim que o robô seja alinhado com a mesma.

Ambas as aplicações do controle, para a locomoção sobre a escada e para a aproximação até a mesma, foram validadas através de simulações. Para avaliar a robustez do controle, foram adicionadas incertezas nos parâmetros do modelo do robô além de fatores de escorregamento, alterando as velocidades das esteiras direita e esquerda.

As simulações demonstraram que o controle é capaz de conduzir o robô através da

trajetória desejada mesmo inserindo escorregamentos e incertezas paramétricas no modelo. Ressalta-se que para a locomoção sobre a escada, a velocidade linear do robô foi variada como uma senoide onde os intervalos de sinal negativo foram atribuídos valor zero, caracterizando paradas do robô durante o trajeto. O resultado positivo observado foi que nos instantes de parada, o controle de orientação age alinhando o robô à direção da trajetória, o que reduz um possível escorregamento causado pelo aumento do erro de orientação do robô em relação à direção de subida da escada.

O problema de locomoção em escadas durante a descida também foi formulado, sendo proposta uma solução baseada nas ferramentas apresentadas para o problema de subida da escada.

## 8.1 Trabalhos Futuros

Esta dissertação propôs uma metodologia para locomoção semiautônoma de robôs EODs em escadas. A etapa de detecção e localização da escada bem como a de localização no ambiente foram avaliadas experimentalmente utilizando o robô *Roomba* com duas rodas e direção diferencial, dois sensores *lasers scan* e o sensor *Kinect*. Entretanto, as outras etapas foram validadas através de simulações apenas. Portanto, os trabalhos futuros podem implementar a estratégia de controle para locomoção em escadas, além das abordagens de reconfiguração das pernas robóticas em um robô EOD real, de modo a validar a metodologia proposta através de experimentos.

Os métodos de reconfiguração das pernas podem ser apresentados utilizando uma abordagem de sistema a eventos discretos, de modo que nos estados podem existir malhas de controle de orientação contínuas atuando nas pernas robóticas, como foi discutido no capítulo 5, a fim de realizar as etapas de transição do piso inferior para a escada e aterrissagem sobre o piso superior. A abordagem utilizando estados e malhas de controle de orientação podem futuramente ser formalizadas, desenvolvidas e aplicadas em um robô móvel com pernas e esteiras.

Outra sugestão de trabalho futuro seria implementar as ferramentas avaliadas para locomoção semiautônoma em escadas durante a subida, adaptadas para realizar experimentos durante a descida da escada.

A locomoção sobre a escada foi aproximada como a locomoção em um plano inclinado definido pelas bordas dos degraus. Desta maneira, foi aplicado o controle cinemático para conduzir o robô sobre o plano inclinado, rastreando a trajetória desejada. A aproximação mencionada e a proposta de um controle cinemático são válidas se considerarmos que a força de atrito entre as esteiras e os degraus apresenta magnitude suficiente para evitar escorregamentos durante a tração das esteiras. A hipótese anterior também elimina a possibilidade de escorregamentos em virtude da variação da orientação do robô em relação à direção de subida da escada. Desta ma-

neira, uma sugestão para trabalho futuro seria considerar a possibilidade da força de atrito não ser suficiente para sustentar a hipótese de não escorregamento, de modo que deve ser realizada uma análise dinâmica do problema e considerar o escorregamento no desenvolvimento da estratégia de controle.

Os métodos de detecção de escadas descritos no texto consideram apenas as escadas com espelhos, ou seja, aquelas que não apresentam vazios entre os degraus. Portanto, futuros trabalhos podem aprimorar as metodologias de detecção de escadas para considerar escadas sem espelhos entre os degraus, tornando os métodos mais robustos a uma variedade maior de escadas.

Considerou-se ainda a presença de paredes laterais à escada para permitir que o robô se localize em relação ao sistema de coordenadas  $O_S$  da mesma. Uma outra abordagem futura seria o desenvolvimento de uma nova estratégia de localização sobre a escada que não necessite considerar paredes ou balaustres laterais.

