

ERRATA

Pg. 23. No último parágrafo, após: " ... é também adequada.", adicionar:

"A parábola estimada, dado que a banda vertical é de aproximadamente ± 4 pixels em torno da reta estimada pela Transformada de Hough, é uma extrapolação para além do domínio no qual medidas foram usadas para estimar os coeficientes a_1, a_2 e a_3 . Esta região de extrapolação é vista onde a guia apresenta grande curvatura. A similaridade entre ambas é circunstancial neste caso.

Pg. 53. Na décima primeira linha do segundo parágrafo, onde se lê: "Ao contrário do método de Horn e Schunck ... é seu elevado custo computacional ", leia-se:

"O inconveniente neste método é seu elevado custo computacional."

Pg. 65. Na última linha do item 1, onde se lê: "... em situações de grande velocidade relativa entre o robô e a cena. ", leia-se:

"A medida que o robô se aproxima ou gira em relação a cena, o que causa maiores deslocamentos de imagem entre quadros quando a distância focal é inadequada."

Pg. 70. No item III.7, onde se lê: "Embora a literatura sobre fluxo óptico ...", leia-se:

"Dada a vasta literatura sobre o fluxo óptico e recuperação do ego-movimento objetivando aplicações em robótica, o presente trabalho baseia-se em, e estende, os reportados em [7], [9] e [14]."

Pg. 75. No último parágrafo, onde se lê: "Convém ressaltar... ", leia-se:

"Convém ressaltar que as estimativas anteriormente citadas são degradadas devido a ruídos."

Tese apresentada à Divisão de Pós-Graduação do Instituto Tecnológico de Aeronáutica como parte dos requisitos para obtenção do título de Mestre em Ciência no Curso de Engenharia

Eletrônica e Computação na Área de Sistemas e Controle

Juliano Cesar Pimentel

Técnicas de Visão Computacional e de Controle para Aplicação em Robótica Móvel

Tese aprovada em sua versão final pelos abaixo assinados

Prof. Dr. Elder Moreira Hemerly

Orientador

Prof. Dr. Homero Santiago Maciel

Chefe da Divisão de Pós-Graduação

Campo Montenegro

São José dos Campos, SP - Brasil

2000

Técnicas de Visão Computacional e de Controle para Aplicação em Robótica Móvel

Juliano Cesar Pimentel

Composição da Banca Examinadora

Prof. Dr. Carlos Henrique Costa Ribeiro	Presidente - ITA
Prof. Dr. Elder Moreira Hemerly	Orientador - ITA
Prof. Dr. Luis Gonzaga Trabasso	ITA
Prof. Dr. Jacques Waldmann	ITA
Prof. Dr. Aluizio Fausto Ribeiro Araújo	USP - São Carlos

ITA

Índice

	Lista de Figuras	v
	Lista de Tabelas	viii
	Lista de Símbolos	ix
	Lista de Siglas	xviii
	Sumário	xix
	Abstract	xxi
	Agradecimentos	xxiii
	Dedicatória	xxiv
I	Introdução	1
II	Guiamento de Robôs Móveis Utilizando Processamento de Imagens em Tempo Real	8
II.1	Estrutura do Algoritmo para Detecção de Guias para Navegação	11
II.2	Estágio (1): Aplicação do Operador Gradiente na Imagem	13
II.3	Estágio (II): Determinação da Primeira Estimativa Paramétrica	15
II.3.1	Transformada de Hough (TH)	17
II.3.2	Critério de Mínimos Quadrados para Interpolação da Parábola	19
II.4	Estágio (III): Recursão para Atualização da Estimativa Paramétrica	24
II.4.1	O Algoritmo de Mínimos Quadrados Recursivos Ponderado com De- caimento Exponencial ($WRLS$)	24
II.4.2	O Algoritmo de Filtro de Kalman na forma de Informação (KIF)	27
II.4.2.1	Atualização da Matriz de Covariância do KIF	29

II.4.3	O Algoritmo de Mínimos Quadrados Normalizado (<i>NLMS</i>)	31
II.5	Resultados Experimentais e Discussões	33
II.6	Conclusões	38
II.7	Transformação de Perspectiva para Guiamento do Robô Móvel	39
II.7.1	Equações da Transformação de Perspectiva considerando os Movimentos da Câmera	42
II.7.1.1	Modelagem da Câmera considerando-se Movimentos de <i>Pan</i> e <i>Tilt</i>	43
II.7.1.2	Modelagem da Câmera considerando-se apenas Movimento de <i>Tilt</i>	46
II.8	Determinação da Postura Relativa do Robô em Relação à Guia de Navegação	47
II.8.1	Determinação do Ângulo de Inclinação	47
II.8.2	Determinação do Desvio Lateral	48
III	Guiamento de Robôs Móveis Utilizando Fluxo óptico	51
III.1	O Fluxo óptico	51
III.2	O Algoritmo de Horn e Schunck Original	54
III.3	Um Algoritmo Alternativo para Estimação do Fluxo óptico	57
III.4	Resultados Experimentais e Discussões	61
III.5	O Uso do Fluxo Óptico para Guiamento do Robô Móvel	66
III.5.1	Equações do Movimento empregando-se Fluxo óptico	67
III.6	Guiamento de Robô Móvel empregando-se o Fluxo Óptico	70
III.7	Recuperação do Ego-Movimento a partir do Fluxo Óptico	70
III.7.1	Recuperação da Rotação e da Profundidade Relativa	70
III.8	Determinação da Postura Relativa para Guiamento do Robô Móvel	71

III.8.1	Determinação do Desvio Lateral e do Ângulo de Inclinação	73
III.9	Conclusões	76
IV	Controle de Robôs Móveis	77
IV.1	Introdução	77
IV.2	Modelo Cinemático de Postura	79
IV.3	Modelo Dinâmico do Robô Móvel Não-Holonômico (<i>RMNH</i>)	84
IV.3.1	Equações de Restrições	84
IV.3.2	Equações Dinâmicas	85
IV.4	Projeto do Controlador	87
IV.4.1	Rastreamento	88
IV.4.2	Seguimento de Caminho	88
IV.4.3	Estabilização em uma Postura	89
IV.5	Controle de <i>RMNH</i> considerando-se o Modelo Cinemático de Postura em Coordenadas Cartesianas	89
IV.5.1	O Controlador de Rastreamento	90
IV.6	Controle de <i>RMNH</i> considerando-se o Modelo Cinemático de Postura em Coordenadas Polares	97
IV.7	Controle de <i>RMNH</i> considerando-se o Modelo Dinâmico	98
IV.8	Controle de <i>RMNH</i> considerando-se o <i>Backstepping Integrator</i>	99
IV.9	Controle de <i>RMNH</i> considerando-se o Modelo Dinâmico em Coordenadas Cartesianas utilizando-se Redes Neurais Artificiais	106
IV.9.1	Capacidade de Aproximação Universal das <i>RNA's</i>	106
IV.9.2	Topologia das <i>RNA's</i>	106
IV.9.3	Estrutura do Controlador para o <i>RMNH</i> via <i>RNA's</i>	109

	IV.10 Comentários sobre a Integração das Técnicas de Visão Computacional	
	e Controle	120
V	Conclusões	122
VI	Referências Bibliográficas	125

Lista de Figuras

II.1	Fluxograma do processamento de imagens em tempo real para detecção de guias de navegação	12
II.2	Determinação do gradiente para imagem em ambiente interno	15
II.3	Determinação do direcional para imagem externa	15
II.4	Modelo para as guias de navegação	16
II.5	Cenário para a determinação da reta via transformada de Hough	19
II.6	Procedimento para geração da primeira estimativa dos coeficientes da parábola para imagem em ambiente interno	21
II.7	Procedimento para geração da primeira estimativa dos coeficientes da parábola para imagem externa	22
II.8	Procedimento para geração da primeira estimativa dos coeficientes da parábola para imagem outra imagem externa	23
II.9	Imagem em ambiente interno em níveis de cinza	36
II.10	Pixels resultantes do operador gradiente direcional para imagem em ambiente interno	36
II.11	Guias para navegação detectadas para imagem em ambiente interno	36
II.12	Imagem externa em níveis de cinza	37
II.13	Pixels resultantes do operador gradiente para imagem externa	37
II.14	Guias para navegação detectadas para imagem externa	37
II.15	Modelo básico para projeção em perspectiva	39
II.16	Configuração dos sistemas de coordenadas para navegação do robô móvel	42

II.17	Configuração para o cálculo do ângulo de inclinação(θ) do robô em relação à guia de navegação	47
II.18	Configuração para o cálculo do desvio lateral (d) do robô em relação à guia de navegação	49
III.1	Procedimento para estimação do fluxo óptico utilizando o algoritmo de Horn e Schunk	57
III.2	Fluxograma do algoritmo de estimação do fluxo óptico	60
III.3	Obtenção do fluxo óptico para cena estática utilizando o algoritmo descrito na Seção 3. A câmera moveu-se para a frente lentamente, aproximando-se da gravura	62
III.4	Obtenção do fluxo óptico para cena estática utilizando o algoritmo descrito na Seção III.3	63
III.5	Obtenção do fluxo óptico para cenas internas utilizando o algoritmo descrito na Seção III.3	64
III.6	Obtenção do fluxo óptico para cenas externas utilizando o algoritmo descrito na Seção III.3	67
III.7	Geometria para análise das equações do movimento empregando fluxo óptico .	67
III.8	Configuração para determinação da postura relativa usando fluxo óptico	72
IV. 1	Sistemas de coordenadas para o robô móvel	80
IV.2	Sistemas de coordenadas considerando referência de postura diferente de zero .	83
IV.3	Estrutura do sistema de controle para rastreamento ,	90
IV.4	Trajetória desejada e trajetória obtida para o robô usando o controlador de ras - treamento apresentado em (IV.32)	93
IV.5	Sinais de controle para as velocidades linear e angular, obtidas via (IV-32)	93

IV.6	Erros de postura, obtidos conforme (IV.31)	94
IV.7	Trajatória desejada e trajetória obtida para o robô, segundo o controlador de rastreamento apresentado em (IV.33), para trajeto tipo circuito fechado	95
IV.8	Sinais de controle para as velocidades linear e angular, obtidas via (IV.32) para trajeto tipo circuito fechado	96
IV.9	Erros de postura obtidos via (IV.32), para trajeto tipo circuito fechado	96
IV.10	Estrutura do sistema de controle tipo <i>backstepping integrator</i>	101
IV.11	Trajetórias desejada e obtida para o robô, empregando-se o controlador tipo <i>backstepping integrator</i>	102
IV.12	Sinais de controle para as velocidades linear e angular, obtidas para o controlador de <i>backstepping integrator</i>	103
IV.13	Erros de postura, obtidos como controlador tipo <i>backstepping integrator</i>	103
IV.14	Torques aplicados nas rodas direita e esquerda, obtidos via (IV.39)	104
IV.15	Evolução das posturas desejada e obtida durante o movimento do robô	104
IV.16	<i>RNA feedforward</i> com uma camada escondida	107
IV.17	Estrutura do sistema de controle utilizando <i>RNA</i>	110
IV.18	Trajetórias desejada e obtida para o robô, empregando-se o controlador utilizando redes neurais	116
IV.19	Sinais de controle para as velocidades linear e angular, obtidos via (IV.32) ...	116
IV.20	Sinais de controle dinâmico considerando os torques atuantes sobre o robô ...	117
IV.21	Sinais de erro resultantes da lei de controle aplicada ao robô	117
IV.22	Evolução das posturas desejadas e obtidas, durante o movimento do robô	118
IV.23	Evolução de alguns dos pesos da <i>RNA</i> durante o treinamento <i>on-line</i>	118

Lista de Tabelas

II.1	Comparação do tempo de processamento entre os métodos recursivos	34
IV.1	Parâmetros físicos do robô <i>Magellan</i> TM	92

Lista de Símbolos

Os símbolos utilizados nesta tese são separados conforme sua ocorrência nos capítulos:

Capítulo II

a	ganho escalar do algoritmo de mínimos quadrado normalizado;
a_i	coeficientes das parábolas que modelam as guias;
\hat{a}_i	estimativa dos coeficientes das parábolas que modelam as guias;
α	ângulo de inclinação vertical, ou <i>tilt</i> ;
c	constante para evitar divisão por zero no algoritmo <i>NLMS</i> ;
d	desvio lateral do robô em relação à parábola;
δ	orientação do robô em relação à guia de navegação;
f	distância focal da lente;
F_t	é a matriz que relaciona $\hat{\theta}_{t+1}$ com $\hat{\theta}_t$ na ausência de uma função forçante;
ϕ_t	matriz de regressão, de dimensão $N \times 3$, contendo termos de y_1 , cujas linhas são dadas por $(1 \quad y_{t,N} \quad y_{t,N}^2)$;
φ	ângulo da coordenada polar que define a reta no método de Hough;
$G(\cdot)$	gradiente da função de intensidade da imagem;
G_t	é a matriz que pondera o efeito de co_o ;
$G_x(\cdot)$	componente x do gradiente da função de intensidade da imagem;
$G_y(\cdot)$	componente y do gradiente da função de intensidade da imagem;

H	altura do centro óptico da lente em relação à superfície de navegação;
γ	ângulo de inclinação horizontal, ou <i>pan</i> ;
i	vetor representando um ponto p na imagem;
i_h	vetor representando um ponto p na imagem na forma homogênea;
k	constante não-nula para conversão de coordenadas no sistema homogêneo;
K_t	matriz de ganho para algoritmo de recursão, de dimensão 3 x N ;
λ	fator de decaimento no <i>WRLS</i> ;
m	inclinação da reta tangente à parábola no ponto P;
M	matriz de ordem $N \times 3$, onde N é o número de pixels da parábola, contendo termos de y_i cujas linhas são dadas por $(y_{i,N}^2 \ y_{i,N} \ 1)$;
o	origem do sistema de coordenadas (x y z);
O	origem do sistema de coordenadas (X Y Z);
P	matriz de transformação que relaciona ω_h e i_h ;
$p(x_p, y_p)$	ponto no sistema de coordenadas cartesianas da imagem;
$P(X_p, Y_p, Z_p)$	ponto no sistema de coordenadas cartesianas do mundo;
P_t	matriz de <u>atualização</u> da covariância de estado para algoritmo de recursão, de dimensão 3x3 ;
Q_t	covariância de ω_t ;
$\hat{\theta}_t$	vetor de estimativa paramétrica dos coeficientes da parábola no instante t, de dimensão 3;
r	distância do ponto p à origem do sistema de coordenadas da imagem;

R	distância do ponto P à origem do sistema de coordenadas do mundo;
R_t	covariância de V_t ;
R_α	matriz de transformação de rotação vertical entre dois sistemas de coordenadas;
R_γ	matriz de transformação de rotação horizontal entre dois sistemas de coordenadas;
ρ	raio da coordenada polar que define a reta no método de Hough;
$S_N(\theta)$	função custo para o <i>WRLS</i> ;
v_t	ruído de medida, com variância R , ;
ω	vetor representando um ponto P no espaço;
ω_h	vetor representando um ponto P no espaço na forma homogênea;
ω_t	ruído de estado, com variância Q ,;
X	vetor de dimensão N , formado pelas coordenadas x_i , onde N é o número de pixels da parábola;
x_i	coordenada horizontal no espaço bidimensional;
$x(y_i)$	coordenada horizontal do ponto (x, y) localizado sobre a parábola, cuja coordenada vertical é y_i ;
$(x \ y \ z)$	sistema de coordenadas cartesianas da imagem para transformação de perspectiva;
$(X \ Y \ Z)$	sistema de coordenadas cartesianas do mundo para transformação de perspectiva;
y_i	coordenada vertical no espaço bidimensional;
$ $	valor absoluto;

Capítulo III

α	fator de ponderação no algoritmo de Horn e Schunck;
$ArgMin(\cdot)$	argumento mínimo de uma função;
d	distância lateral do robô à parede;
$E(x, y, t)$	padrão de brilho, no instante t, do ponto (x, y) da imagem;
E_x	derivada do padrão de brilho com relação a x;
E_y	derivada do padrão de brilho com relação a y;
E_t	derivada do padrão de brilho com relação a t;
f	distância focal da lente;
ϕ	ângulo de inclinação do robô móvel em relação à parede;
Γ_x	componente rotacional do fluxo óptico;
H	variação temporal do fluxo óptico em uma região fixa da imagem;
H_x	componente x da variação temporal;
H_y	componente y da variação temporal;
\vec{N}	vetor normal à superfície da parede;
$(N_x \ N_y \ N_z)$	componentes do vetor normal à superfície da parede;
$\nabla^2(\cdot)$	laplaciano de uma função;
O	origem do sistema de coordenadas (X Y Z);
$O(\delta t, \delta x, \delta y)$	termos de segunda ordem e ordens mais elevadas na série de Taylor;
(P_x, P_y, P_z)	ponto no sistema de coordenadas do mundo;
\tilde{r}_x	valores medidos (com ruído) do componente x do fluxo óptico;

\tilde{r}_y	valores medidos (com ruído) do componente y do fluxo óptico;
(r_x, r_y, f)	ponto no sistema de coordenadas do plano da imagem;
\vec{T}	vetor de velocidade translacional;
$(T_x \ T_y \ T_z)$	componentes da velocidade de translação;
τ	profundidade relativa;
Y_x	componente translacional do fluxo óptico;
u	fluxo óptico da imagem na direção x ;
v	fluxo óptico da imagem na direção y ;
$(\omega_x \ \omega_y \ \omega_z)$	componentes da velocidade de rotação;
$\tilde{\omega}_y$	rotação estimada do robô móvel;
$\vec{\Omega}$	vetor de velocidade angular;
$(X \ Y \ Z)$	sistema de coordenadas cartesianas do mundo;

Capítulo IV

$A(q)$	matriz das restrições de movimento do robô móvel de dimensão $m \times n$;
α	diferença entre o ângulo de orientação do robo e o erro de posição;
C_i 's	constantes positivas conhecidas;
d	distância entre P_o e P_c ;
e	valor absoluto do erro de posição em coordenadas polares;
e_c	erro da velocidade de rastreamento, dado por $\Gamma v_c - v$;
$(e_x \ e_y \ e_\theta)$	erro de posição no sistema de coordenadas $(X_c \ Y_c \ P_c)$, para x, y e θ ;

e_p	erro de postura entre a referência p_r e a postura atual do robô p , com a devida transformação de coordenadas, T_e ;
$E(q)$	matriz de transformação das entradas, de dimensão 5×2 ;
ϵ	erro de reconstrução da <i>RNA</i> ;
ϵ_{\max}	limitante do erro de reconstrução da <i>RNA</i> ;
ϵ_N	limitante positivo do erro de reconstrução da <i>RNA</i> ;
$f(x)$	$\overline{M}\Gamma\dot{v}_c + \overline{V}v$;
$\hat{f}(x)$	estimativa da função de saída da <i>RNA</i> ;
$\tilde{f}(x)$	erro de estimação de $f(x)$, onde $\tilde{f}(x) = f(x) - \hat{f}(x)$;
$F(\dot{q})$	matriz de atrito de contato com a superfície;
γ	termo de robustecimento no sinal de controle;
γ_1	ganho 1 para o controlador cinemático em coordenadas polares;
γ_2	ganho 2 para o controlador cinemático em coordenadas polares;
Γ	matriz que relaciona as velocidades de deslocamento angular das rodas com as velocidades linear e angular do robô;
h	ganho 3 para o controlador cinemático em coordenadas polares;
I	$I_c + 2m_w R^2 + 2I_m$ - momento de inércia total do robô;
I_c	momento de inércia do robô sem as rodas e os motores em relação a um eixo vertical através de P_o ;
I_m	momento de inércia de cada conjunto roda e motor em relação ao eixo vertical através do diâmetro;
λ_i	multiplicadores de Lagrange, onde $i = 1, 2 \text{ e } 3$;

K	vetor de ganho para o controlador cinemático em coordenadas cartesianas, onde $K = (k_1 \quad k_2 \quad k_3)$;
K_4	matriz diagonal positiva definida, dada por $K_4 = k_4 I$;
K_z	ganho do termo robusto usado no controlador dinâmico;
m	$m_c + 2m_w$ - massa total do robô;
m_c	massa do robô sem as rodas e os motores;
m_w	massa de cada conjunto roda e motor;
$M(q)$	matriz de inércias, de dimensão 5 x 5 e simétrica positiva definida;
\bar{M}	$S^T M S$;
MC	matriz do modelo cinemático, de dimensão 2 x 2 ;
μ	soma de $F(\dot{q})$ e τ_d ;
$N(\text{média, variância})$	distribuição normal;
N_h	número de neurônios na camada escondida;
$O(\tilde{V}^T x)$	termos de segunda ordem e ordens mais elevadas na série de Taylor;
p	postura atual, dada por $p = (x_c \quad y_c \quad \theta_c)$;
p_r	postura de referência, dada por $p_r = (x_r \quad y_r \quad \theta_r)$;
P_c	centro de massa do robô móvel;
P_o	interseção do eixo de simetria com o eixo das rodas;
q	vetor de coordenadas generalizadas, sendo $q = (x_c \quad y_c \quad \theta \quad \varphi_r \quad \varphi_l)$;
r	raio de cada roda;
$\rho(t)$	somatório de $e(x)$, dos distúrbios e dinâmica não-modelada $\bar{\mu}$ e dos termos de ordens elevadas na expansão em série de Taylor de $f(x)$;