# Referências Bibliográficas

- Adiwahono, A., Saputra, B., Chang, T. W. & Yong, Z. X. (2014), ‘Autonomous stair identification, climbing, and descending for tracked robots’, *IEEE International Conference on Control Automation Robotics & Vision* pp. 48–53.
- Armesto, L. & Tornero, J. (2006), ‘Robust and efficient mobile robot self-localization using laser scanner and geometrical maps’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 3080–3085.
- Arun, K., Huang, T. & Blostein, S. (1987), ‘Least-squares fitting of two 3-d point sets’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-9**(5), 698–700.
- Atmeh, G., Ranatunga, I., Popa, D. & Subbarao, K. (2014), ‘Implementation of an adaptive, model free, learning controller on the atlas robot’, *American Control Conference* pp. 2887–2892.
- Brooks, A., Kaupp, T. & Makarenko, A. (2009), ‘Randomised mpc-based motion-planning for mobile robot obstacle avoidance’, *IEEE International Conference on Robotics and Automation* pp. 3962–3967.
- Brown, R. & Hwang, P. (1997), *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions*, Vol. 1 of *Introduction to Random Signals and Applied Kalman Filtering: With MATLAB Exercises and Solutions*, 3 edn, Wiley.
- Canny, J. (1986), ‘A computational approach to edge detection’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**(6), 679–698.
- Canudas de Wit, C., Siciliano, B. & Bastin, G. (1996), *Theory of robot control*, Springer.
- Carbonara, S. & Guaragnella, C. (2014), ‘Efficient stairs detection algorithm assisted navigation for vision impaired people’, *IEEE International Symposium on Innovations in Intelligent Systems and Applications* pp. 313–318.

- Cassandras, C. G. & Lafortune, S. (2008), *Introduction to Discrete Event Systems*, 2 edn, Springer.
- Choi, M., Choi, J., Nam, S. Y. & Chung, W. K. (2012), ‘Direction augmented probabilistic scan matching’, *IEEE International Conference on Robotics and Automation* pp. 3607–3612.
- Christiand, Lee, Y.-C., Yu, W. & Cho, J. (2010), ‘Augmented EKF localization for mobile robots in urban environments’, *IEEE International Conference on Robotics and Biomimetics* pp. 986–991.
- Colas, F., Mahesh, S., Pomerleau, F., Liu, M. & Siegwart, R. (2013), ‘3d path planning and execution for search and rescue ground robots’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 722–727.
- Corke, P. (2011), *Robotics, Vision and Control - Fundamental Algorithms in MATLAB*, Springer.
- Dellaert, F., Fox, D., Burgard, W. & Thrun, S. (1999), ‘Monte carlo localization for mobile robots’, *IEEE International Conference on Robotics and Automation* .
- Delmerico, J. A., Baran, D., David, P., Ryde, J. & Corso, J. J. (2013), ‘Ascending stairway modeling from dense depth imagery for traversability analysis’, *IEEE International Conference on Robotics and Automation* pp. 2283–2290.
- Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D. & Burgard, W. (2012), ‘An evaluation of the rgb-d slam system’, *IEEE International Conference on Robotics and Automation* pp. 1691–1696.
- Fallon, M., Antone, M., Roy, N. & Teller, S. (2014), ‘Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing’, *IEEE/RAS International Conference on Humanoid Robots* pp. 112–119.
- Fischler, M. A. & Bolles, R. C. (1981), ‘Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography’, *Communications of the ACM* pp. 381–395.
- Freitas, G. M. (2008), Reconfiguração cinemática quase-estática de robôs móveis em terrenos irregulares, Master’s thesis, UFRJ.
- Helmick, D., Roumeliotis, S., McHenry, M. & Matthies, L. (2002), ‘Multi-sensor, high speed autonomous stair climbing’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 733–742.