$S(q)$	matriz auxiliar para representar a equação do movimento, na forma de espaço de estados, de dimensão 5×2 ;
$\sigma(\cdot)$	função de ativação sigmoideal;
$\tilde{\sigma}$	erro de saída da camada escondida para uma dada entrada x , definido por $\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x)$;
T	função de transferência que relaciona a velocidade de cada roda e a tensão aplicada no motor correspondente;
T_e	matriz de rotação entre o sistema de coordenadas inercial e o sistema de coordenadas $(X_c \ Y_c \ P_c)$;
τ	lei de controle para cálculo do torque a ser aplicado nos motores;
τ_i	torque no eixo da roda gerado pelo motor;
τ_d	perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada;
$\tau_{d,\max}$	limitante dos distúrbios dinâmicos, r_d ;
θ	ângulo de orientação do robô;
W	matriz de pesos da camada de saída da rede neural, de dimensão $N_h \times m$;
\hat{W}	estimativa da matriz de pesos da camada de saída da RNA;
\tilde{W}	$W - \hat{W}$;
ω	velocidade angular;
v	vetor de velocidades de deslocamento angular das rodas, $v = (\dot{\phi}_r \ \dot{\phi}_l)^T$;
v	velocidade linear;

\mathbf{v}_c	vetor de velocidades de controle, $\mathbf{v}_c = (v_c \quad \omega_c)^T$;
V	matriz de pesos da camada de entrada da rede neural, de dimensão $m \times N_h$;
$V(.,.)$	candidata a função de Lyapunov;
$V(q)$	vetor dependente da posição e da velocidade, de dimensão 5;
\bar{V}	$S^T M \dot{S} + S^T V S$;
\tilde{V}	$\tilde{V} = V - \hat{V}$;
\hat{V}	estimativa da matriz de pesos da camada de entrada da RNA;
φ_i	ângulo de rotação da roda com relação ao seu próprio eixo, para a i-ésima roda;
x	vetor de entrada da rede neural, de dimensão n::
x_c	coordenada horizontal do centro de massa do robô móvel;
y_c	coordenada vertical do centro de massa do robô móvel;
\hat{z}	estimativa do argumento da função de ativação sigmoidal, onde $\hat{z} = \hat{V}^T x$;
\tilde{Z}	$\tilde{Z} = Z - \hat{Z}$ é o erro de estimação de pesos Z, onde $Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}$;
Z_m	limitante positivo de Z, tal que $\ Z\ _F \leq Z_m$;
$2R$	distância entre as rodas;
$\ \cdot\ $	norma dois de qualquer vetor ou matriz;
$\ \cdot\ _F$	norma de Frobenius: dada uma matriz $A = (a_{ij})$, $\ A\ _F^2 = tr(A^T A) = \sum_{i,j} a_{ij}^2 .$

Lista de Siglas

As siglas utilizadas nesta tese são:

<i>CCD</i>	<i>Charge Coupled Device;</i>
FOE	Foco de Expansão;
<i>KIF</i>	<i>Kalman Information Filter;</i>
MQ	Método dos Mínimos Quadrados;
NLMS	<i>Normalized Least Mean Squares;</i>
RMNH	Robô Móvel Não-Holonômico;
RNA	Rede Neural Artificial;
TH	Transformada de Hough;
WRLS	<i>Weighted Recursive Least Squares.</i>

Sumário

Nesta tese são investigados dois subproblemas básicos envolvidos na aplicação de robôs móveis: guiamento e controle. No caso do guiamento, ênfase é dada aos sistemas baseados em visão, e são abordados dois procedimentos distintos: a) um baseado na detecção de guias, e b) outro baseado na técnica de fluxo óptico, que relaciona as velocidades aparentes entre cena e observador.

No procedimento baseado em detecção de guias proposto nesta tese, as guias são modeladas por um polinômio de segunda ordem, cujos parâmetros são atualizados por um algoritmo recursivo, o filtro de Kalman na forma de informação (*KIF*). Tal procedimento requer menor intervenção humana que aquela exigida em [41], e necessita menos informação inicial para extração de características e detecção de guias. Testes com imagens reais, tanto internas quanto externas, e em presença de situações desfavoráveis como baixa luminosidade, indicam que o procedimento exhibe bom desempenho e robustez em condições de testes similares a [41].

Um procedimento alternativo para guiamento baseado em fluxo óptico é também implementado objetivando comparar custo computacional, hipóteses necessárias e restrições de uso. Os resultados apresentados para ambos os procedimentos não consideram a interação com a malha de controle e guiamento do robô.

No que se refere ao controle, são investigadas leis visando a implementação, ou seja, leis que gerem sinais suaves e respeitem os limites de *hardware* do robô. Utilizam-se estratégias de controle empregando redes neurais artificiais (*RNA*'s) para o modelo dinâmico completo do robô móvel. As leis de controle são sintonizadas para possibilitar uma partida suave, em trajetórias retilíneas ou do tipo circuito fechado, desenvolvendo um movimento uniformemente acelerado a partir do repouso até atingir sua velocidade

nominal. Tem-se assim uma contribuição em relação a [11] e [44], onde não há qualquer preocupação em relação aos aspectos práticos de implementação. Resultados de simulação utilizando os parâmetros de um robô real, *MagellanTM*, sugerem o bom desempenho das estratégias investigadas, nas mesmas condições de testes de [11] e [44].

Abstract

This thesis addresses two important issues related to mobile robotics: navigation and control. Concerning the navigation, systems based on vision are considered and two procedures are investigated: a) the first is based on lanemarks detection, and b) the second relies on the optical flow technique, which relates the relative movements between scene and observer.

In the lanemarks detection procedure, proposed in this thesis, the lanemarks are modelled by a second order polynomial, whose coefficients are updated by a recursive algorithm, the Kalman information filter (*KIF*). Such procedure requires less human intervention than that in [41], and less initial information for feature extraction and lanemarks detection. Experiments with real images, both indoor and outdoor, even in the presence of non-ideal conditions as low luminosity, indicate that this procedure presents good performance and robustness, under test conditions similar to [41].

An alternative procedure based on optical flow is also implemented, in order to compare computational cost, underlying hypothesis and applications constraints. The results presented here for both procedures do not consider the integration with the robot guidance and control loop.

Regarding the control problem, laws are investigated aiming at implementation. More precisely, laws that generate smooth signals and do not violate the hardware limits of the robot. A control strategy is devised by applying artificial neural networks to the complete dynamic model of the mobile robot. For such model, the control laws are tuned to allow a smooth start up and to accelerate to the nominal speed, both in straight trajectories and closed ones. This represents a contribution with regard to [11] and [144], where no attention is paid to practical implementation. Simulation results employing the

*Magellan*TM parameters show the good performance of the investigated strategies, under test conditions similar to [11] and (44).

Agradecimentos

- a Deus, e seu filho Jesus, por sempre me conduzirem por um caminho sobremodo excelente;
- ao Prof. Dr. Elder Moreira Hemerly, pela orientação precisa e por ter apresentado temas de relevância e atuais para a pesquisa;
- ao Prof. Dr. Jacques Waldmann pelas consultas em temas relacionados a Visão Computacional;
- aos colegas do ITA, em especial aos de Pós-Graduação em Sistemas e Controle pela convivência e companheirismo; e
- a CAPES pela concessão da bolsa de mestrado que possibilitou a realização desta tese, e à FAPESP pelo apoio financeiro via processo n° 99/05772-2.

A Jesus, que é digno de toda glória,
A minha mãe, Diva (*in memorian*), a quem devo o que sou,
A minha futura esposa, Kelly Alves, e
A minha família.

Capítulo I

Introdução

Robôs móveis têm sido utilizados em uma ampla gama de aplicações, tais como em tarefas de transporte em fábricas, em aplicações hospitalares, domésticas e até de vigilância [2], [42]. Um grande esforço de pesquisa tem sido dedicado na última década para se permitir que robôs móveis tenham a capacidade de se mover de forma autônoma em ambientes pouco estruturados e sem a necessidade de supervisão.

Há dois temas básicos envolvidos na aplicação de robôs móveis autônomos: a) a determinação da postura (posição e orientação do veículo) em um dado instante de tempo, e b) o controle, que se preocupa com o rastreamento de uma trajetória, ou com a regulação em torno de uma certa postura. Estes dois temas são tratados nesta tese. No caso da determinação da postura, considera-se os sistemas baseados em visão. A informação visual eleva a autonomia e possibilita guiamento em ambientes não-estruturados. No âmbito do controle, são abordadas técnicas que tratam desde o modelo cinemático, que considera apenas as velocidades de movimento do robô, até as que consideram o modelo dinâmico completo, incluindo perturbações e atritos. Estas técnicas podem fazer uso das informações de postura oriundas dos sensores de visão ou da integração destes sensores com outros, como *o dead-reckoning, bússola, GPS ou outros*.

Um sistema usando processamento de imagens em tempo real para navegação autônoma de veículos é apresentado em [38]. Além do subsistema de visão computacional, utiliza-se outro subsistema baseado em *dead-reckoning*, que conta as rotações das rodas direita e esquerda do robô. Como a acurácia deste método depende diretamente de quão confiáveis sejam os sensores, a integração dos dois subsistemas é feita através do filtro de Kalman estendido, que reduz os erros absolutos e relativos causados por cada subsistema. Os resultados obtidos mostram que se obteve uma estimação acurada da posição e da direção do veículo.

Em [19] investiga-se o desempenho de estratégias para guiamento de robôs móveis que incorporam informações sobre o ambiente por intermédio de processamento digital de imagens em tempo real. O procedimento utilizado basicamente baseia-se na determinação do desvio lateral e da inclinação do robô em relação a uma faixa no solo. São apresentados resultados obtidos sem a realimentação visual e utilizando-se a realimentação visual. No primeiro caso, utiliza-se apenas sensores de posição acoplados nas rodas do veículo. Para o segundo caso, além da informação dos sensores, utiliza-se a realimentação visual que possibilita o alinhamento do veículo com a trajetória desejada.

Uma aplicação em ambiente interno é realizada em [8], com guiamento utilizando informações visuais. A navegação correspondente requer o aprendizado de um modelo do ambiente. Para tanto, o veículo é guiado manualmente através de um ambiente e as imagens e dados para a situação de controle são gravados automaticamente. Um procedimento *of line* é então executado para criar um modelo inicial do ambiente, e no estágio de navegação este modelo vai sendo atualizado a cada novo percurso pelo circuito. Linhas verticais são definidas no percurso e são tratadas pelo algoritmo de aprendizado como pontos-padrão. Na extração de características e identificação das linhas verticais é utilizada uma modificação da transformada generalizada de Hough.

Em [41] propõe-se um método para detecção das guias de navegação para o veículo, através da compensação para dois importantes tipos de incerteza na detecção de guias: mudanças na direção das guias e incertezas nas medidas devido a cada nova imagem. Há, no entanto, a necessidade de modelos iniciais aproximados, antes que o rastreamento seja efetivamente inicializado. A correspondência entre o modelo e as guias é estabelecida manualmente por um agente humano (teleoperador), que posiciona os modelos para alinhá-los com a aparência das guias na imagem. Já em [23], apresenta-se um procedimento baseado em modelos para reconhecimento de formas em vias horizontais, não se detendo apenas na presença das guias, mas principalmente nas formas das vias. Entretanto, utiliza-se processamento estático das imagens para que haja classificação dos trechos da via em quatro modelos básicos, *straight*, *curve*, *curve-in*, *curve-out*. Devido ao processamento de imagens estáticas, requer-se o uso de câmeras de distância focal adequada, isto é, que permitam detectar a forma da estrada na faixa de 20m a 100m (ou mais, dependendo da velocidade do veículo).

Em [9] apresenta-se a tarefa de guiar o robô móvel pelo centro de um corredor, baseado exclusivamente em uma sequência de imagens advindas de uma câmera *onboard*. As medidas necessárias para o guiamento, a distância à parede e a orientação do robô, são obtidos diretamente do fluxo óptico. Em [14] considera-se a navegação em estradas também utilizando fluxo óptico. Cenas externas são obtidas e processadas, e resultados são reportados no sentido de auxiliar o motorista na tomada de decisão quanto à presença de obstáculos e ações de guiamento.

No que tange ao controle de robôs móveis, em [1] apresenta-se uma lei de controle em malha fechada para o guiamento e rastreamento de uma trajetória pré-definida. Com o auxílio de uma candidata a função de Lyapunov e utilizando apenas o modelo cinemático para o robô, é proposta uma lei de controle suave e efetiva em coordenadas polares. Resultados de

simulação são reportados e a estabilidade é garantida através do segundo método de f yapunov. Já em [10] e [11], considera-se o modelo dinâmico além do cinemático. O algoritmo de controle é aplicado para as mesmas situações que em [11], com o adicional de se considerar o modelo mais completo do robô móvel, levando-se em conta perturbações desconhecidas e a dinâmica não modelada do tipo não estruturada. Isto é feito através de uma rede neural que é treinada *on-line*, que garante pequenos erros de rastreamento e sinais de controle limitados.

Nesta tese objetiva-se investigar técnicas de visão computacional e de controle para aplicação em robótica móvel. Como principais contribuições podem ser citadas:

- proposição de um algoritmo para detecção de guias visando o guiamento de robôs móveis que, ao contrário de [41], requer menor intervenção humana e necessita menos informação inicial para a extração de características e detecção de guias. Um estágio importante deste algoritmo diz respeito à atualização da estimativa dos coeficientes das parábolas que modelam os percursos a serem seguidos. Nesta tese é proposto um método recursivo baseado no filtro de Kalman na forma de informação (VF). Este método tem seu desempenho comparado com outros métodos recursivos: o algoritmo de mínimos quadrados ponderado com decaimento exponencial (WRVS), proposto em [41], e o algoritmo de mínimos quadrados normalizado (NLMS), também investigado nesta tese. Para melhorar o desempenho numérico dos métodos baseados no *KIF* e do *WRLS* estabelecem-se formas alternativas para a atualização das matrizes de covariância. Testes sem se considerar o problema do controle, com imagens reais, tanto internas quanto externas, e em presença de situações desfavoráveis como baixa luminosidade, sombras e presença de objetos, indicam que o algoritmo proposto para detecção de guias, que apresenta bom desempenho e robustez, em condições de testes similares a [41].

- Investigação e o uso de um método qualitativo para a supervisão da tarefa de guiamento, baseado na estimação do fluxo óptico, que segundo estudos preliminares [7], [9] e [14], foi indicado com potencialidade de bom desempenho. Através de exaustiva pesquisa bibliográfica, selecionou-se um algoritmo robusto e eficiente [46] que, ao contrário de métodos clássicos de estimação de fluxo óptico, como o algoritmo de Horn e Schunck [20], [21] e [22], lida bem com ruídos na imagem e apresenta boas estimativas nas condições investigadas. Resultados são apresentados e discutidos, bem como limitações e deficiências deste método. Apresenta-se também um método para guiamento de robôs móveis baseado no emprego do fluxo óptico.
- Com respeito ao controle, nesta tese, são consideradas estratégias como as propostas em [11] e [44], baseadas em redes neurais, visto que possibilitam tratar o modelo completo do robô móvel, incertezas paramétricas e perturbações não modeladas. A superioridade destas técnicas em relação às usuais já foi claramente explicitada em [11] e [44] e portanto não será repetida neste trabalho. Por outro lado, em [11] e [44] não há preocupação com os aspectos práticos de implementação. Assim, nesta tese, ênfase é dada à implementação, ou seja, requerendo-se que as leis de controle gerem sinais suaves e que respeitem os limites de *hardware* do robô. Em [11] há a preocupação com a estrutura do controle, de forma a se integrar um controlador cinemático e uma rede neural para se obter uma lei de controle com estabilidade garantida. Em [44] também estuda-se a estabilidade para o controlador utilizando redes neurais e controle cinemático, incluindo a análise da situação na qual o ponto de guiamento não esteja localizado no ponto médio do eixo das rodas. Nesta tese, as leis de controle foram sintonizadas, empregando-se parâmetros do *MagellanTM* [35] da *IS Robotics*, para possibilitar, em trajetórias retilíneas ou do tipo circuito fechado, uma partida suave do robô, desenvolvendo um movimento uniformemente acelerado a partir do repouso até atingir sua velocidade nominal. É

utilizado o modelo dinâmico completo do robô móvel, incluindo-se atrito de contato com a superfície e perturbações não modeladas que incluem a dinâmica não modelada do tipo não estruturada, de modo a se elevar o realismo das simulações. Resultados de simulação sugerem o bom desempenho das estratégias investigadas.

Esta tese está estruturada da seguinte forma: no Capítulo II, propõe-se o algoritmo para detecção de guias de navegação, onde é introduzida sua estrutura, o método de recursão, baseado no *KIF*, além dos métodos baseados em *WRSS* e *NLMS*, com contribuições na forma de cálculo do *KIF* e do *WRLS*. São apresentados os resultados experimentais e discussões; e finalmente, um método para guiamento do robô, utilizando-se transformação de perspectiva, através da informação visual oriunda do algoritmo de processamento de imagens em tempo real. No Capítulo III, apresenta-se um resumo sobre fluxo óptico e o algoritmo de Horn e Schunck, e investiga-se um algoritmo alternativo mais robusto [46] e que apresenta boas estimativas para o fluxo para os casos considerados, sendo apresentados os resultados e discussões. É também proposto um método para guiamento do robô móvel através de um ambiente semi-estruturado utilizando-se unicamente as informações oriundas do fluxo óptico. No Capítulo IV, é tratado o problema de controle, sendo considerados os modelos de postura, onde apenas as velocidades das rodas e sua orientação são importantes, e modelos de configuração, onde outras variáveis são importantes, como o deslocamento angular das rodas do robô. Resultados para simulação com parâmetros do robô *MagellanTM* [35] são apresentados e discutidos.

Convém ressaltar que a integração das estratégias de guiamento e controle não foi efetuada neste trabalho. Esta integração pode ser feita em dois níveis: a) a nível de simulação, com modelagem do sistema de visão e do robô móvel, seguida de uma análise estatística de desempenho, objetivando avaliar a influência dos diversos parâmetros de projeto, tanto do sistema de visão quanto do de controle; b) implementação em tempo real, com integração dos

subsistemas *de hardware e software*. Na seção IV.10 são apresentados mais comentários sobre a integração das técnicas de visão computacional e controle.

Finalmente, no Capítulo V, são apresentadas as conclusões desta tese.

Capítulo II

Guiamento de Robôs Móveis Utilizando Processamento de Imagens em Tempo Real

Algoritmos para extração de características e detecção de guias para navegação são necessários para manter o veículo no trajeto desejado sem desvio, informar sobre a presença de obstáculos e outros, além de mapear as informações do "mundo" em 3D para 2D, o que é de interesse para fins de controle de posição e velocidade do veículo. Dificuldades surgem entretanto no processamento de imagens provindas de ambientes não estruturados, devido à complexidade de uma modelagem eficaz das guias, que podem ser retas, curvas, contínuas ou tracejadas, estar sempre presentes ou eventualmente ausentes. Para estes casos são necessários algoritmos robustos que possam lidar com incertezas nas medidas.

Neste capítulo, propõe-se um algoritmo para detecção de guias de navegação para robôs móveis através do processamento de imagens em tempo real, que apresenta bons resultados para aplicações internas e externas, além de ser robusto e rápido o suficiente para lidar com a dinâmica de robôs móveis.

Muitos métodos têm sido propostos para detecção de características de interesse para navegação de robôs móveis e veículos autoguiados. Por exemplo, em [31], um método para detecção de guias utiliza um conjunto de características no domínio da frequência, calculado

particionando-se a imagem em blocos de pixels de dimensão 8x8. Estas características refletem a quantidade de energia de borda diagonalmente dominante que é contida em cada bloco, sendo combinadas via método bayesiano para se detectar as guias de interesse. Em [28], apresenta-se a necessidade de um processamento *off-line* para extração de características e detecção de bordas, além da presença humana para guiar o veículo por um corredor. O sistema trabalha em duas fases: na primeira, o veículo é guiado por alguém através de um corredor. Durante este movimento, o sistema de visão extrai as características indesejadas, tais como portas ou escadas. A categoria e localização destas características são gravadas em um mapa. Na segunda fase, o sistema examina o mapa e seleciona uma sequência de guias que orientem o veículo ao destino no menor tempo.

Neste capítulo, utilizam-se parábolas como modelo geométrico de aproximação para cada guia [30], [41], e aplica-se um método recursivo, como o filtro de Kalman na forma de informação (*KIF*) para atualizar a estimativa dos coeficientes destas parábolas, obtendo-se assim um procedimento robusto e eficiente, bastante veloz na aquisição e processamento de várias imagens por segundo.

Para o procedimento proposto nesta tese, ao contrário de [41], não há intervenção humana em nenhuma fase do processo, e o algoritmo realiza processamento de imagens *on-line*, sem a necessidade de modelos previamente estabelecidos. Obtiveram-se bons resultados em presença de situações desfavoráveis, tais como baixa luminosidade no ambiente, guias pouco visíveis, presença de objetos na região de navegação, tais como mesas e extintores nas aplicações em ambientes internos e sombras, rachaduras no pavimento e carros nas aplicações externas. Os pontos agrupados para a detecção das guias vêm diretamente do operador gradiente aplicado à imagem em *níveis de cinza*, sem a necessidade de comparação com nenhum modelo.

Este capítulo está estruturado da seguinte forma: Na Seção II.1 apresenta-se a estrutura do algoritmo proposto para detecção de guias para navegação, e nas Seções II.2, II.3 e II.4 os estágios de processamento. Na Seção II.4 apresentam-se os algoritmos de recursão testados e comparados no processamento de imagens em tempo real: a) o algoritmo de mínimos quadrados ponderado com decaimento exponencial (*WRLS*), b) o algoritmo de Kalman na forma de informação (*KIF*), e c) o algoritmo de mínimos quadrados normalizado (*NLMS*). O *KIF*, proposto nesta tese, é o algoritmo de recursão utilizado na obtenção dos resultados experimentais. Tanto para o *WRLS* quanto para o *KIF*, estabelecem-se expressões para atualização da matriz de informação que asseguram a propriedade positiva definida, já que nas suas formas originais não se verificava tal propriedade. Na Seção II.5, são apresentados os resultados experimentais e discussões para a detecção de guias de navegação utilizando o algoritmo de processamento de imagens em tempo real, e na Seção II.6 algumas conclusões sobre a utilização e aplicação deste procedimento. Na Seção II.7, trata-se da transformação de perspectiva e seu uso para guiamento do robô móvel. Finalmente, na Seção II.8, apresenta-se um método de guiamento de robôs móveis, através de transformação de perspectiva, utilizando-se as informações oriundas do algoritmo de processamento de imagens em tempo real.

II.1 Estrutura do Algoritmo para Detecção de Guias para Navegação

O algoritmo identifica as guias através da atualização das estimativas dos coeficientes da parábola obtida da imagem processada. O fluxograma com a estrutura e ordenação do procedimento utilizado consta na Figura II.1, e há três estágios principais para obtenção do resultado, que é repetido para cada nova imagem adquirida:

- I. Operador Gradiente da Imagem - remove da imagem, em níveis de cinza, a maioria das características indesejadas.
- II. Aplicação da Transformada de Hough [13],[24] e Interpolação por Mínimos Quadrados [6], [15], [30] - possibilita a determinação da primeira estimativa para os coeficientes da parábola, e em casos de poucos pontos coletados, para as estimativas posteriores.
- III. *Recursão* - *WRLS* [15], [16] ou *KIF* [3], [18], [26] para atualizar recursivamente a estimativa dos coeficientes das parábolas.

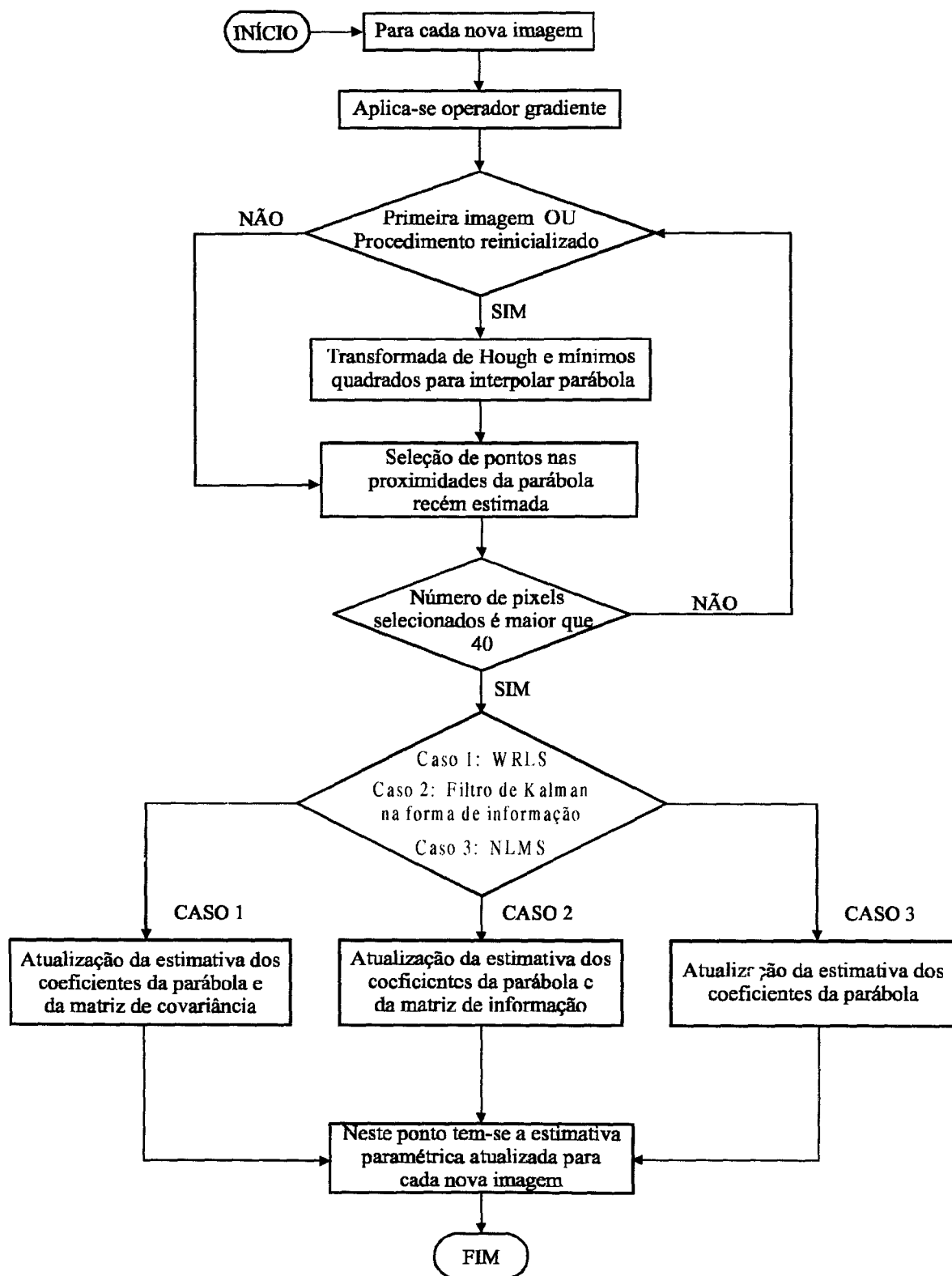


Figura II.1. Fluxograma do processamento de imagens em tempo real para detecção de guias para navegação.

II.2 Estágio (I): Aplicação do Operador Gradiente na Imagem

No estágio (I), a imagem processada é aquela em níveis de cinza (*gray level* - 256 níveis de resolução). O operador gradiente remove da imagem as características indesejadas, tais como linhas próximas da horizontal e aquelas próximas da vertical em função de que, na imagem real, as guias nunca estarão nesta situação. O limiar do gradiente pode ser alterado dependendo do ambiente, por exemplo, imagens em ambiente interno permitem o uso de um limiar menor, ao passo que imagens externas, dado o grande contraste entre luz, pavimento e objetos requerem um limiar maior. Há ainda a possibilidade de se utilizar limiar adaptativo, o que porém eleva o tempo de processamento.

A função que implementa o cálculo do operador gradiente usa a área da imagem definida como de interesse, chamada de janela de processamento, isto é, eliminando-se limites superior, inferior, direito e esquerdo, cada um destes predefinidos. Nas aplicações em ambientes internos e externos usualmente aplicou-se o valor 20 para a janela de processamento, significando que o algoritmo despreza os primeiros 20 pixels na vertical descendente, na vertical ascendente e nas horizontais, da direita para a esquerda e vice-versa.

Mudanças significativas nos valores de níveis de cinza de uma imagem podem ser detectados usando uma aproximação discreta para o gradiente. A magnitude do gradiente, considerando-se esta aproximação, em valores absolutos, pode ser aproximada por

$$G[f(x, y)] \approx |G_x| + |G_y| \quad (II.1)$$

onde $G[f(x, y)]$ é o gradiente de $f(x, y)$ que por sua vez é a função de intensidade de brilho da imagem. G_x e G_y são os componentes do gradiente nas direções x e y, respectivamente. A aproximação numérica aqui utilizada para o cálculo do gradiente é a forma mais simples, conforme [24],

$$G_x \approx f[i, j+1] - f[i, j] \quad (\text{II.1 a})$$

e

$$G_y \approx f[i, j] - f[i+1, j] \quad (\text{II.1 b})$$

As máscaras correspondentes para os operadores também são as mais simples: 2x1 para o cálculo de G_x e 1x2 para o cálculo de G_y . Convém ressaltar que diversos outros operadores poderiam ser utilizados, vide [24], contudo, o desempenho do operador gradiente não é crítico neste trabalho, uma vez que objetiva apenas eliminar da imagem pontos evidentemente inviáveis, como por exemplo retas verticais oriundas de portas em corredor, vide Figura II.2

Através de análise vetorial, a direção do gradiente é dada por

$$\alpha(x, y) = \text{atan}\left(\frac{G_y}{G_x}\right) \quad (\text{II.2})$$

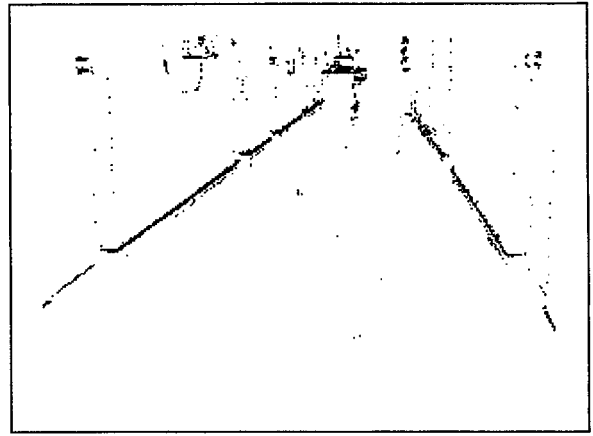
onde o ângulo α é medido em relação ao eixo x da imagem.

Neste ponto aplica-se o limiar para as magnitudes do gradiente e considera-se as direções de interesse. Considerando-se um prévio conhecimento do conteúdo da imagem, para valores acima do limiar e para ângulos na região de interesse, adota-se zero para $G[f(x, y)]$. Nos testes efetuados, constatou-se que os valores adequados para o limiar foram: 40 para aplicações internas e 80 para as externas. A imagem foi discretizada em níveis de cinza de 0 (preto) a 255 (branco). Os ângulos na região de interesse também advém de um conhecimento prévio do ambiente e do conteúdo da imagem. Valores experimentais com ângulos na região de interesse variando de 15° a 75° mostraram-se satisfatórios para os experimentos realizados tanto em ambientes internos quanto externos.

As Figuras II.2 e II.3 ilustram a aplicação do método para obtenção do gradiente da imagem em níveis de cinza.



(a)

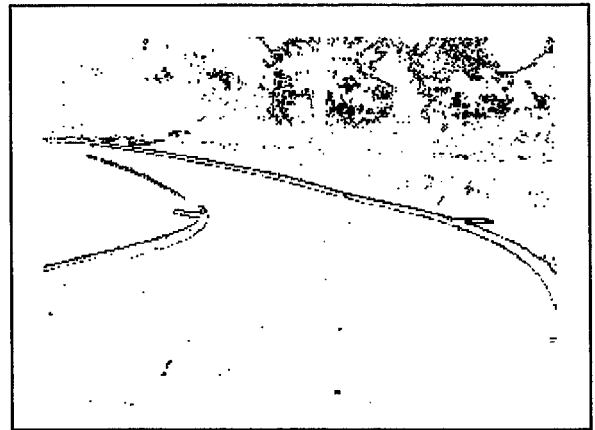


(b)

Figura II.2. Determinação do gradiente para imagem em ambiente interno. A imagem original está em (a). Em (b), os pixels resultantes do operador gradiente.



(a)



(b)

Figura II.3. Determinação do gradiente para imagem externa. A imagem original está em (a). Em (b), os pixels resultantes do operador gradiente.

II.3 Estágio (II): Determinação da Primeira Estimativa Paramétrica

No estágio (II), supõe-se que as guias sejam aproximadas por polinômios de segunda ordem [30], conforme a equação

$$x = a_1 y^2 + a_2 y + a_3 \quad (\text{II.3})$$

onde x e y são as coordenadas horizontais e verticais, respectivamente, e a_1 , a_2 e a_3 são os coeficientes da parábola. Note-se que estes coeficientes governam a forma e a posição da parábola no modelo. Os pontos finais de cada parábola são dados em função dos limites da janela de processamento. Um modelo de segunda ordem foi escolhido por fornecer, com boa aproximação, uma representação da forma real das guias, considerando as necessidades do problema para guiamento de veículos autônomos [41]. Um esboço deste cenário é mostrado na Figura II.4.

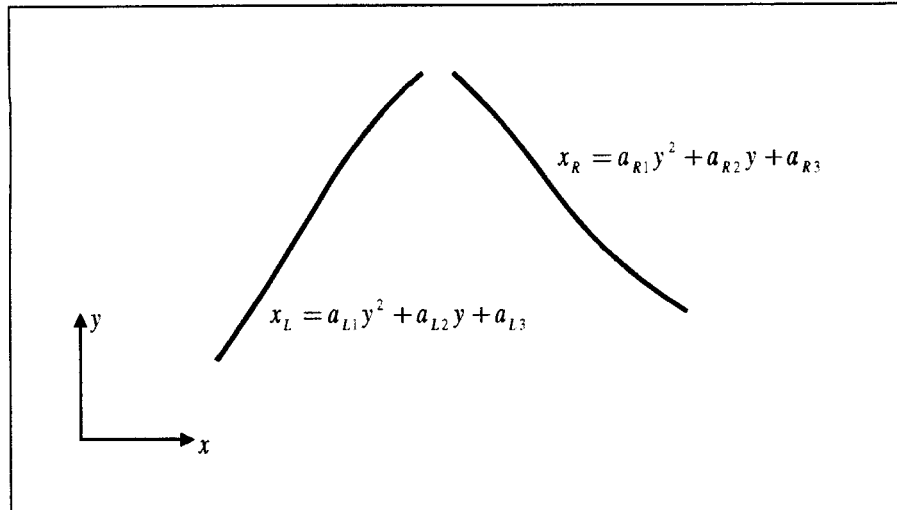


Figura II.4. Modelo para as guias de navegação.

A transformada de Hough é utilizada a fim de se determinar, para cada guia, a melhor reta que aproxime o conjunto de pixels obtido através do operador gradiente, e o critério de mínimos quadrados interpolará uma parábola na região definida por pixels vizinhos à reta. Mais precisamente, a partir da reta resultante da transformada de Hough, determina-se uma banda vertical com 4 pixels acima e 4 pixels abaixo de cada pixel da reta determinada pelo método de Hough. Obtém-se assim uma nuvem de pontos, com base na qual se determina os

coeficientes da parábola que interpola tais pontos, segundo o critério de mínimos quadrados. Estes procedimentos são explicados em detalhes nas Seções II.3.1 e II.3.2.

Esta estimativa é, em geral, muito boa, e serve para inicializar o algoritmo de recursão. Este estágio só é aplicado para a primeira imagem adquirida no processamento ou para a situação onde o número de pixels na imagem se tornar muito pequeno (isto pode ocorrer, por exemplo, pelo fato de o robô se mover em direção a uma parede, onde não há guias a serem detectadas).

Neste estágio, utiliza-se o seguinte procedimento: para o conjunto de pixels resultantes do operador gradiente, tal como nas Figuras II.2(b) e II.3(b), aplica-se a transformada de Hough a fim de se encontrar retas, à esquerda e à direita, que melhor representem este conjunto de pontos.

II.3.1 Transformada de Hough (*TH*)

Uma das técnicas de estimação paramétrica mais usadas, que utiliza mecanismo de eleição, é a *TH*. Neste tipo de mecanismo, cada ponto na curva vota para uma série de combinação de parâmetros, e a combinação de parâmetros que receber a maioria dos votos é declarado como vencedor [13], [24].

No caso desta aplicação, a *TH* é utilizada, como já dito anteriormente, para se determinar uma reta que interpole um conjunto de pixels distribuídos em uma dada faixa de interesse da imagem, o que requer o prévio conhecimento do ambiente e do conteúdo da cena.

A motivação para o uso da *TH* é a seguinte: suponha-se inicialmente que existam n pixels dados no plano xv da imagem, conforme a Figura II.5, e que se deseja encontrar subconjuntos que formem linhas retas. Este problema foi abordado por Hough em 1962 e pode ser considerado da seguinte forma:

Considere-se a parametrização da reta em coordenadas polares, tal que

$$\rho = x \cos \varphi + y \sin \varphi \quad (\text{II.4})$$

onde x e y são as coordenadas dos pontos da reta e ρ e φ são o raio e ângulo das coordenadas polares que definem a reta, respectivamente.

Para a determinação da reta vencedora, procede-se conforme a seguir:

1. Definem-se os valores máximos e mínimos permitidos para a discretização do espaço de parâmetros: raios e ângulos da reta no método de Hough. Considerando-se o prévio conhecimento do ambiente e do conteúdo da cena, atribuiu-se busca exaustiva para ângulos variando de 5° a 75° , com incrementos de 10° . Para os raios, considerando que a imagem formada é da dimensão de 320×240 pixels, tomaram-se raios entre 50 e 350 pixels, com incrementos de 2 pixels. Tais valores mostraram-se adequados para os experimentos realizados tanto em ambientes internos quanto externos. Inicializa-se uma matriz com valores nulos, que guardará em seus elementos o número de votos de cada possível reta. A dimensão desta matriz é dada pelo número de ângulos e raios visitados pelo método.
2. Para cada par (x, y) dentro da janela de processamento da imagem, após se aplicar o operador gradiente, verifica-se se o pixel é preto. Em caso afirmativo, segue para o passo 3, caso contrário testa-se um novo par de (x, y) .
3. Para cada ângulo permitido na nos limites de varredura, calcula-se o raio correspondente, conforme (II.4). Se este raio estiver dentro dos valores máximos e mínimos predefinidos, o elemento da matriz dos votos correspondente é acrescido de '1'.
4. No final do procedimento, analisa-se a matriz acima para se encontrar o elemento com maior número de votos. Com este valor, recupera-se o raio e ângulo correspondentes.

Estes valores definem a reta no método de Hough vencedora, que melhor interpola a nuvem de pixels advinda do operador gradiente.

O resultado do processamento é conforme a Figura II.5, onde a reta desenhada em linha cheia é a vencedora, portanto a reta pelo método de Hough. As outras retas exemplificam o procedimento do algoritmo de busca e votação. Nota-se que a reta vencedora é a que melhor aproxima os pixels na vizinhança.

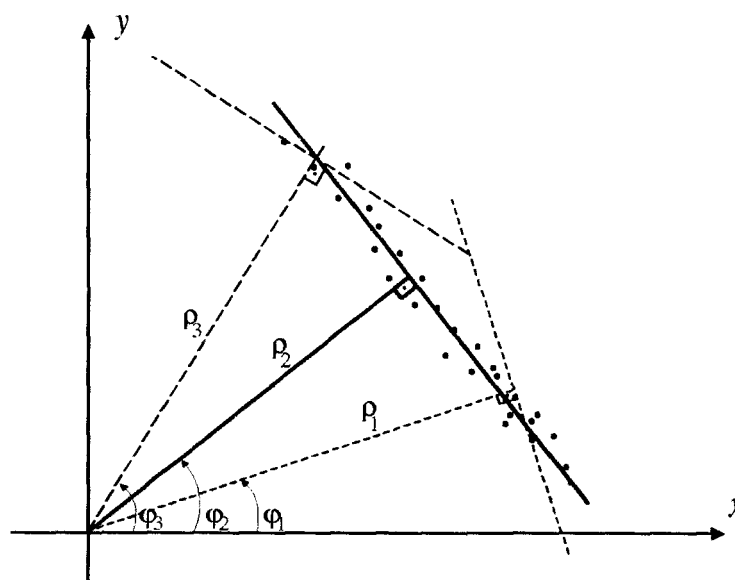


Figura II.5. Cenário para a determinação da reta via transformada de Hough.

II.3.2 Critério de Mínimos Quadrados (MQ) para Interpolação da Parábola

Após a TH, determina-se a abertura de uma banda vertical, para cima e para baixo, em cada ponto da reta, o que permite definir um novo conjunto de pixels a ser utilizado na interpolação da parábola pelo método dos mínimos quadrados. Nos testes efetuados, constatou-se que o valor adequado da banda situa-se em torno de 4 pixels.

O procedimento utilizado neste passo é baseado na aproximação de funções via mínimos quadrados. Neste caso, deseja-se estabelecer um polinômio de segundo grau $x(y)$ que interpole uma série de pontos (x, y_i) da melhor maneira possível.