- Jang, G., Kim, S., Kim, J. & Kweon, I. (2005), ‘Metric localization using a single artificial landmark for indoor mobile robots’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 2857–2862.
- Kalantari, A., Mihankhah, E. & Moosavian, S. (2009), ‘Safe autonomous stair climbing for a tracked mobile robot using a kinematics based controller’, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* pp. 1891–1896.
- Khalil, H. K. (1996), *Nonlinear systems*, 2 edn, Prentice Hall.
- Kiryati, N., Eldar, Y. & Bruckstein, A. (1991), ‘A probabilistic hough transform’, *Pattern recognition* **24**(4), 303–316.
- Li, J., Bao, J. & Yu, Y. (2010), ‘Study on localization for rescue robots based on NDT scan matching’, *IEEE International Conference on Information and Automation (ICIA)* pp. 1908–1912.
- Li, N., Ma, S., Li, B., Wang, M. & Wang, Y. (2012), ‘An online stair-climbing control method for a transformable tracked robot’, *IEEE International Conference on Robotics and Automation* pp. 923–929.
- Lionis, G. S. & Kyriakopoulos, K. J. (2002), ‘A laser scanner based mobile robot slam algorithm with improved convergence properties’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 582–587.
- Lizarralde, F., Nunes, E., Hsu, L. & Wen, J. (2003), ‘Mobile robot navigation using sensor fusion’, *IEEE International Conference on Robotics and Automation* **1**, 458–463.
- Luo, J., Zhang, Y., Hauser, K., Park, H., Paldhe, M., Lee, C., Grey, M., Stilman, M., Oh, J. H., Lee, J., Kim, I. & Oh, P. (2014), ‘Robust ladder-climbing with a humanoid robot with application to the DARPA Robotics Challenge’, *IEEE International Conference on Robotics and Automation* pp. 2792–2798.
- Luo, R., Hsiao, M. & Liu, C.-W. (2013a), ‘Erect wheel-legged stair climbing robot for indoor service applications’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 2731–2736.
- Luo, R., Hsiao, M. & Liu, C.-W. (2013b), ‘Multisensor integrated stair recognition and parameters measurement system for dynamic stair climbing robots’, *IEEE International Conference on Automation Science and Engineering* pp. 318–323.

- Magid, E., Tsubouchi, T., Koyanagi, E. & Yoshida, T. (2010), ‘Static balance for rescue robot navigation : Losing balance on purpose within random step environment’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 349–356.
- Marques, C., Cristóvão, J. a., Lima, P., Frazão, J. a., Ribeiro, I. & Ventura, R. (2006), ‘Raposa: Semi-autonomous robot for rescue operations’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 3988–3993.
- Mihankhah, E., Tehran Kalantari, A., Aboosaeedan, E., Taghirad, H., Ali, S. & Moosavian, A. (2009), ‘Autonomous staircase detection and stair climbing for a tracked mobile robot using fuzzy controller’, *IEEE International Conference on Robotics and Biomimetics* pp. 1980–1985.
- Mourikis, A. I., Trawny, N., Roumeliotis, S. I., Helmick, D. M. & Matthies, L. (2007), ‘Autonomous stair climbing for tracked vehicles’, *International Journal of Robotics Research* .
- Murphy, R. R. (2012), ‘A decade of rescue robots’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 5448–5449.
- Nguyen, V., Martinelli, A., Tomatis, N. & Siegwart, R. (2005), ‘A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics’, *IEEE/RSJ International Conference on Intelligent Robots and Systems* pp. 1929–1934.
- Shao, L., Han, J., Xu, D. & Shotton, J. (2013), ‘Computer vision for rgb-d sensors: Kinect and its applications’, *IEEE Transactions on Cybernetics* pp. 1314–1317.
- Siciliano, B., Sciavicco, L., Villani, L. & Oriolo, G. (2009), *Robotics - Modelling, Planning and Control*, Springer.
- Siegwart, R., Nourbakhsh, I. R. & Scaramuzza, D. (2004), *Introduction to Autonomous Mobile Robots*, 2 edn, The MIT Press.
- Slotine, J.-J. E. & Li, W. (1991), *Applied Nonlinear Control*, Prentice Hall.
- Stentz, A., Herman, H., Kelly, A., Meyhofer, E., Haynes, G. C., Stager, D., Zajac, B., Bagnell, J. A., Brindza, J., Dellin, C., George, M., Gonzalez-Mora, J., Hyde, S., Jones, M., Laverne, M., Likhachev, M., Lister, L., Powers, M., Ramos, O., Ray, J., Rice, D., Scheifflee, J., Sidki, R., Srinivasa, S., Strabala, K., Tardif, J.-P., Valois, J.-S., Weghe, J. M. V., Wagner, M. &

- Wellington, C. (2015), ‘Chimp, the CMU highly intelligent mobile platform’, *Journal of Field Robotics* .
- Thrun, S., Burgard, W. & Fox, D. (2005), *Probabilistic Robotics*, The MIT Press.
- Valenti, R., Dryanovski, I., Jaramillo, C., Perea Strom, D. & Xiao, J. (2014), ‘Autonomous quadrotor flight using onboard rgb-d visual odometry’, *IEEE International Conference on Robotics and Automation* pp. 5233–5238.
- Wang, W., Wu, D., Wang, Q., Deng, Z. & Du, Z. (2012), ‘Stability analysis of a tracked mobile robot in climbing stairs process’, *IEEE International Conference on Mechatronics and Automation* pp. 1669–1674.
- Zhang, Q., Ge, S. S. & Tao, P. Y. (2011), ‘Autonomous stair climbing for mobile tracked robot’, *IEEE International Symposium on Safety, Security and Rescue Robotics* .