Este critério pode ser expresso da forma

$$\hat{a} = \underset{a \in \mathbb{R}^3}{\text{ArgMin}} \sum_{i=1}^N (x_i - x(y_i))^2 \quad (\text{II.5})$$

Considerando-se N pixels encontrados na banda vertical definida, tem-se o sistemas de equações

$$\begin{cases} a_1 y_{t,1}^2 + a_2 y_{t,1} + a_3 = x_{t,1} \\ a_1 y_{t,2}^2 + a_2 y_{t,2} + a_3 = x_{t,2} \\ \vdots \\ a_1 y_{t,N}^2 + a_2 y_{t,N} + a_3 = x_{t,N} \end{cases} \quad (\text{II.6})$$

A Equação (II.6) pode ser escrita na forma vetorial,

$$M \cdot a = X \quad (\text{II.7})$$

onde

$$M = \begin{bmatrix} y_{t,1}^2 & y_{t,1} & 1 \\ y_{t,2}^2 & y_{t,2} & 1 \\ \vdots & \vdots & \vdots \\ y_{t,N}^2 & y_{t,N} & 1 \end{bmatrix}; \quad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad \text{e} \quad X = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,N} \end{bmatrix}$$

Supondo-se $M^T M$ inversível, tem-se:

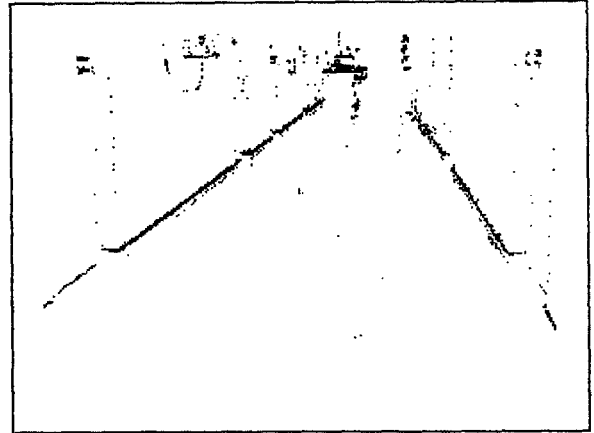
$$a = (M^T M)^{-1} M^T X \quad (\text{II.8})$$

que dá os coeficientes da parábola interpolada por MQ.

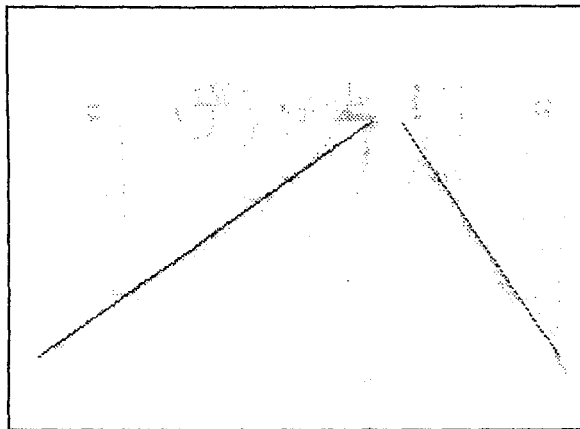
As figuras seguintes apresentam resultados obtidos para processamento de imagens tanto internas quanto externas na obtenção da primeira estimativa paramétrica, ou seja, considerando-se os estágios (I) e (II) do algoritmo de processamento de imagens.



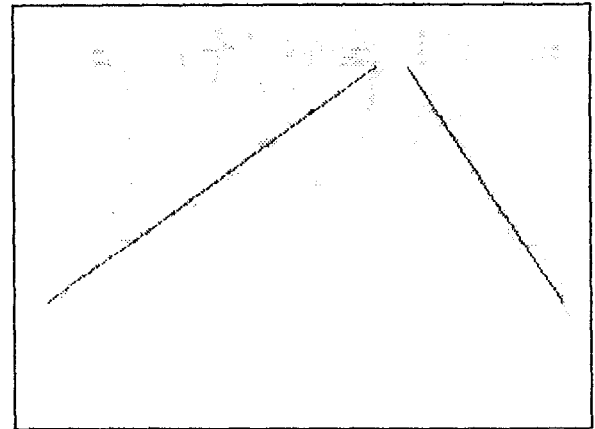
(a)



(b)



(c)



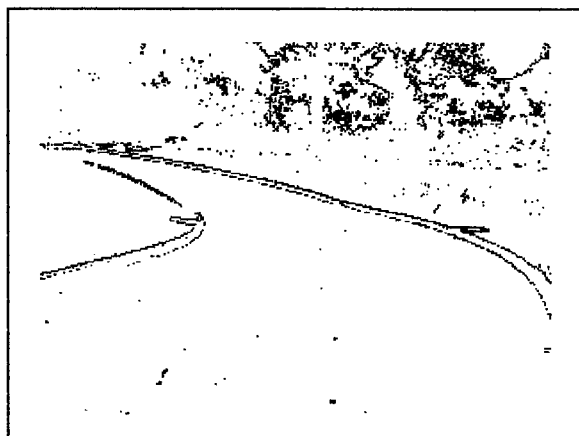
(d)

Figura 11.6. Procedimento para geração da primeira estimativa dos coeficientes da parábola para imagem em ambiente interno. A imagem original em ^{níveis} *de cinza* está em (a). Em (b), os pixels resultantes do operador gradiente. A reta resultante da transformada de Hough está em (c), e em (d), a parábola interpolada por mínimos quadrados.

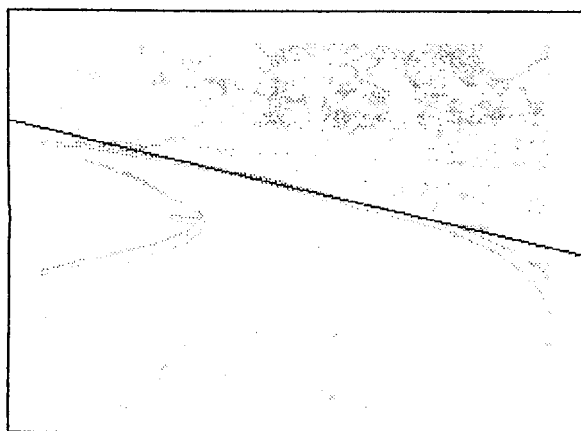
Neste caso, os coeficientes são tais que a parábola resultante se assemelha ao resultado obtido na Figura II.6(c), dado se tratar das guias de um corredor retilíneo.



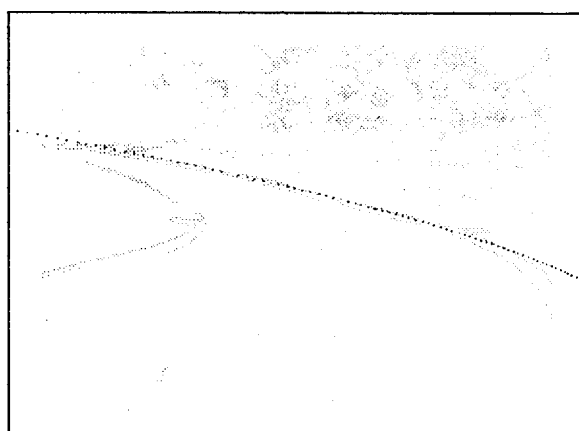
(a)



(b)



(c)



(d)

Figura II.7. Procedimento para geração da primeira estimativa dos coeficientes da parábola para imagem externa. A imagem original em *níveis de cinza* está em (a). Em (b), os pixels resultantes do operador gradiente. A reta resultante da transformada de Hough está em (c), e em (d), a parábola interpolada por mínimos quadrados.

A parábola se apresenta ligeiramente diferente da reta em Figura II.7(c), dado que os pixels na nuvem estão distribuídos sobre a curva na imagem externa. Como a curva a esquerda não se apresenta propícia para identificação, mostra-se apenas a curva a direita. Esta posição não seria, por exemplo, adequada para se iniciar o processo de identificação das guias.

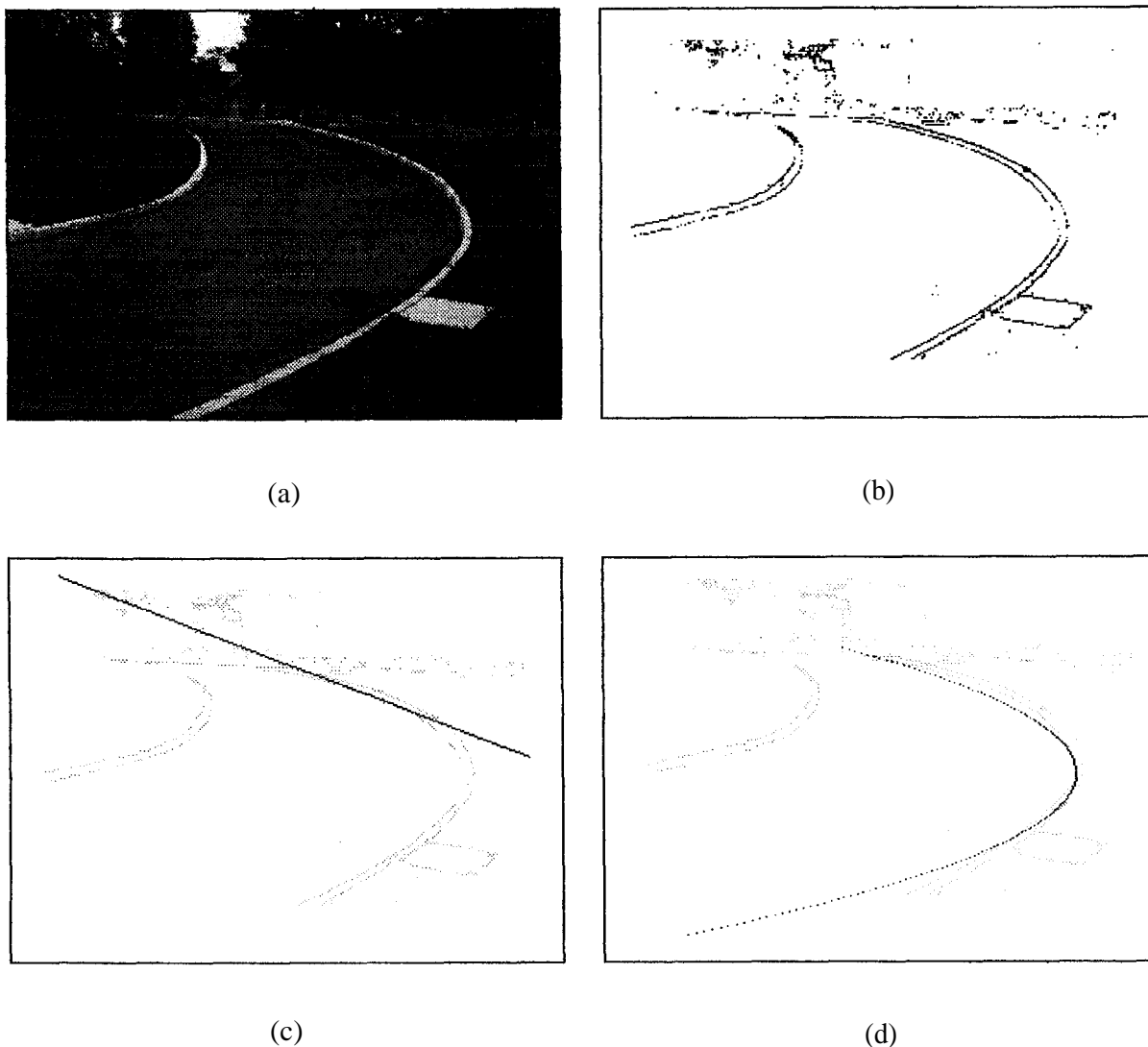


Figura II.8. Procedimento para geração da primeira estimativa dos coeficientes da parábola para outra imagem externa. A imagem original em ^{níveis} de cinza está em (a). Em (b), os pixels resultantes do operador gradiente. A reta resultante da transformada de Hough está em (c), e em (d), a parábola interpolada por mínimos quadrados.

Para esta situação, a parábola na Figura II.8(d) é totalmente diferente da reta na Figura II.8(c), pois os pontos da reta não conseguem definir a curva, que neste caso é bastante acentuada. Porém, devido à banda vertical de pixels estabelecida, a estimativa via *MQ* é também adequada.

II.4 Estágio (III): Recursão para Atualização da Estimativa Paramétrica

Para o processamento de, imagens *on-line* é necessário se recalcularem a_1 , a_2 e a_3 para cada nova imagem. Esta solução pode ser recursivamente formulada e é efetuada no estágio (III).

Há várias possibilidades para a utilização dos métodos recursivos. Três métodos foram testados e comparados para a atualização: o algoritmo de mínimos quadrados recursivos ponderados com decaimento exponencial, *Weighted Recursive Least Squares (WRLS)*, proposto em [41], o algoritmo de mínimos quadrados normalizado, *Normalized Least Mean Squares (NLMS)*, e um método baseado no filtro de Kalman na forma de informação, *Kalman Information Filter (KIF)*, proposto nesta tese e que apresenta desempenho mais adequado para a atualização da estimativa paramétrica.

Nesta seção, estes algoritmos serão apresentados e discutidos, bem como suas propriedades e limitações.

II.4.1 O Algoritmo de Mínimos Quadrados Recursivos Ponderado com Decaimento Exponencial (WRLS)

O *WRLS* efetua um compromisso entre a importância de novos dados coletados em relação a dados passados na estimativa. O algoritmo preserva todas as informações passadas, representadas pela estimativa dos coeficientes e pela covariância estimada do estado anterior. Uma ponderação exponencial é usada para enfatizar o efeito dos dados atuais em relação aos dados passados.

Para o *WRLS* com decaimento exponencial, considera-se a seguinte função custo,

$$S_{N+1}(\theta) = \lambda S_N(\theta) + (y_{N+1} - \phi_{N+1}^T \theta)^2 \quad (\text{II.9})$$

onde $0 < \lambda < 1$. Note-se que, se $\lambda = 1$, resulta a função custo padrão para mínimos quadrados.

Portanto, para $0 < \lambda < 1$, tem-se o estimador de mínimos quadrados recursivo com ponderação exponencial que satisfaz

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_{t+1}[y_{t+1} - \phi_{t+1}^T \hat{\theta}_t] \quad (\text{II.10})$$

com

$$K_{t+1} = P_t \phi_{t+1} (\lambda I + \phi_{t+1}^T P_t \phi_{t+1})^{-1} \quad (\text{II.11})$$

e P_t é computado recursivamente

$$P_{t+1} = \lambda^{-1} (I - K_{t+1} \phi_{t+1}^T) P_t \quad (\text{II.12})$$

onde

$$\phi_{t+1}^T = \begin{bmatrix} 1 & y_{t,1} & y_{t,1}^2 \\ 1 & y_{t,2} & y_{t,2}^2 \\ \vdots & \vdots & \vdots \\ 1 & y_{t,N} & y_{t,N}^2 \end{bmatrix}, \quad \hat{\theta}_t = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \text{ e } y_{t+1} = \begin{bmatrix} x_{t,1} \\ x_{t,2} \\ \vdots \\ x_{t,N} \end{bmatrix} \quad (\text{II.13})$$

O **WRLS** é também denominado algoritmo recursivo com fator de esquecimento (*forgetting factor*). Isto concede ao **WRLS** uma característica de adaptação, fator bastante desejado em sistemas dinâmicos variantes no tempo.

No caso de aplicação em processamento de imagens com o objetivo de se adequar uma parábola a um conjunto de pixels, o **WRLS** demonstra capacidade de interpolar parábolas muito semelhantes às vistas na imagem real, com um adicional de que a imagem não necessita ser estática, pelo contrário, várias imagens são capturadas e processadas por segundo.

Todavia, na implementação do **WRLS** há a necessidade de inversão de matrizes de grande dimensão, o que aumenta o risco de mal-condicionamento de matrizes e erro numérico, além de ser um procedimento computacional mais lento, onde a ordem da matriz varia de acordo com a imagem obtida a cada instante. Isto pode ser contornado pelo uso do *Square Root Filter (SRF)* [3], [5] e [37], que evita a inversão matricial, tem bom desempenho

e é matematicamente equivalente a (II.10), (II.11) e (II.12), contudo a *SRF* possui codificação mais complexa e necessita tratar todas as leituras individualmente.

No caso da *WRLS*, o único parâmetro a ser ajustado é λ . Se $\lambda=1.0$, todas as imagens teriam o mesmo peso. Para $0 < \lambda < 1$, tem-se o fator de decaimento exponencial, que confere pesos maiores para imagens mais recentes. Neste trabalho, foram testados valores na faixa de $0.5 < \lambda < 0.75$, e os valores experimentais mais adequados encontraram-se próximos de 0.70.

Outra dificuldade encontrada na implementação do *WRLS* está relacionada com o fato que não se conseguiu garantir, na forma original do algoritmo, a positividade definida nem a simetria da matriz de covariância P_t , o que induziu durante a implementação a erros numéricos e resultados não desejados. Esta dificuldade foi contornada neste trabalho empregando-se uma forma alternativa para a equação da matriz de covariância, o que foi alcançado, fazendo que P_t seja sempre positiva definida de acordo com a equação

$$P_{t+1} = \lambda^{-1}(I - K_{t+1}\phi_{t+1}^T)P_t(I - K_{t+1}\phi_{t+1}^T)^T + K_{t+1}K_{t+1}^T \quad (\text{II.14})$$

A Equação II.14 pode ser verificada igualando-se as Equações (II.14) e (II.12) de onde tem-se

$$(I - K_{t+1}\phi_{t+1}^T)P_t(I - K_{t+1}\phi_{t+1}^T)^T + \lambda K_{t+1}K_{t+1}^T = (I - K_{t+1}\phi_{t+1}^T)P_t \quad (\text{II.15})$$

Multiplicando-se $(I - K_{t+1}\phi_{t+1}^T)P_t(I - K_{t+1}\phi_{t+1}^T)^T$ e comparando-se termo a termo a igualdade em (II.15), deve-se ter

$$-P_{t+1}\phi_{t+1}K_{t+1} + K_{t+1}\phi_{t+1}^TP_t\phi_{t+1}K_{t+1}^T + \lambda K_{t+1}K_{t+1}^T = 0 \quad (\text{II.16})$$

A Equação (II.16) pode ser reescrita como

$$K_{t+1}(-K_{t+1}^{-1}P_{t+1}\phi_{t+1}K_{t+1}^{-T} + \phi_{t+1}^TP_t\phi_{t+1} + \lambda I)K_{t+1}^T = 0 \quad (\text{II.17})$$

c, de (II.11), segue que

$$K_{t+1}^{-1} = (\lambda I + \phi_{t+1}^T P_t \phi_{t+1}) \phi_{t+1}^{-1} P_{t+1}^{-1} \quad (\text{II.18})$$

Substituindo-se (II.18) em (II.17) vê-se que a igualdade se verifica, e portanto (II.14) é verdadeira e pode ser utilizada para substituir (II.12), com o adicional de se garantir que P conserva sua propriedade de positividade definida.

II.4.2 Filtro de Kalman na forma de Informação (KIF)

Para cada conjunto de dados provenientes de uma nova imagem, o algoritmo estima novos coeficientes para as parábolas.

O sistema considerado na estimação é descrito pelas equações

$$\hat{\theta}_{t+1} = F_t \hat{\theta}_t + G_t \omega_t \quad (\text{II.19})$$

e

$$x_t = \phi_t^T \hat{\theta}_t + v_t \quad (\text{II.20})$$

onde

$\hat{\theta} \in \mathbb{R}^{3 \times 1}$ são as estimativas dos coeficientes da parábola;

$\phi^T \in \mathbb{R}^{N \times 3}$ é definido conforme (II.13);

$x \in \mathbb{R}^{N \times 1}$ é definido conforme (II.13);

$F_t \in \mathbb{R}^{3 \times 3}$ é a matriz que relaciona $\hat{\theta}_{t+1}$ com $\hat{\theta}_t$ na ausência de uma função forçante;

$G_t \in \mathbb{R}^{3 \times 3}$ é a matriz que pondera o efeito de ω_t ;

ω_t e v_t são os ruídos de estado e medida, respectivamente, modelando as incertezas das estimativas dos coeficientes das parábolas e também da captura de cada nova imagem, com variâncias Q_t e R_t .

As variâncias Q_t e R_t , positivas definidas, são dadas por

$$Q_t = E[\omega_t \omega_t^T] > 0 \quad (\text{II.21})$$

c

$$R_t = E[v_t v_t^T] > 0 \quad (\text{II.22})$$

O filtro de Kalman na forma de informação deriva da aplicação do *matrix inverse lemma* (MIL) às bem conhecidas equações do filtro de Kalman discreto usual [6], [19]. As equações decorrentes desta transformação, usadas no filtro são

$$P_{t|t}^{-1} = P_{t|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (\text{II.23})$$

$$P_{t+1|t}^{-1} = (I - B_t G_t^T) A_t \quad (\text{II.24})$$

onde

$$B_t = A_t G_t (G_t^T A_t G_t + Q_t^{-1})^{-1} \quad (\text{II.25})$$

e

$$A_t = (F_t^{-1})^T P_{t|t}^{-1} F_t^{-1} \quad (\text{II.26})$$

A matriz de ganho do filtro pode ser expressa em função de $P_{t|t-1}^{-1}$, isto é,

$$K_t = F_t P_{t|t-1}^{-1} \phi_t R_t^{-1} \quad (\text{II.27})$$

e as estimativas paramétricas são obtidas através de relações indiretas segundo as equações

$$\hat{\theta}_{t+1|t} = (I - B_t G_t^T) F_t^{-1} \hat{\theta}_{t|t} \quad (\text{II.28})$$

e

$$\hat{\theta}_{t|t} = \hat{\theta}_{t|t-1} + \phi_t R_t^{-1} \phi_t \quad (\text{II.29})$$

O conjunto de equações de (II.23) a (II.29) constituem as equações do *KIF*, que são algebricamente equivalentes às equações do filtro de Kalman usual. Contudo, nesta aplicação,

a dimensão do vetor de leitura é elevada, e o filtro de Kalman usual necessitaria inverter uma matriz desta dimensão. Por outro lado, o *KIF* requer a inversão de matriz 3x3. Logo, apresenta grande simplicidade computacional e robustez à propagação de erro.

Uma melhoria adicional neste algoritmo, que é proposta neste trabalho, consiste na obtenção de uma expressão para atualizar a matriz de covariância associada ao *KIF*, que preserva a propriedade positiva definida.

Apresenta-se, a seguir, a obtenção de uma forma de atualizar a matriz $P_{t+1|t}^{-1}$, garantindo-se a manutenção da propriedade positiva definida.

II.4.2.1 Atualização da Matriz de Covariância do *KIF*

Seja a Equação (II.23), e considere-se as Equações (II.24), (II.25) e (II.26) defasadas de um período de amostragem Tem-se, portanto,

$$P_{t|t-1}^{-1} = (I - B_{t-1}G_{t-1}^T)A_{t-1} \quad (II.30)$$

$$B_{t-1} = A_{t-1}G_{t-1}(G_{t-1}^T A_{t-1}G_{t-1} + Q_{t-1}^{-1})^{-1} \quad (II.31)$$

C

$$A_{t-1} = (F_{t-1}^{-1})^T P_{t-1|t-1}^{-1} F_{t-1}^{-1} \quad (II.32)$$

Substituindo-se (II.32) em (II.31) e o resultado em (II.30), segue-se que

$$P_{t|t-1}^{-1} = P_{t-1|t-1}^{-1} - P_{t-1|t-1}^{-1}(P_{t-1|t-1}^{-1} + Q_{t-1}^{-1})^{-1}P_{t-1|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (II.33)$$

Seja agora

$$T_t = P_{t-1|t-1}^{-1}(P_{t-1|t-1}^{-1} + Q_{t-1}^{-1})^{-1} > 0 \quad (II.34)$$

Logo, de (II.33) tem-se

$$P_{t|t-1}^{-1} = P_{t-1|t-1}^{-1} - T_t P_{t-1|t-1}^{-1} + \phi_t R_t^{-1} \phi_t^T \quad (II.35)$$

Analisando-se (II.35), vê-se que a parcela $\phi_t R_t^{-1} \phi_t^T$ é positiva definida, bastando encontrar uma relação que garanta que $P_{t-1|t-1}^{-1} - T_t P_{t-1|t-1}^{-1}$ seja positiva definida.

Para tanto, escreve-se

$$P_{t-1|t-1}^{-1} - T_t P_{t-1|t-1}^{-1} = (I - T_t) P_{t-1|t-1}^{-1} (I - T_t)^T + M_t > 0 \quad (\text{II.36})$$

onde $M_t \in R^{3 \times 3}$ é uma matriz positiva definida a ser determinada. Desenvolvendo-se o lado direito antes da desigualdade de (II.36), verifica-se que para que $P_{t-1|t-1}^{-1} - T_t P_{t-1|t-1}^{-1} = (I - T_t) P_{t-1|t-1}^{-1} (I - T_t)^T + M_t$ seja verdadeira, deve-se ter

$$-P_{t-1|t-1}^{-1} T_t^T + K_t P_{t-1|t-1}^{-1} T_t^T + M_t = 0 \quad (\text{II.37})$$

que pode ser reescrita como

$$T_t [-T_t^{-1} P_{t-1|t-1}^{-1} + P_{t-1|t-1}^{-1} + T_t^{-1} M_t (T_t^{-1})^T] T_t^T = 0 \quad (\text{II.38})$$

Por outro lado, de (II.34) tem-se

$$T_t^{-1} = (P_{t-1|t-1}^{-1} + Q_{t-1}^{-1}) P_{t-1|t-1} \quad (\text{II.39})$$

Logo, substituindo-se (II.39) em (II.38), lembrando-se que Q_{t-1} e T_t são positivas definidas, verifica-se que (II.36) é satisfeita para

$$M_t = T_t Q_{t-1}^{-1} T_t^T > 0 \quad (\text{II.40})$$

que origina finalmente a expressão positiva definida para atualização da matriz de covariância no *KIF*, isto é,

$$P_{t|t}^{-1} = (I - T_t) P_{t-1|t-1}^{-1} (I - T_t)^T + T_t Q_{t-1}^{-1} T_t^T + \phi_t R_t^{-1} \phi_t^T \quad (\text{II.41})$$

Para o *KIF*, apenas o conhecimento da imagem no instante atual e a estimativa imediatamente anterior são suficientes para se determinar a estimativa atual desses

coeficientes. Apenas as covariâncias dos ruídos, Q , e R , dadas pelas Equações (II.21) e (II.22), devem ser especificadas inicialmente. A estrutura destas matrizes de covariância é aqui suposta diagonal, sendo seus valores ajustados de acordo com a incerteza das estimativas dos coeficientes da parábola e da captura de cada nova imagem, o que requer também um conhecimento prévio do ambiente a ser testado. Testes efetuados neste trabalho demonstraram a robustez a estas covariâncias, pois variou-se suas inicializações da ordem de dez vezes, e não houve degradação na estimativa dos coeficientes das parábolas.

11.4.3 O Algoritmo de Mínimos Quadrados Normalizado (NLMS)

Um outro método recursivo investigado no estudo para estimação dos parâmetros da parábola foi o *normalized least-mean-squares (NLMS)* [3], também conhecido como algoritmo de projeção. A equação que rege a atualização paramétrica é dada por

$$\hat{\theta}_t = \hat{\theta}_{t-1} + a\phi_{t-1}(cI + \phi_{t-1}^T\phi_{t-1})^{-1}(y_t - \phi_{t-1}^T\hat{\theta}_{t-1}) \quad (\text{II.42})$$

com $\hat{\theta}_0$ dado e $c > 0$; $0 < a < 2$ e $\hat{\theta}_{t-1}$ representando a estimativa paramétrica no instante $t-1$. A constante c serve, no caso escalar, para evitar divisão por zero e, no caso vetorial, para melhorar o condicionamento da matriz a ser invertida.

Uma avaliação experimental do comportamento do algoritmo *NLMS* realizado neste trabalho, no caso de estimação paramétrica *on-line* para parábolas de interesse, conduziram às seguintes conclusões:

1. Os valores de a mais convenientes (para as aplicações de processamento de imagens para detecção de guias para navegação de robôs móveis) estão próximos de 1, sendo que valores menores que 1 tornam a convergência da estimativa mais lenta.
2. O algoritmo mostra-se susceptível a erros, pois a matriz $\phi_t^T\phi_t$, de dimensão $N \times N$, que deveria ser invertida, apresentava-se, em muitos casos, quase singular devido à

proximidade dos valores que compunham ϕ_i^T . Mais precisamente, tem-se imagens de dimensão 320x 240 pixels, e portanto quando encontra-se, através do processamento da imagem, parábolas de interesse, suas coordenadas (i, j) são muito próximas, o que leva a matriz $\phi_i^T \phi_i$ a tornar-se quase singular, o que é extremamente indesejável quando se trata de inversão de matrizes. Ainda relativamente ao problema de inversão de matrizes, segue a verificação de que, quanto maior fosse o valor de c, melhor condicionada ficaria a matriz $cI + \phi_{i-1}^T \phi_{i-1}$, o que auxiliaria na inversão. Porém a precisão da estimativa paramétrica ficaria bastante comprometida, dado que, como dito anteriormente, c é, usada apenas para evitar singularidade na matriz citada.

Via experimentos, concluiu-se neste trabalho que os valores mais adequados para as constantes a e c são $a = 1.0$ e $c = 0.1$.

3. Para uma boa estimação são necessários pontos suficientes. Na implementação do NLMS, verificou-se dificuldade na confiabilidade e robustez dos resultados obtidos do algoritmo quando os pontos selecionados no conjunto de pixels em torno da parábola eram muitos. Portanto, de um lado, havia a necessidade de pontos suficientes para uma boa estimativa; de outro, muitos pontos agravavam a dificuldade de inversão, pois $\phi_i^T \phi_i$ torna-se mais mal condicionada à medida em que a sua dimensão se eleva.

Uma possível alternativa para melhorar o desempenho do NLMS seria utilizar o *Orthogonalized Projection Algorithm* [16], dado pela equação

$$\hat{\theta}_i = \hat{\theta}_{i-1} + P_{i-2} \phi_{i-1} (\phi_{i-1}^T P_{i-2} \phi_{i-1})^{-1} (y_i - \phi_{i-1}^T \hat{\theta}_{i-1}) \quad (II.43)$$

onde

$$P_{i-1} = P_{i-2} - P_{i-2} \phi_{i-1} (\phi_{i-1}^T P_{i-2} \phi_{i-1})^{-1} \phi_{i-1}^T P_{i-2} \quad (II.44)$$

No entanto, este algoritmo apresenta custo computacional muito próximo daquele exibido pelo *WRLS*, no sentido de que envolve inversão de matrizes de ordem elevada, motivo pelo qual não foi investigado.

II.5 Resultados Experimentais e Discussões

A placa de aquisição de imagens utilizada no experimento é a Intel Smart Video Recorder III, com processador Brooktree BT848. Foi utilizado um conjunto câmera CCD e receptor modelo TARGA[®] IX, *wireless*, em sistema NTSC, transmitindo na frequência de 2.4 GHz. A distância entre transmissor e receptor pode atingir até 200 m, sem a presença de obstáculos. As imagens adquiridas são de dimensão 320 x 240 pixels. A câmera possui lente com distância focal de 11 mm e CCD de $1/3''$. O algoritmo foi executado em um PENTIUM II, 450 MHz, com 128 Mbytes de memória RAM, em sistema operacional LINUX, distribuição RED HAT 5.2, versão de Kernel 2.0.36. O driver utilizado na captura das imagens está disponível em <http://hydra.dis.uniroma1.it/iocchi/bt848>.

A Tabela II.1 apresenta uma comparação entre os tempos de processamento para os possíveis métodos de recursão utilizados no algoritmo.

Os valores de tempo foram obtidos para uma imagem padrão, e o número de pixels máximo para definição de cada guia foi de 150 pontos. Obviamente, o tempo de processamento varia para cada nova imagem, porém a Tabela II.1 é bastante ilustrativa para explicitar o desempenho do algoritmo dado que, para diversas situações, os valores na tabela podem ser considerados médios.

Tabela II.1. Comparação do tempo de processamento entre os métodos recursivos.

<i>Método Recursivo</i>	<i>Tempo de Processamento (ms)</i>
Transformada de Hough + Mínimos Quadrados	122.2
KIF	8.2
WRUS	22.2

O processamento mais básico, que é comum a todos os métodos da Tabela II.1, é o de captura e obtenção da imagem em *níveis de cinza*, e a taxa de aquisição de imagens é de *30.0 fps* (quadros por segundo), o que resulta em *33.3 ms* para cada nova imagem em *níveis de cinza*. Em seguida, o operador gradiente, que também é necessário, requer *11.1 ms*, baixando para *22.5 fps* o número de quadros processados por segundo.

A Tabela II.1 justifica o uso do *KIF* para atualização das estimativas dos coeficientes das parábolas em detrimento do *WRLS*. Os resultados experimentais neste capítulo foram obtidos utilizando-se o *KIF*.

Conforme a Tabela II.1, percebe-se também que o processamento envolvendo a transformada de Hough e interpolação por mínimos quadrados apresenta custo computacional muito elevado, apesar de a estimativa resultante ser muito precisa. Por este motivo, é utilizado apenas nas condições de inicialização do algoritmo na Figura II.1. Embora haja procedimentos de menor custo computacional para determinação da transformada de Hough, vide [24], não é apenas o tempo de processamento que impede a utilização da combinação *TH+MQ*: ao contrário do *KIF* e do *WRLS*, não há procedimento recursivo eficiente para atualização das estimativas dos coeficientes da parábola.

As Figuras II.9, II.10 e II.11 apresentam, respectivamente, as imagens em *níveis de cinza*, a imagem binarizada (operador gradiente) e as guias detectadas pelo algoritmo em tempo real. As imagens foram coletadas no corredor da Divisão de Engenharia Eletrônica do ITA.

As Figuras II.12, II.13 e II.14 apresentam a mesma sequência para imagens externas, obtidas no Centro Técnico Aeroespacial (CTA).



Figura II.9. Imagem em ambiente interno em níveis de cinza.

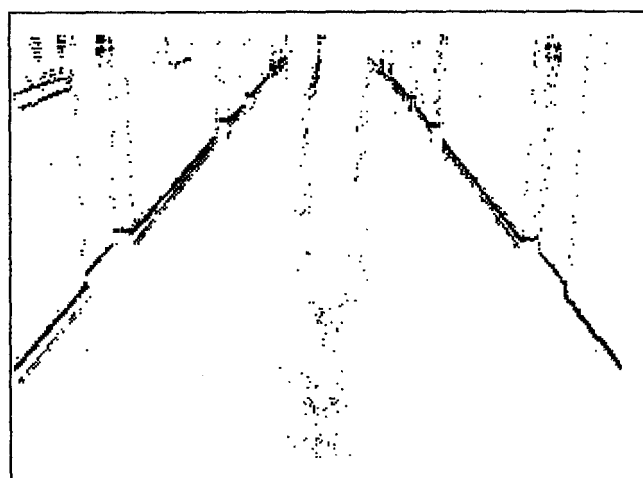


Figura II. 10. Pixels resultantes do operador gradiente para imagem em ambiente interno.

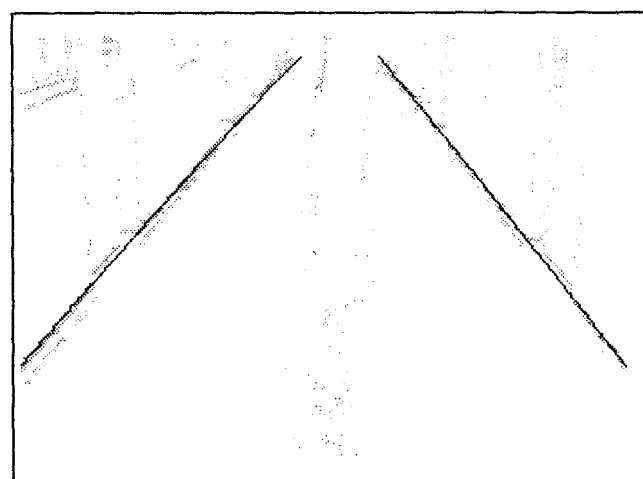


Figura II.11. Guias para navegação detectadas para imagem em ambiente interno.



Figura II.12. Imagem externa em níveis de cinza.



Figura II. 13. Pixels resultantes do operador gradiente para imagem externa.

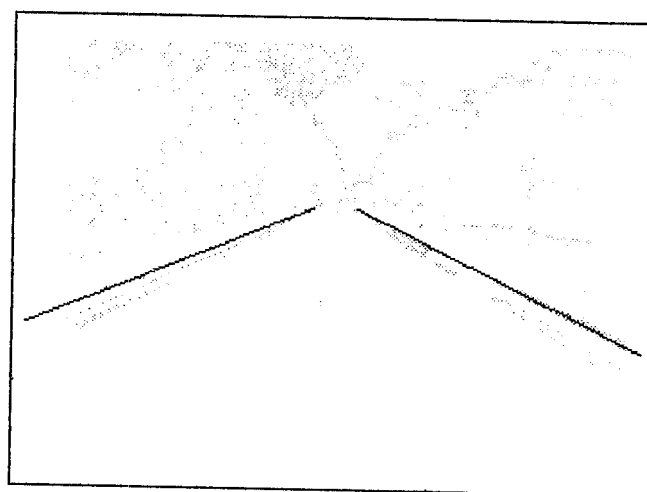


Figura II. 14. Guias para navegação detectadas para imagem externa.

II.6 Conclusões

Com os testes efetuados tanto para aplicações internas quanto externas, constatou-se a robustez do algoritmo para detecção de guias para navegação de robôs móveis baseado no KY, mesmo em presença de situações desfavoráveis como sombra, baixa luminosidade e irregularidades no pavimento da estrada. As Figuras II.12, II.13 e II.14 reforçam esta conclusão. Para aplicações externas, devido à presença de árvores, houve a formação de sombras na imagem, o que pode ser solucionado aplicando-se novos valores de limiar para o operador gradiente.

O algoritmo proposto tem o objetivo de detectar guias para navegação de robôs móveis. O processamento envolveu três importantes estágios: a) extração de características de baixo nível (detecção por operador gradiente) da imagem, b) detecção de retas e interpolação de parábolas de interesse, e c) algoritmo recursivo para atualização destes coeficientes. Os resultados experimentais apontam no sentido da validação do procedimento, que mesmo em presença de situações desfavoráveis, obteve bons resultados. Uma análise experimental mais aprofundada seria recomendada., de modo a se exibir estatisticamente a influência dos parâmetros de projeto no desempenho do algoritmo. Isto porém necessitaria de um aparato mais sofisticado que o disponível no momento.

O sistema de processamento de imagens atingiu média de atualização dos coeficientes da parábola da ordem de 19 Hz, sem se considerar o guiamento do robô móvel.

A seguir, na Seção II.7, analisa-se a obtenção da informação das coordenadas do mundo físico para o robô móvel através da informação visual, obtida por uma transformação de perspectiva, com o objetivo de se efetuar o controle de posição e velocidade do robô.

II.7 Transformação de Perspectiva para Guiamento de Robô Móvel

A transformação de perspectiva permite recuperar parcialmente informação sobre o mundo tridimensional a partir de informações da imagem (mundo bidimensional). Esta transformação projeta um ponto no mundo (3D) em um plano (2D). Por mundo, entenda-se a referência ao sistema tridimensional.

Um modelo do processo de formação de uma imagem em relação a um ponto no mundo é mostrado na Figura II.15. Note-se que se mostra um arranjo mais conveniente para compreensão do processo de formação de imagem em um dispositivo, por exemplo, na câmera, pois a imagem real é formada atrás das lentes e é invertida em relação ao objeto. Na Figura II.15 coloca-se o plano da imagem em uma posição simétrica com relação ao centro da lente, gerando uma imagem não-invertida [12].

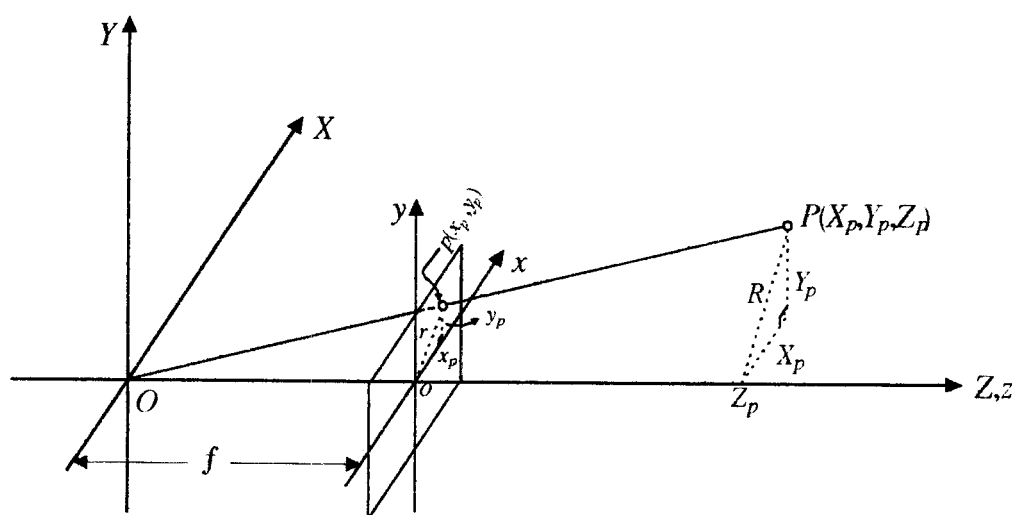


Figura II.15. Modelo básico para projeção em perspectiva.

Na Figura II.15, tem-se o sistema de coordenadas do mundo $(X \ Y \ Z)$, e o sistema de coordenadas da imagem $(x \ y \ z)$. O plano da imagem é paralelo ao plano $(X \ Y)$ do mundo e o centro óptico da lente da camera coincide com a origem 0 do sistema $(X \ Y \ Z)$.

A distância entre os pontos de origem O e o é dada pela distância focal da lente, representada por f .