# Apêndice A

## Cálculo da matriz de rotação ótima $R_l$

A função custo a ser minimizada apresenta o seguinte formato:

$$E_\phi(R_l) = \sum_{i=1}^{n_l} \|u_{li}^m - R_l u_{li}^d\|^2 \quad (\text{A.1})$$

onde  $R_l$  é a matriz de rotação ortogonal ótima que minimiza a função custo descrita. Expandindo a função, temos:

$$\begin{aligned} E_\phi(R_l) &= \sum_{i=1}^{n_l} \|u_{li}^m - R_l u_{li}^d\|^2 \\ &= \sum_{i=1}^{n_l} (u_{li}^m - R_l u_{li}^d)^T (u_{li}^m - R_l u_{li}^d) \\ &= \sum_{i=1}^{n_l} (u_{li}^{mT} u_{li}^m - u_{li}^{mT} R_l u_{li}^d - (R_l u_{li}^d)^T u_{li}^m + (R_l u_{li}^d)^T R_l u_{li}^d) \\ &= \sum_{i=1}^{n_l} [(\|u_{li}^m\|^2 + \underbrace{(R_l u_{li}^d)^T R_l u_{li}^d}_{\|u_{li}^d\|^2}) - (u_{li}^{mT} R_l u_{li}^d + (R_l u_{li}^d)^T u_{li}^m)] \\ &= \sum_{i=1}^{n_l} (\|u_{li}^m\|^2 + \|u_{li}^d\|^2) - \sum_{i=1}^{n_l} (u_{li}^{mT} R_l u_{li}^d + (R_l u_{li}^d)^T u_{li}^m) \\ &= \sum_{i=1}^{n_l} (\|u_{li}^m\|^2 + \|u_{li}^d\|^2) - 2 \underbrace{\sum_{i=1}^{n_l} (u_{li}^{mT} R_l u_{li}^d)}_{\bar{E}_\phi} \end{aligned}$$

Portanto, minimizar a função custo  $E_\phi(R_l)$  é equivalente a maximizar a função  $\bar{E}_\phi$ .

$$\begin{aligned}\bar{E}_\phi &= \sum_{i=1}^{n_l} u_{li}^{mT} R_l u_{li}^d \\ \bar{E}_\phi &= \text{trace} \left( \sum_{i=1}^{n_l} R_l u_{li}^d u_{li}^{mT} \right) = \text{trace} (R_l S_{uu^T})\end{aligned}\quad (\text{A.2})$$

em que

$$S_{uu^T} = \sum_{i=1}^{n_l} u_{li}^d u_{li}^{mT} \quad (\text{A.3})$$

Realizando a decomposição em valores singulares da matriz  $S_{uu^T}$ , temos:

$$S_{uu^T} = U \Sigma V^T \quad (\text{A.4})$$

onde  $U$  e  $V$  são matrizes ortogonais  $3 \times 3$  e  $\Sigma$  é uma matriz diagonal  $3 \times 3$  com elementos não negativos.

Substituindo a decomposição em valores singulares (A.4) na equação (A.2) temos

$$\bar{E}_\phi = \text{trace} (R_l U \Sigma V^T) \quad (\text{A.5})$$

O traço de matrizes apresenta a seguinte propriedade para matrizes quadradas: Sejam as matrizes  $A$  e  $B$  quadradas

$$\text{trace} (AB) = \text{trace} (BA)$$

Aplicando a propriedade do traço das matrizes apresentada acima, temos:

$$\bar{E}_\phi = \text{trace} (R_l U \Sigma V^T) = \text{trace} (\Sigma V^T R_l U) \quad (\text{A.6})$$

Dado que  $R_l$  é uma matriz de rotação ortogonal por definição e as matrizes  $U$  e  $V$  também são matrizes ortogonais, desta maneira denotando  $M = V^T R_l U$  temos:

$$MM^T = (V^T R_l U)(V^T R_l U)^T = V^T R_l \underbrace{UU^T}_I R_l^T V = V^T \underbrace{R_l R_l^T}_I V = V^T V = I$$

da mesma maneira  $M^T M = I$ . Logo,  $M$  também é uma matriz ortogonal.