Seja um ponto representativo no mundo, dado por $P(X_p, Y_p, Z_p)$. Este ponto é projetado em $p(x_p, y_p)$ no sistema de imagem. Os triângulos formados pelos vértices O, p e P e O, P, F são semelhantes, e portanto a razão dos lados correspondentes deve ser a mesma, donde

$$\frac{f}{Z_p} = \frac{r}{R} \quad (11.45)$$

Observando-se ainda os triângulos formados por x_p, y_p e r e por X_p, Y_p e R , percebe-se que são semelhantes. Portanto,

$$\frac{x_p}{X_p} = \frac{y_p}{Y_p} = \frac{r}{R} \quad (11.46)$$

e combinando as Equações (11.45) e (11.46), encontram-se as equações para projeção em perspectiva,

$$x_p = \frac{f}{Z_p} X_p \quad (11.47)$$

e

$$y_p = \frac{f}{Z_p} Y_p \quad (11.48)$$

Torna-se mais conveniente expressar as equações acima sob a forma matricial. Isto é obtido através da transformação das coordenadas cartesianas no espaço euclidiano (X, Y, Z) em coordenadas homogêneas. As coordenadas homogêneas são obtidas através da inserção de um quarto elemento no sistema, definido arbitrariamente por uma constante não-nula k , de forma a se obter (kX, kY, kZ, k) . Percebe-se que a conversão de coordenadas no sistema homogêneo para o sistema cartesiano é obtido pela divisão dos três primeiros elementos pelo quarto no sistema homogêneo.

Um ponto no sistema de coordenadas do mundo pode ser reescrito sob a forma vetorial como

$$\omega = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} \quad (\text{II.49})$$

e sua forma homogênea é

$$\omega_h = \begin{bmatrix} kX_p \\ kY_p \\ kZ_p \\ k \end{bmatrix} \quad (\text{II.50})$$

Um ponto no sistema de coordenadas da imagem é dado por

$$i = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} \quad (\text{II.51})$$

Pode-se, portanto, escrever sua forma homogênea considerando-se (II.47) e (II.48), o que resulta

$$i_h = \begin{bmatrix} \frac{f}{Z_p} X_p \\ \frac{f}{Z_p} Y_p \\ \frac{f}{Z_p} Z_p \\ k \end{bmatrix} \quad (\text{II.52})$$

Agora, pode-se obter a matriz de transformação que relaciona ω_h e i_h , da forma

$$i_h = P\omega_h \quad (\text{II.53})$$

fazendo-se $k = f/Z_p$ em (II.50). Portanto, a matriz de transformação obtida para a configuração da Figura (II.15) é

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \quad (\text{II.54})$$

A matriz obtida em (11.54) serve para o caso da Figura 11.15, ou seja, para os casos onde os sistemas de coordenadas do mundo e da imagem estejam alinhados por um dos eixos, no caso o eixo Z. Na próxima seção, apresenta-se o uso da transformação de perspectiva para navegação de robôs móveis considerando-se os movimentos da câmera.

II.7.1 Equações da Transformação de Perspectiva considerando os

Movimentos da Camera

Em relação ao movimento, a câmera pode mover-se na horizontal e na vertical. Ao movimento na horizontal chama-se *pan*, e ao movimento na vertical, *tilt*.

O sistema de interesse na aplicação de transformação de perspectiva em guiamento de robôs móveis é conforme a Figura II.16:

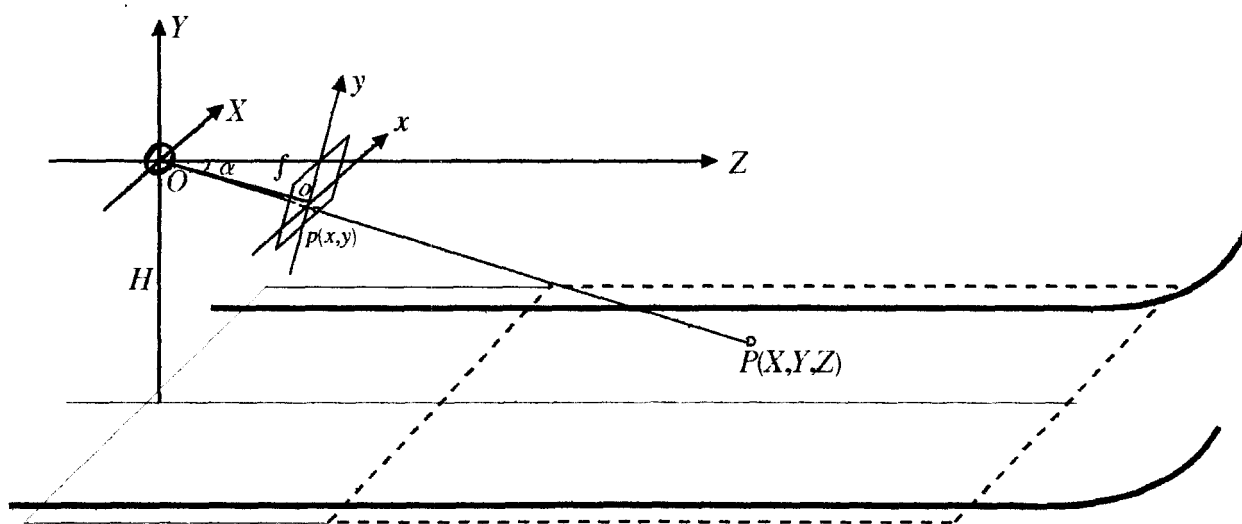


Figura II.16. Configuração dos sistemas de coordenadas para navegação do robô móvel.

Na Figura II. 16, tem-se a origem do sistema de coordenadas do mundo 0 colocada no centro óptico das lentes da câmera que acompanha o veículo. A distância entre 0 e a origem do sistema de coordenadas da imagem o é a distância focal f , e α é o ângulo *de tilt* da câmera em torno de 0, isto é, o ângulo de inclinação vertical da câmera. O ângulo de inclinação horizontal, *pan*, não é mostrado na figura, mas é representado nos cálculos por y . Finalmente, H representa a altura do centro óptico da lente em relação à superfície de navegação.

II.7.1.1 Modelagem da Camera considerando-se Movimentos de *Pan e Tilt*

Para a configuração da Figura 11.16, já não se pode mais usar as Equações (II.47) e (II.48) para se obter informações de coordenadas no mundo a partir de coordenadas da imagem, nem vice-versa, dado que agora se dispõe de um modelo mais complexo. Há de se considerar, adicionalmente ao modelo proposto, movimentos de rotação da câmera nas duas direções: *pan e tilt*.

As equações de transformação de perspectiva para o modelo da Figura II.16 podem ser obtidas através do conhecimento das matrizes de rotação em relação aos eixos X e Y, conforme [12], [13].

A rotação vertical de um ponto em torno do eixo X (*tilt*), de um ângulo α , é obtida através da matriz de transformação R_α ,

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (II.55)$$

A rotação horizontal de um ponto em torno do eixo Y (*pan*), de um ângulo y , no sentido anti-horário, é obtida através da matriz de transformação R_y ,

$$R_\gamma = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma & 0 \\ 0 & 1 & 0 & 0 \\ \sin \gamma & 0 & \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.56)$$

As duas relações de rotação R_α e R_γ podem ser concatenadas em uma única matriz, considerando sequencialmente *tilt* e *pan*, $R = R_\alpha R_\gamma$, resultando

$$R = \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma & 0 \\ \sin \gamma \sin \alpha & \cos \alpha & \sin \alpha \cos \gamma & 0 \\ \sin \gamma \cos \alpha & -\sin \alpha & \cos \alpha \cos \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11.57)$$

Agora, pode-se obter a expressão que relaciona as coordenadas da imagem às coordenadas do mundo, inserindo-se a Equação (II.57) em (II.53), obtendo-se

$$i_h = PRw_h \quad (II.58)$$

que resulta em

$$i_h = \begin{bmatrix} kX_p \cos \gamma - kZ_p \sin \gamma \\ kX_p \sin \gamma \sin \alpha + kY_p \cos \alpha + kZ_p \sin \alpha \cos \gamma \\ kX_p \sin \gamma \cos \alpha - kY_p \sin \alpha + kZ_p \cos \alpha \cos \gamma \\ \frac{1}{f} (kX_p \sin \gamma \cos \alpha - kY_p \sin \alpha + kZ_p \cos \alpha \cos \gamma) \end{bmatrix} \quad (II.59)$$

Portanto, para se obter as coordenadas $(x \ y \ z)$ divide-se os tres primeiros elementos de (II.59) pelo último, com $(X_p \ Y_p \ Z_p) = (X \ Y \ Z)$ para simplificar a notação, obtendo-se

$$x = f \frac{X \cos \gamma - Z \sin \gamma}{X \sin \gamma \cos \alpha - Y \sin \alpha + Z \cos \alpha \cos \gamma} \quad (II.60)$$

$$y = f \frac{X \sin \gamma \sin \alpha + Y \cos \alpha + Z \sin \alpha \cos \gamma}{X \sin \gamma \cos \alpha - Y \sin \alpha + Z \cos \alpha \cos \gamma} \quad (II.61)$$

$$z = f \quad (II.62)$$

As Equações (II.60), (II.61) e (II.62) fornecem a transformada de perspectiva direta, que relaciona os pontos do mundo com os pontos na imagem. Estas equações são obtidas para o caso geral, conforme dito anteriormente, considerando-se a liberdade da câmera rotacionar em duas direções (*pan e tilt*). Note-se que a Equação (II.62) representa a terceira coordenada da imagem, que é constante e igual à distância focal f , o que é óbvio, dado que a imagem é uma representação bidimensional.

De (II.60) e (II.61) pode-se obter a relação inversa que mapeia as coordenadas da imagem para o mundo. A transformação inversa de perspectiva traz consigo uma questão: como obter a partir do espaço bidimensional informação sobre o espaço tridimensional?

Como resposta, aponta-se que há a necessidade de se conhecer uma das três coordenadas no espaço ($X \ Y \ Z$).

Na aplicação para navegação de robôs móveis, conforme a Figura II.16, e levando-se em conta que se trata de uma camera monocular, sabe-se o valor de Y , que é igual à distância do centro óptico da lente à superfície de navegação. Portanto, com o conhecimento desta coordenada, é possível se obter a transformada de perspectiva inversa, que mapeia pontos do sistema da imagem para o mundo¹.

De (II.60) e (II.61) são obtidas as equações para a transformação de perspectiva inversa,

$$Z = Y \frac{y \sin \alpha \cos \gamma - x \sin \gamma + f \cos \alpha \cos \gamma}{y \cos \alpha - f \sin \alpha} \quad (II.63)$$

¹ Notar que a unidade de (x,v,z) é dada em *mm*, devido à transformação considerando a dimensão nominal do *CCD* da câmera e o número de pixels na imagem. Conseqüentemente, a unidade de (X,Y,Z) também é dada em *mm*.

$$X = \frac{-xY \sin \alpha + Z(x \cos \alpha \cos \gamma + f \sin \gamma)}{-x \sin \gamma \cos \alpha + f \cos \gamma} \quad (\text{II.64})$$

II.7.1.2 Modelagem da Camera considerando-se apenas Movimento de *Tilt*

Este caso particular é importante, pois o *setup* para navegação e controle de robô móvel pode ser montado considerando-se a câmera monocular fixa, inclinada para baixo de alguns graus e sem inclinação na horizontal, isto é voltada para a direção do movimento. As equações básicas são obtidas a partir de (II.60), (II.61), (II.63) e (II.64) simplesmente fazendo-se $\gamma = 0$.

As equações resultantes para a transformação de, perspectiva direta são

$$x = f \frac{X}{Z \cos \alpha - Y \sin \alpha} \quad (\text{II.65})$$

c

$$y = f \frac{Y \cos \alpha + Z \sin \alpha}{Z \cos \alpha - Y \sin \alpha} \quad (\text{II.66})$$

e as equações para a transformação de perspectiva inversa são

$$Z = Y \frac{y \sin \alpha + f \cos \alpha}{y \cos \alpha - f \sin \alpha} \quad (\text{II.67})$$

e

$$X = Y \frac{x}{y \cos \alpha - f \sin \alpha} \quad (\text{II.68})$$

onde $Y = -H$ indica a elevação da câmera em relação à superfície de navegação.

II.8 Determinação da Postura Relativa do Robô em Relação a Guia de Navegação

Para fins de controle é necessário se conhecer a posição relativa do móvel em relação a uma guia de navegação ou obstáculo. Para tanto, determina-se a inclinação do robô em relação à guia e também seu desvio lateral em relação a mesma.

II.8.1 Determinação do Ângulo de Inclinação

A Figura II.17 mostra a situação para o cálculo do ângulo de inclinação (δ) do robô em relação à parábola que representa a guia de navegação.

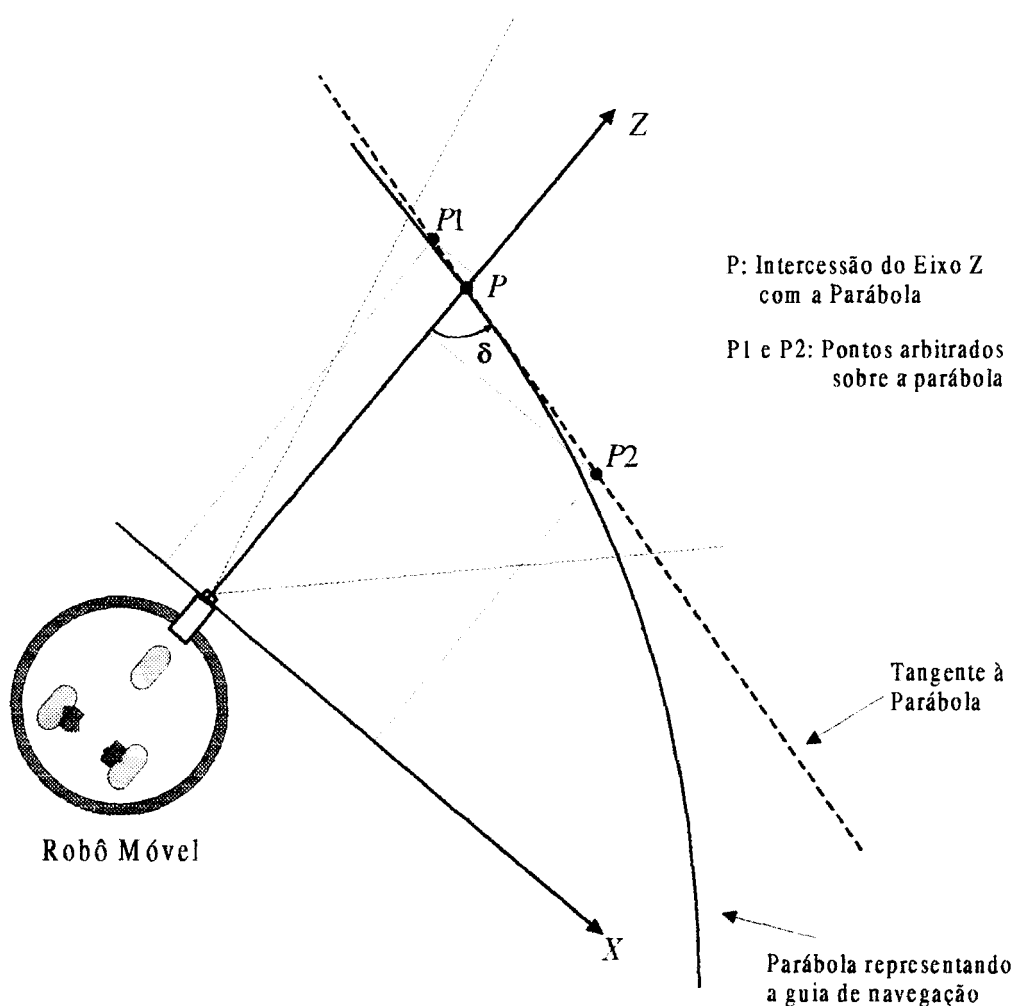


Figura II.17. Configuração para o cálculo do ângulo de inclinação (δ) do robô em relação à guia de navegação.

Supõe-se que a cada nova imagem processada um cenário semelhante ao da Figura II.17 seja formado. A parábola representa a guia de navegação e a linha tracejada representa a reta tangente à parábola no ponto P, que é exatamente o ponto onde o eixo Z intercepta a guia. Para obtenção do ângulo de inclinação, procede-se conforme os seguintes passos:

1. Determinação, na imagem, do ponto onde a parábola intercepta o eixo y. Para a imagem, o sistema de coordenadas é formado pelo par ordenado (x, y).
2. Cálculo da derivada da parábola neste ponto, o que dá a inclinação da reta tangente à parábola naquele ponto. A equação que dá a inclinação da reta tangente à parábola é

$$m = 2a_1 y + a_2 \quad (\text{II.69})$$

onde m é a inclinação da tangente à parábola em P , a_1 e a_2 são o primeiro e segundo coeficientes da parábola, respectivamente, obtidos através de métodos recursivos da Seção II.4.

Com o conhecimento da inclinação e de um ponto, traça-se a reta, representada pela linha tracejada na Figura II.17.

3. A partir do conhecimento da reta, toma-se dois pontos sobre ela, que são mapeados em $P1(X1, Z1)$ e $P2(X2, Z2)$ através da transformação de perspectiva, como em (II.67) e (II.68), determina-se o ângulo de inclinação segundo

$$\delta = \arctan\left(\frac{X2 - X1}{Z2 - Z1}\right) \quad (\text{II.70})$$

II.8.2 Determinação do Desvio Lateral

A Figura II.18 mostra a situação para a determinação do desvio lateral (d) do robô em relação à parábola que representa a guia de navegação.

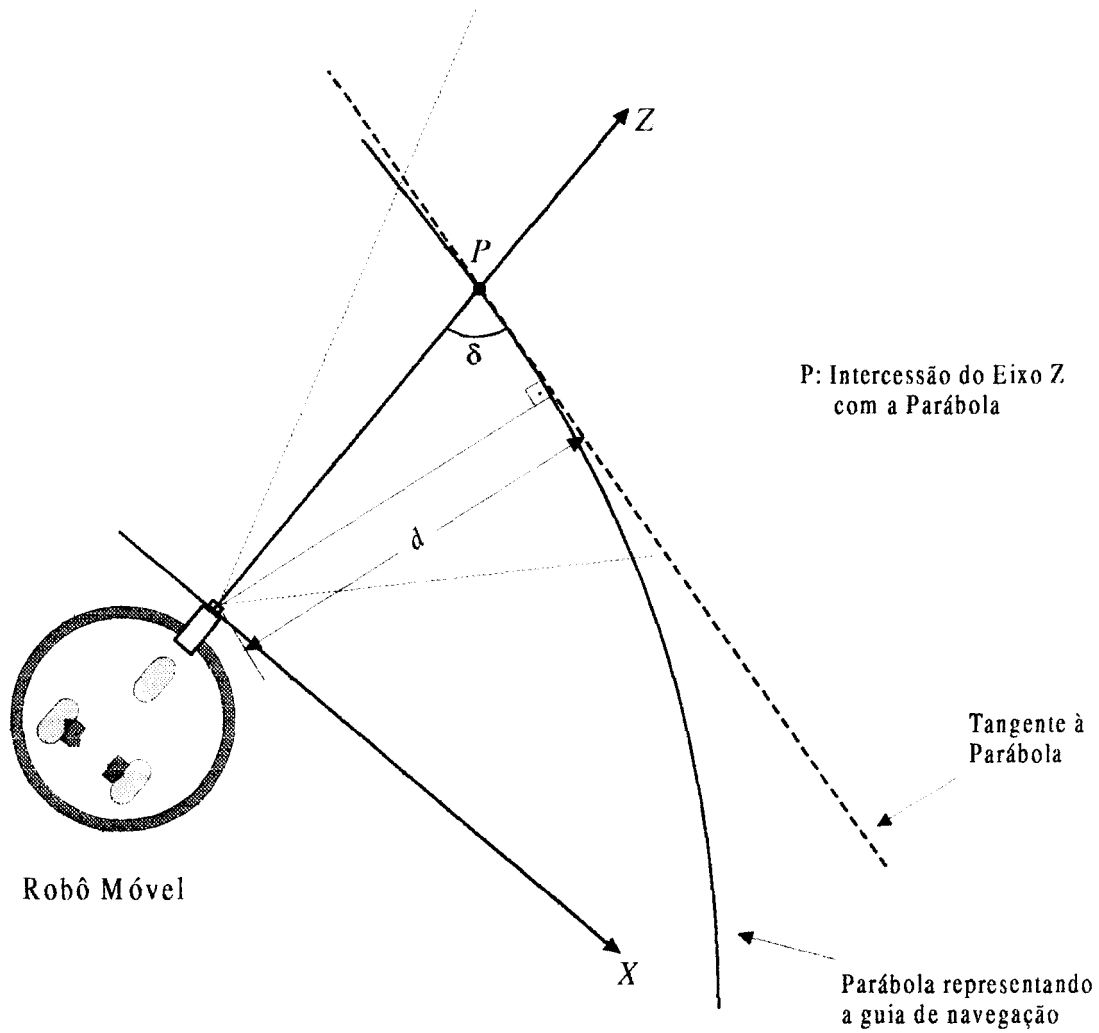


Figura II.18. Configuração para o cálculo do desvio lateral (d) do robô em relação à guia de navegação.

Para se determinar o desvio lateral basta calcular a distância do ponto P ao centro óptico da lente O , rebatido na posição $Y = -H$, dado por $|\vec{OP}|$, obtido no procedimento anterior, da origem do sistema de coordenadas $(X \ Y \ Z)$ localizado no robô móvel. A partir disto, pela Figura II. 18, tem-se

$$d = |\vec{OP}| \sin(\delta) \quad (\text{II.71})$$

Com este procedimento, determina-se a postura: (orientação e distância do robô em relação às guias de navegação), podendo-se assim determinar a ação de controle (de regulação) adequada para sua navegação.

Capítulo III

Guiamento de Robôs Móveis Utilizando Fluxo Óptico

III.1 O Fluxo Óptico

A análise do movimento é um componente importante envolvendo visão computacional, que lida com medidas de movimento de uma sequência de imagens variantes no tempo. A análise de sequências de imagens dinâmicas encontra um grande número de aplicações, incluindo guiamento de robôs [7], [9].

O método de análise de movimento considerado neste capítulo é o baseado no fluxo óptico. Fluxo óptico é uma técnica que permite obter informações sobre o movimento relativo de um objeto e uma câmera, baseando-se em uma sequência de imagens que são adquiridas a uma taxa constante. Conforme [22], o fluxo óptico é definido como a distribuição de velocidades aparentes do movimento dos padrões de brilho em uma imagem e surge do movimento relativo entre objeto e observador. Consequentemente, ele pode dar importantes informações sobre a disposição espacial dos objetos vistos na cena e sobre a taxa de mudança desta disposição.

Mais especificamente, o fluxo óptico é caracterizado por um par (u, v) , onde u é a velocidade do movimento na direção x e v a velocidade do movimento na direção y na imagem, e valores específicos de x e y representam os pixels na imagem.

O conceito do fluxo óptico está intrinsecamente relacionado com sua equação de restrição:

Seja $E(x, y, t)$ a irradiância (ou padrão de brilho) no instante t do ponto (x, y) da imagem. Seja $u(x, y)$ e $v(x, y)$ os componentes xv do vetor do fluxo óptico naquele ponto. Considerando-se que a irradiância seja a mesma no instante $t + \delta t$ no ponto $(x + \delta x, y + \delta y)$ para pequenos intervalos de tempo δt , tem

$$E(x, y, t) = E(x + \delta x, y + \delta y, t + \delta t) \quad (\text{III.1})$$

Considerando-se que o campo de movimento seja contínuo a maior parte do tempo, pode-se expandir o lado direito da Equação III.1 em uma série de Taylor em torno de (x, y, t) e obter-se

$$E(x + \delta x, y + \delta y, t + \delta t) = E(x, y, t) + \delta x \frac{\partial E}{\partial x} + \delta y \frac{\partial E}{\partial y} + \delta t \frac{\partial E}{\partial t} + O(\delta t, \delta x, \delta y) \quad (\text{III.2})$$

onde $O(\delta t, \delta x, \delta y)$ contém os termos de segunda ordem e ordens mais elevadas na série de Taylor. Estes termos podem ser negligenciados para $\delta t, \delta x$ e δy muito pequenos. Portanto, negligenciando-se $O(\delta t, \delta x, \delta y)$, cancelando-se termos comuns, dividindo-se por δt e tomando o limite a medida que $\delta t \rightarrow 0$, obtém-se

$$\frac{\partial E}{\partial x} \frac{dx}{dt} + \frac{\partial E}{\partial y} \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \quad (\text{III.3})$$

Usando-se a notação $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$, $E_x = \frac{\partial E}{\partial x}$, $E_y = \frac{\partial E}{\partial y}$ e $E_t = \frac{\partial E}{\partial t}$, escreve-se

(III.3) como

$$E_x u + E_y v + E_t = 0 \quad (\text{III.4})$$

A Equação (III.4) é conhecida como equação de restrições do fluxo óptico, onde E_x , E_y e E_t são estimados diretamente da imagem.

Métodos distintos podem ser usados na determinação do fluxo óptico. O mais popular deles é o método baseado em gradiente, como na Equação (III.4), proposto por Horn e Schunck, pois são fundamentados na determinação do gradiente da imagem original nas direções x , y e t . Neste caso, considera-se pelo menos dois quadros distintos em uma sequência de imagens. Outros métodos são: métodos baseados em correlação, métodos baseados em energia e métodos baseados em fase. Em [4], estudaram-se vários destes métodos propostos na literatura e reportaram-se os resultados obtidos a partir de um conjunto de imagens sintéticas e reais. Os três algoritmos de melhor desempenho foram: o algoritmo de Horn e Schunck [22], uma versão modificada deste algoritmo [4] e o método baseado em fase [4]. Adicionalmente, um novo método baseado em gradiente foi reportado em [33] exibindo robustez e eficiência. Ao contrário do método de Horn e Schunck que supõe padrão de brilho diferenciável (reflectância varia suavemente), em [33] a descontinuidade no padrão de brilho da imagem é tratada utilizando-se convolução do Laplaciano do filtro Gaussiano em 2D com a imagem. O inconveniente neste método é seu elevado custo computacional.

Nesta tese, optou-se por estudar o comportamento e resultados de um algoritmo baseado em gradiente [46], que é uma variante do método de Horn e Schunck. Esta variante apresenta vantagens sobre o método de Horn e Schunck original no que se refere a maior imunidade a ruídos na imagem, menor custo computacional e maior confiabilidade nos resultados verificados na implementação.

Na Seção III.2 apresenta-se o método original de Horn e Schunck, na Seção III.3 a abordagem variante deste método, baseada em [46], com resultados e discussões sendo

apresentados na Seção III.4. Nas Seções III.5 a III.8 relaciona-se o fluxo óptico e as equações de movimento, resultando em um método para determinação da posição e orientação do robô móvel, e finalmente, na Seção III.9, as conclusões e comentários.

III.2 O Algoritmo de Horn e Schunck Original

A essência deste algoritmo supõe que o padrão de brilho da imagem é quase constante como expresso na Equação (III.4). Todavia, esta equação, por si só, não é suficiente para se determinar u e v para cada pixel, dado que se possui apenas uma equação e duas incógnitas, o que levaria a infinitas soluções. Isto significa que uma restrição adicional precisa ser incluída. O método de Horn e Schunck [20], [22] usa a restrição de que a soma dos Laplacianos de u e v devam ser minimizados (conhecida por restrição de suavidade).

Isto significa que

$$\nabla^2 u + \nabla^2 v \quad (III.5)$$

deva ser minimizada, onde

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \quad (III.6)$$

e

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \quad (III.7)$$

Para facilitar a implementação computacional, as Equações (III.6) e (III.7) são, respectivamente, aproximadas por

$$\nabla^2 u = 3(\bar{u} - u) \quad (III.8)$$

e

$$\nabla^2 v = 3(\bar{v} - v) \quad (III.9)$$

onde \bar{u} e \bar{v} são os valores médios dos componentes do fluxo óptico tomados em um quadrado de 3x3 pixels, centrados no pixel em consideração, e u e v o próprio valor dos componentes do fluxo neste pixel. As médias locais de \bar{u} e \bar{v} são definidas por

$$\begin{aligned}\bar{u}_{i,j,t} = & \frac{1}{6}(u_{i-1,j,t} + u_{i,j+1,t} + u_{i+1,j,t} + u_{i,j-1,t}) \\ & + \frac{1}{12}(u_{i-1,j-1,t} + u_{i-1,j+1,t} + u_{i+1,j+1,t} + u_{i+1,j-1,t})\end{aligned}\quad (\text{III.10})$$

e

$$\begin{aligned}\bar{v}_{i,j,t} = & \frac{1}{6}(v_{i-1,j,t} + v_{i,j+1,t} + v_{i+1,j,t} + v_{i,j-1,t}) \\ & + \frac{1}{12}(v_{i-1,j-1,t} + v_{i-1,j+1,t} + v_{i+1,j+1,t} + v_{i+1,j-1,t})\end{aligned}\quad (\text{III.11})$$

onde os sub-índices i, j e t representam, na imagem, a direção y , a direção x e o tempo, respectivamente.

Além das estimativas de \bar{u} e \bar{v} , deve-se estimar as derivadas do padrão de brilho de um conjunto de imagens. Usando discretização da imagem, E_x, E_y e E_t são as estimativas das taxas de variação do brilho na imagem com respeito a x, y e t no ponto (i, j) . Em [221], usa-se uma aproximação para o cálculo de E_x, E_y e E_t , considerando-se um ponto no centro de um cubo formado por oito medidas. Esta aproximação é expressa pelas equações

$$\begin{aligned}E_x \approx & \frac{1}{4}(E_{i,j+1,t} - E_{i,j,t} + E_{i+1,j+1,t} - E_{i+1,j,t} + E_{i,j+1,t+1} - E_{i,j,t+1} \\ & + E_{i+1,j+1,t+1} - E_{i+1,j,t+1})\end{aligned}\quad (\text{III.12})$$

$$\begin{aligned}E_y \approx & \frac{1}{4}(E_{i+1,j,t} - E_{i,j,t} + E_{i+1,j+1,t} - E_{i,j+1,t} + E_{i+1,j,t+1} - E_{i,j,t+1} \\ & + E_{i+1,j,t+1} - E_{i,j,t+1})\end{aligned}\quad (\text{III.13})$$

$$E_t \approx \frac{1}{4}(E_{i,j,t+1} - E_{i,j,t} + E_{i+1,j,t+1} - E_{i+1,j,t} + E_{i,j+1,t+1} - E_{i,j+1,t} + E_{i+1,j+1,t+1} - E_{i+1,j+1,t}) \quad (\text{III.14})$$

De posse das informações acima, Horn e Schunck propuseram a seguinte solução iterativa para determinação os valores de u e v , iniciando-se de valores arbitrários,

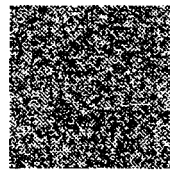
$$u^{n+1} = \bar{u}^n - E_x \frac{E_x \bar{u}^n + E_y \bar{v}^n + E_y}{\alpha^2 + E_x^2 + E_y^2} \quad (\text{III.15})$$

e

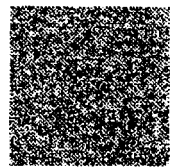
$$v^{n+1} = \bar{v}^n - E_y \frac{E_x \bar{u}^n + E_y \bar{v}^n + E_y}{\alpha^2 + E_x^2 + E_y^2} \quad (\text{III.16})$$

onde α é um fator de ponderação, $n + 1$ é a iteração atual e n é a iteração anterior. Este procedimento demanda um certo número de iterações que depende dos valores iniciais.

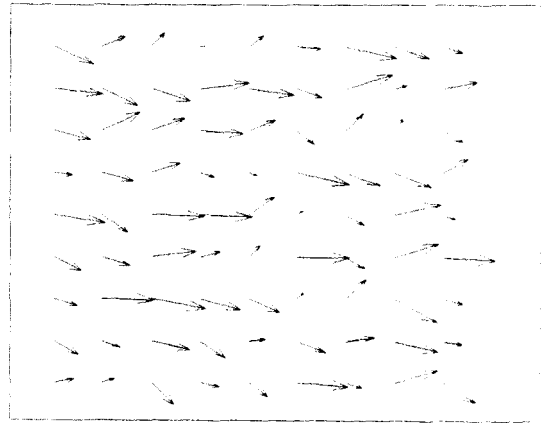
A Figura III.1 exemplifica o algoritmo de Horn e Schunck, para $\alpha = 1.0$.



(a)



(b)



(c)

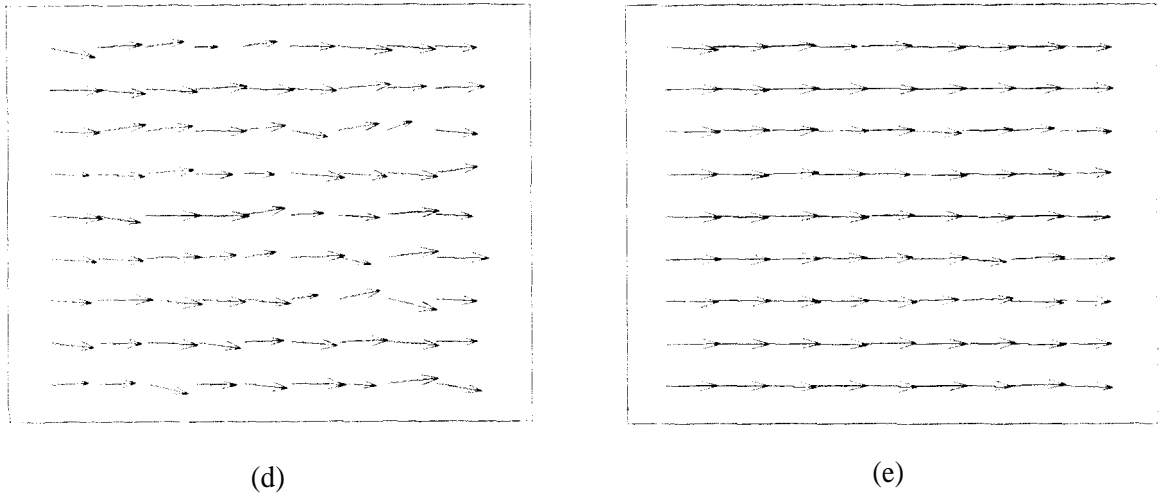


Figura III.1. Procedimento para estimação do fluxo óptico utilizando o algoritmo de Horn e Schunck.

Na Figura III.1, foram considerados dois quadros de uma imagem sintética de dimensão 100x100 pixels. As imagens foram geradas através de uma distribuição uniforme de números randômicos. A primeira imagem está na Figura III.1(a). Na Figura III.1(b), todos os pixels da imagem foram deslocados uma unidade para a direita. As Figuras III.1(c), III.1(d) e III.1(e) mostram o fluxo óptico estimado pelo método de Horn e Schunck para 2, 5 e 10 iterações, respectivamente. Uma vez que o deslocamento imposto é homogêneo em toda a imagem, o campo vetorial estimado converge para o resultado correto, como esperado.

III.3 Um Algoritmo Alternativo para Estimação do Fluxo Óptico

O algoritmo a ser apresentado baseia-se em [46) e considera, como na Seção III.2, que a cena a ser imageada é relativamente suave e não há oclusão, portanto, que o fluxo óptico resultante dela também é relativamente suave. Outra hipótese considerada, é que a imagem pode ser particionada em regiões de distâncias aproximadamente constantes, implicando que o fluxo óptico dentro de cada região seja constante. Esta hipótese, como foi verificado nesta

tese, apesar de mais geral do que a apresentada na Seção III.2, que considera cada pixel na imagem, apresenta algumas vantagens no que tange à robustez do algoritmo em relação a ruídos na imagem, maior velocidade computacional e resultados mais confiáveis, como foi verificado na prática.

Considerando-se o fluxo óptico constante para cada região na imagem, tem-se apenas duas incógnitas, porém, um grande número de equações em cada região, que podem ser resolvidas aplicando-se o método de minimização por mínimos quadrados para a restrição de brilho constante. Basicamente, isto significa resolver a seguinte equação,

$$E = \underset{u,v}{\operatorname{ArgMin}} \sum_{x,y,t} (E_x u + E_y v + E_t)^2 \quad (\text{III.17})$$

para u e v , onde todas as variáveis são conforme a Equação (III.4), e a soma se dá em todos os pixels em cada região de tamanho constante. Dimensionando-se adequadamente o tamanho das regiões, não há perda de generalidade, dado que o mapeamento das informações na imagem para o mundo pode ser feito sem perda de acurácia dimensionando-se o espaço de trabalho a ser considerado para guiamento do robô móvel. A utilização de regiões de tamanho constante tem o benefício extra de não requerer filtragem ou média das estimativas antes do mapeamento para o mundo.

Desenvolvendo-se a Equação (III.17) e expandido-se o somatório, considerando-se o critério de minimização, obtém-se as equações:

$$u \sum E_x^2 + v \sum E_x E_y + \sum E_x E_t = 0 \quad (\text{III.18})$$

e

$$u \sum E_x E_y + v \sum E_y^2 + \sum E_y E_t = 0 \quad (\text{III.19})$$

As equações lineares (III. 18) e (III.19) podem ser resolvidas simultaneamente através de vários métodos. Uma opção é usar o método dos determinantes, que resulta em

$$u = \frac{\begin{vmatrix} -\sum E_x E_t & \sum E_x E_y \\ -\sum E_y E_t & \sum E_y^2 \end{vmatrix}}{\begin{vmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{vmatrix}} \quad (III.20)$$

c

$$v = \frac{\begin{vmatrix} \sum E_x^2 & -\sum E_x E_t \\ \sum E_x E_y & -\sum E_y E_t \end{vmatrix}}{\begin{vmatrix} \sum E_x^2 & \sum E_x E_y \\ \sum E_x E_y & \sum E_y^2 \end{vmatrix}} \quad (III.21)$$

que também podem ser expressas, respectivamente como

$$u = \left(\frac{\sum E_x E_y \sum E_y E_t - \sum E_x E_t \sum E_y^2}{\sum E_x^2 \sum E_y^2 - \sum E_x E_y \sum E_x E_y} \right) \quad (III.22)$$

c

$$v = \left(\frac{\sum E_x E_y \sum E_x E_t - \sum E_x^2 \sum E_y E_t}{\sum E_x^2 \sum E_y^2 - \sum E_x E_y \sum E_x E_y} \right) \quad (III.23)$$

onde E_x , E_y e E_t são conforme (III.12), (III.13) e (III.14).

Na implementação do algoritmo em tempo real, considera-se imagens de dimensão 320x240 pixels. Todavia trabalha-se com uma janela de processamento, que despreza pequenas regiões nas bordas das imagens. Isto é justificável, dado que nas bordas tem-se risco maior de distorção e ruídos, além do que, com este procedimento, reduz-se o tempo de computação. O fluxograma deste algoritmo é mostrado na Figura III.2.

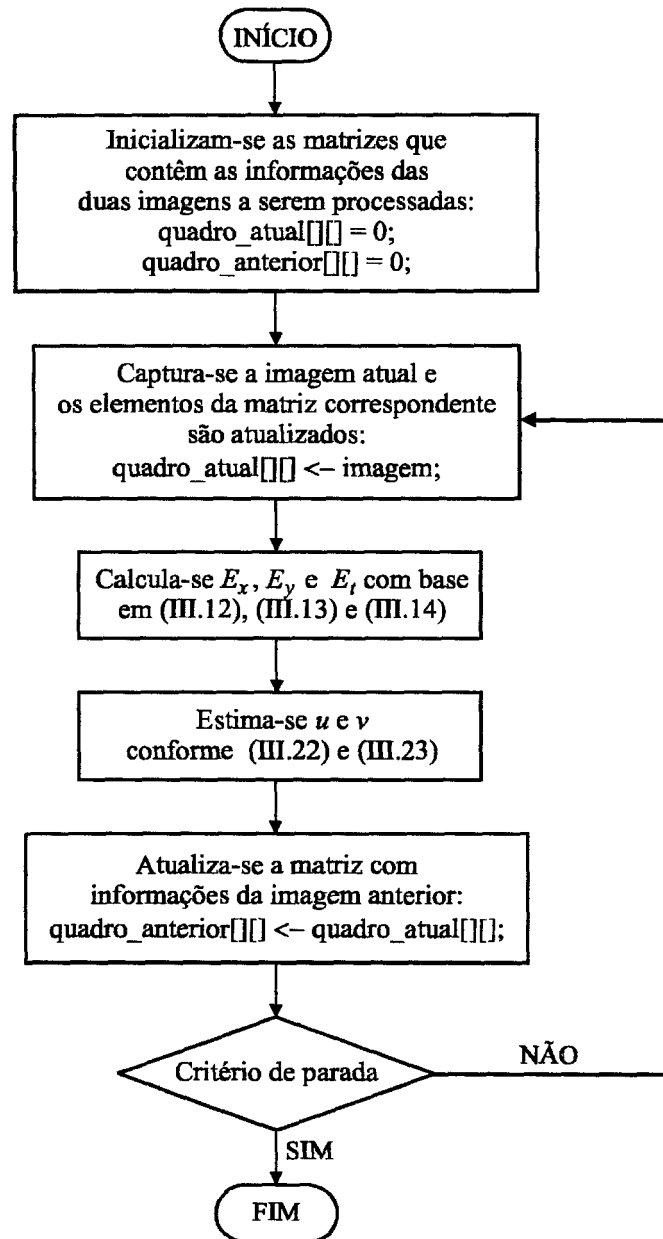


Figura III.2. Fluxograma do algoritmo de estimação do fluxo óptico.

III.4 Resultados Experimentais e Discussões

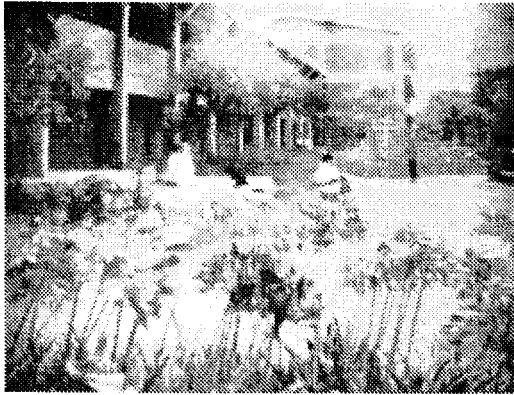
As mesmas plataformas de *hardware e software* usadas na Seção III.3 foram usadas neste experimento. O algoritmo descrito na Seção III.3 foi implementado em tempo real em linguagem C para LINUX, no compilador *GCC*. A seguir apresentam-se alguns resultados e discussões para duas situações distintas: a) uso de uma cena estática (uma gravura disponível no Laboratório de Visão Computacional do ITA) e b) utilização de uma seqüência de imagens em ambiente interno, tiradas do corredor do Divisão de Engenharia Eletrônica do ITA. Em todos os casos, utilizou-se partição da imagem em blocos de dimensão 10 x 10 pixels como as regiões de tamanho constante em (III.17).