Substituindo  $M$  em (A.6), temos

$$\bar{E}_\phi = \text{trace}(\Sigma M) = \text{trace} \left( \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \right) = \sigma_1 m_{11} + \sigma_2 m_{22} + \sigma_3 m_{33} \quad (\text{A.7})$$

Como  $M$  é ortogonal, seus vetores colunas são ortonormais, assim para cada coluna  $m_j$  de  $M$  temos  $\|m_j\| = 1$ . Desta maneira, todos os elementos  $|m_{ij}| \leq 1$ . Assim, podemos concluir a seguinte inequação:

$$\bar{E}_\phi = \sigma_1 m_{11} + \sigma_2 m_{22} + \sigma_3 m_{33} \leq \sigma_1 + \sigma_2 + \sigma_3 \quad (\text{A.8})$$

Portanto,  $\bar{E}_\phi$  é maximizado quando  $m_{ii} = 1$ . Com  $M$  ortogonal, então  $M = I$  é a matriz identidade.

Desta maneira, a matriz  $R_l$  ótima é definida como

$$\begin{aligned} M = I &= V^T R_l U \\ U^T &= V^T R_l \\ R_l &= V U^T \end{aligned} \quad (\text{A.9})$$

# Apêndice B

## Lemas e provas: Controle para locomoção em escadas

### B.1 Lema de Barbalat

Lema de Barbalat apresentado em Khalil (1996) é descrito a seguir:

**Lema B.1** *Seja  $f : \mathbb{R} \rightarrow \mathbb{R}$  uma função uniformemente contínua em  $[0, \infty)$ . Supondo que o  $\lim_{t \rightarrow \infty} \int_0^t f(\tau) d\tau$  existe e é finito. Então,*

$$f(t) \rightarrow 0 \quad \text{quando } t \rightarrow \infty \quad (\text{B.1})$$

### B.2 Prova que a derivada $d(v_c^3 l \frac{\sin(e_\phi)}{e_\phi})/dt$ é limitada

Supondo  $v_c, \dot{v}_c, l, \dot{l}, e_\phi$  limitados por hipótese.

Derivando o termo  $(v_c^3 l \frac{\sin(e_\phi)}{e_\phi})$ , temos:

$$\begin{aligned} \frac{d}{dt} \left( v_c^3 l \frac{\sin(e_\phi)}{e_\phi} \right) &= 3v_c^2 \dot{v}_c l \frac{\sin(e_\phi)}{e_\phi} + v_c^3 \left( \dot{l} \frac{\sin e_\phi}{e_\phi} + l \frac{\cos(e_\phi) \dot{e}_\phi e_\phi - \dot{e}_\phi \sin(e_\phi)}{e_\phi^2} \right) \\ &= \underbrace{3v_c^2 \dot{v}_c l \frac{\sin(e_\phi)}{e_\phi} + v_c^3 \dot{l} \frac{\sin(e_\phi)}{e_\phi}}_{\gamma_1(t)} + v_c^3 l \dot{e}_\phi \underbrace{\left( \frac{\cos(e_\phi) e_\phi - \sin(e_\phi)}{e_\phi^2} \right)}_{\gamma_2(t)} \end{aligned}$$

Com  $v_c, \dot{v}_c, l, \dot{l}, e_\phi$  limitados, conclui-se que os dois primeiros termos representados por  $\gamma_1(t)$  são limitados, restando agora demonstrar que a última parcela  $\gamma_2(t)$  é limitada.

O termo  $\gamma_2(t)$  apresenta uma indeterminação do tipo  $\frac{0}{0}$  quando  $e_\phi \rightarrow 0$ . Aplicando a regra de *L'Hôpital* temos:

$$\begin{aligned} \lim_{e_\phi \rightarrow 0} \left[ \frac{d(\cos(e_\phi)e_\phi - \sin(e_\phi))/de_\phi}{d(e_\phi^2)/de_\phi} \right] &= \lim_{e_\phi \rightarrow 0} \left[ \frac{(-\sin(e_\phi)e_\phi + \cos(e_\phi) - \cos(e_\phi))}{2e_\phi} \right] \\ &= \lim_{e_\phi \rightarrow 0} \left[ -\frac{\sin(e_\phi)}{2} \right] = 0 \end{aligned}$$

Portanto, a indeterminação do termo  $\gamma_2(t)$  converge para zero.

A partir dos resultados anteriores, conclui-se que  $d(v_c^3 l \frac{\sin(e_\phi)}{e_\phi})/dt$  é limitada.