Para o caso da imagem estática, utilizou-se o seguinte arranjo: fixando-se a câmera em uma dada posição frontal à gravura, obteve-se a primeira imagem. Depois, movendo-se a câmera lentamente em várias direções e sentidos, analisou-se o fluxo óptico resultante. O resultado mostrou-se bastante satisfatório, como mostrado nas Figuras III.3, III.4 e III.5.

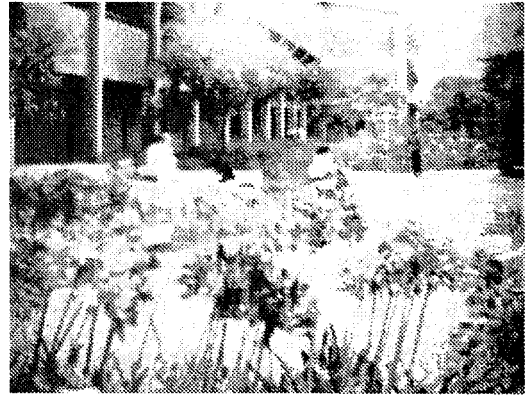
Nota-se no caso da Figura III.3 que o fluxo óptico resultante é orientado para fora. Isto acontece devido ao movimento dominante de translação no sentido da cena. Além dos vetores de fluxo estarem orientados de forma ordenada, percebe-se que eles têm magnitude crescente a partir de um certo ponto. Este ponto, cuja magnitude de fluxo é teoricamente nula, no caso de translação pura, chama-se foco de expansão (*FOE*) [36]. Por definição, o *FOE* é o ponto no plano da imagem do qual o fluxo óptico irradia. Em outras palavras, o *FOE* é a singularidade do fluxo óptico, o que significa que no *FOE*, caracterizado pelo par (x_{FOE}, y_{FOE}) o fluxo óptico é nulo, ou

$$u(x_{FOE}, y_{FOE}) = 0 \text{ e } v(x_{FOE}, y_{FOE}) = 0 \quad (III.24)$$

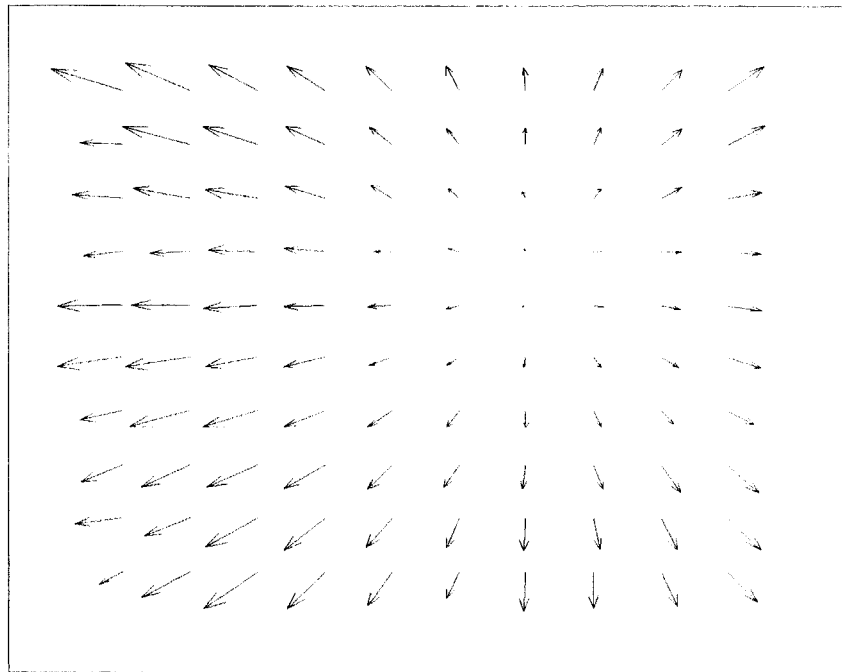
O conhecimento do *FOE* é importante para o caso do guiamento, onde se necessita informações sobre a distância entre o objeto e a câmera.



(a)

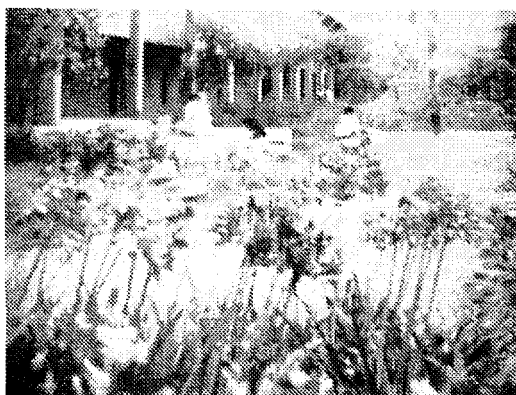


(b)



(c)

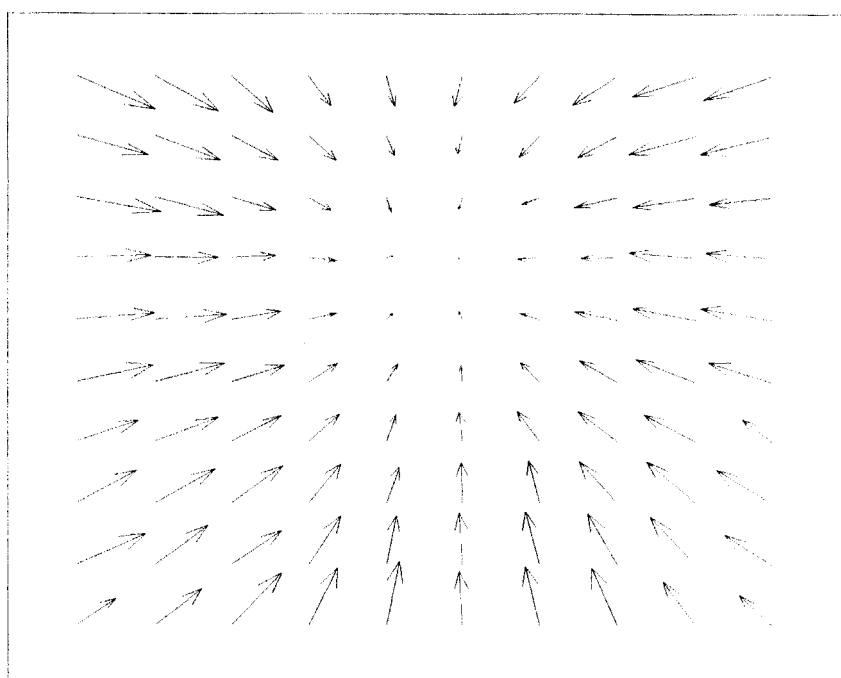
Figura III.3. Obtenção do fluxo óptico para cena estática utilizando o algoritmo descrito na Seção III.3. A câmera moveu-se para a frente lentamente, aproximando-se da gravura. Em (a) e (b) são mostrados dois quadros escolhidos durante o deslocamento. Em (c), o fluxo óptico resultante.



(a)



(b)



(c)

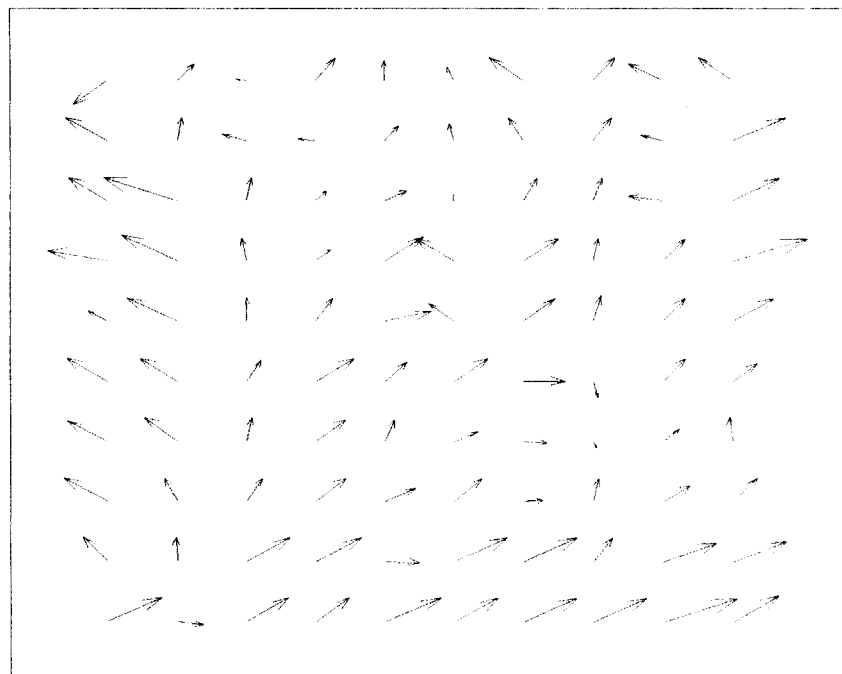
Figura III.4. Obtenção do fluxo óptico para cena estática utilizando o algoritmo descrito na Seção III.3. A câmera moveu-se para trás lentamente, afastando-se da gravura. Em (a) e (b) são mostrados dois quadros escolhidos durante o deslocamento. Em (c), o fluxo óptico resultante.



(a)



(b)



(c)

Figura III.5. Obtenção do fluxo óptico para cenas internas utilizando o algoritmo descrito na Seção III.3. Foi obtida uma seqüência de imagens no corredor da Engenharia Eletrônica do ITA, deslocando-se lentamente a camera para frente. Em (a) e (b) são mostradas duas quadras consecutivas da seqüência. Em (c), o fluxo óptico resultante.

No caso da Figura III.4, nota-se que o sentido do fluxo é invertido em relação à Figura III.3, o que era de se esperar dado que neste caso a câmera está se movendo em sentido contrário ao da cena. Percebe-se que nesta configuração os vetores tendem para o *FOE*. Em ambos os casos, tanto na Figura III.3 quanto na RIA, como as regiões mais a margem da imagem movem-se mais depressa, a magnitude do fluxo é maior. Convém ressaltar que a imagem obtida da gravura apresenta conteúdo com profundidade constante, o que faz com que a estimativa do fluxo óptico via algoritmo da Seção III.3 apresente bons resultados.

Para a Figura III.5, a configuração requer um estudo mais cuidadoso, pois o fluxo óptico resultante não apresenta orientação definida, apesar de as imagens terem sido obtidas em seqüência deslocando-se para frente no corredor. Alguns fatores prejudicam o uso de técnicas diferenciais, como as descritas nas Seções III.2 e III.3, para estimação do fluxo óptico. Três fatores podem ser enumerados:

1. Grandes deslocamentos entre quadros consecutivos: as técnicas diferenciais para estimação do fluxo óptico têm desempenho deteriorado quando se trata de grandes deslocamentos dos pixels entre quadros consecutivos. Os melhores resultados são obtidos quando não há mais que quatro ou cinco pixels de deslocamento. A descontinuidade no cálculo das derivadas temporais introduz erros na estimação do fluxo óptico. Atualmente, com máquinas mais rápidas, consegue-se processar vários quadros por segundo. Em relação ao guiamento de robôs, considerando-se ainda que sua dinâmica de movimentação seja consideravelmente mais lenta que a dinâmica de processamento das imagens para se determinar o fluxo óptico, este problema não é particularmente grave. A qualidade da estimativa do fluxo óptico será degradada, porém, em situações de grande velocidade relativa entre o robô e a cena.
2. Cenas com textura pobre: este pode ser apontado como o maior problema na estimação do fluxo óptico em cenas como a da Figura III.5. A pobreza de textura pode ser

caracterizada como grandes áreas na imagem onde as variações de níveis de cinza são pequenas. No caso da Figura III.5, como as paredes são brancas e lisas, isto se verifica em larga escala. Como é de se esperar, a falta de textura na cena introduz significativa imprevisão no cálculo das derivadas no espaço.

3. Choques e vibrações: em seqüências de imagens, o movimento não é causado única e exclusivamente pelo deslocamento relativo, mas também por choques e vibrações sofridos pelo veículo que transporta a câmera. Tais choques e vibrações podem introduzir transientes grandes e muito rápidos, com deslocamentos globais altos.
4. Violação da hipótese de profundidade constante: a imagem obtida é de um corredor.

Finalmente, o tempo de processamento do algoritmo implementado na Seção III.3 para estimação dos valores de fluxo óptico u e v é de $67.0\ ms$. Este valor é consideravelmente maior do que aqueles apresentados na Tabela III.1 para os algoritmos recursivos.

III.5 O Uso do Fluxo Óptico para Guiamento do Robô Móvel

Uma vez que o fluxo óptico é conhecido, torna-se possível recuperar informação do movimento da câmera com relação a um objeto. Esta informação é bastante significativa para guiamento de robôs móveis, dado que o robô deve ser apto a evitar colisões enquanto navega em ambiente semi-estruturado. Portanto, do fluxo óptico é possível se recuperar parcialmente informação sobre seu ego-movimento, profundidade (donde vem a possibilidade de evitar colisões) e orientação do veículo. Por ego-movimento define-se a velocidade linear instantânea V e a velocidade angular instantânea ω .

III.5.1 Equações do Movimento empregando-se Fluxo Óptico

A Figura III.7 representa a projeção de um ponto do mundo (P_x, P_y, P_z) no plano da imagem (r_x, r_y, f) , onde f representa a distância focal da lente.

Assume-se que a origem do sistema de coordenadas O do sistema $(X \ Y \ Z)$ esteja rigidamente conectada ao centro óptico da lente da câmera. Assume-se ainda que a camera esteja em movimento relativo à cena imageada de um corpo rígido com velocidade translacional \vec{T} e velocidade angular $\vec{\Omega}$.

De forma equivalente, pode-se considerar um corpo rígido estando em movimento com respeito à câmera com velocidade de translação $-\vec{T}$ e velocidade rotacional $-\vec{\Omega}$. Os componentes da velocidades de translação são $(T_x \ T_y \ T_z)$, e da velocidade de rotação $(\omega_x \ \omega_y \ \omega_z)$, conforme mostrados na Figura III.7.

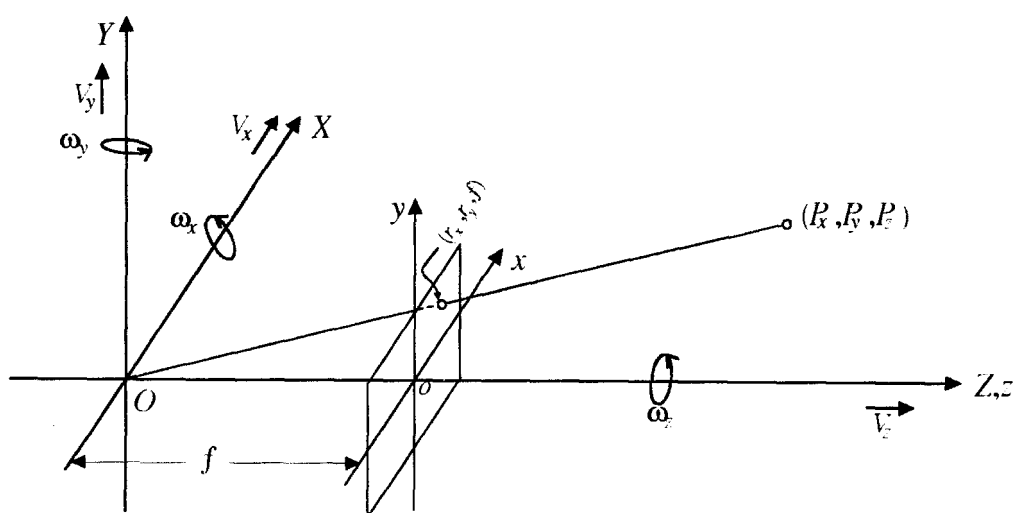


Figura III.6. Geometria para análise das equações do movimento empregando o fluxo óptico.

Portanto, a velocidade relativa instantânea total de qualquer ponto do mundo na cena imageada pode ser expressa pela equação de Coriolis de um sistema de coordenadas em movimento, isto é

$$\dot{\vec{P}} = -\vec{T} - \vec{\Omega} \times \vec{P} \quad (\text{III.25})$$

com $\vec{P} = (P_x \ P_y \ P_z)$, $\vec{T} = (T_x \ T_y \ T_z)$ e $\vec{\Omega} = (\omega_x \ \omega_y \ \omega_z)$.

De (III.25) resulta

$$\dot{P}_x = -T_x - \omega_y P_z + \omega_z P_y \quad (\text{III.26})$$

$$\dot{P}_y = -T_y - \omega_z P_x + \omega_x P_z \quad (\text{III.27})$$

e

$$\dot{P}_z = -T_z - \omega_x P_y + \omega_y P_x \quad (\text{III.28})$$

Considerando-se a transformação de perspectiva, pode-se reescrever as Equações (II.47) e (II.48) segundo a Figura III.7, como

$$r_x = \frac{f}{P_z} P_x \quad (\text{III.29})$$

e

$$r_y = \frac{f}{P_z} P_y \quad (\text{III.30})$$

Portanto, de (III.26) e (III.27), obtém, com base em (III.29) e (III.30)

$$\dot{P}_x = \frac{1}{f} (\dot{P}_z r_x + P_z \dot{r}_x) \quad (\text{III.31})$$

e

$$\dot{P}_y = \frac{1}{f} (\dot{P}_z r_y + P_z \dot{r}_y) \quad (\text{III.32})$$

Substituindo-se (III.29), (III.30), (III.31) e (III.32) em (III.26), obtém-se

$$\dot{r}_x = \frac{-fT_x + T_z r_x}{P_z} + \frac{\omega_x r_x r_y}{f} - \omega_y \left(f + \frac{r_x^2}{f} \right) + \omega_z r_y \quad (\text{III.33})$$

Ou

$$\dot{r}_x = Y_x + \Gamma_x \quad (\text{III.34})$$

onde

$$Y_x = \frac{-fT_x + T_z r_x}{P_z} \quad \text{e} \quad \Gamma_x = \frac{\omega_x r_x r_y}{f} - \omega_y \left(f + \frac{r_x^2}{f} \right) + \omega_z r_y \quad (\text{III.35})$$

Similarmente, substituindo-se (III.29), (III.30), (III.31) e (III.32) em (III.27). obtém-se

$$\dot{r}_y = \frac{-fT_y + T_z r_y}{P_z} - \frac{\omega_y r_x r_y}{f} + \omega_y \left(f + \frac{r_y^2}{f} \right) - \omega_z r_x \quad (\text{III.36})$$

ou

$$\dot{r}_y = Y_y + \Gamma_y \quad (\text{III.37})$$

onde

$$Y_y = \frac{-fT_y + T_z r_y}{P_z} \quad \text{e} \quad \Gamma_y = -\frac{\omega_y r_x r_y}{f} + \omega_y \left(f + \frac{r_y^2}{f} \right) - \omega_z r_x \quad (\text{III.38})$$

As Equações (III.33)-(III.38) são conhecidas por equações do movimento da imagem.

Note-se que apenas o componente translacional do fluxo óptico depende da posição espacial do ponto da cena. O componente rotacional depende apenas da posição da imagem (r_x, r_y, f) .

III.6 Guiamento de Robô Móvel empregando-se o Fluxo Óptico

Considere-se as seguintes restrições ao movimento do robô móvel, semelhantemente a [9]: 1) a câmera deve estar fixa no aparato sem nenhum tipo de inclinação ou grau de liberdade, isto é, sem *pan* ou *tilt* e na direção e sentido do movimento; 2) o movimento seja limitado a um plano horizontal, ou seja, $T_x = T_y = \omega_x = \omega_z = 0$. Considerando-se estas condições em (III.33) e (III.36), obtém-se

$$\dot{r}_x = \frac{T_z r_x}{P_z} - \Omega_x, \text{ onde } \Omega_x = \omega_y \left(f + \frac{r_x^2}{f} \right) \quad (\text{III.39})$$

c

$$\dot{r}_y = \frac{T_z r_y}{P_z} - \Omega_y, \text{ onde } \Omega_y = \frac{\omega_x r_x r_y}{f} \quad (\text{III.40})$$

III.7 Recuperação do Ego-movimento a partir do Fluxo Óptico

O modelo do ego-movimento é dado pelas Equações (III.33) e (III.36). A recuperação do ego-movimento requer a estimação dos parâmetros velocidade translacional \vec{T} e velocidade angular $\vec{\Omega}$, a partir das estimativas dos fluxos ópticos e da estrutura da cena. Embora a literatura sobre fluxo óptico seja vasta, aquela referente a recuperação do ego-movimento objetivando aplicação explícita em robótica móvel é limitada, vide por exemplo [7] [9] e [14].

III.7.1 Recuperação da Rotação e da Profundidade Relativa

Para se estimar a rotação do veículo conforme em [9], elimina-se a profundidade relativa T_z/P_z combinando-se as Equações (III.39) e (III.40). A quantidade T_z/P_z é chamada de profundidade relativa por relacionar a velocidade do movimento com a distância da câmera

ao objeto imageado. Esta quantidade tem dimensão de tempo (s) e é também conhecida como tempo de colisão, r , por estimar o tempo de colisão entre o observador e o objeto.

Portanto, eliminando-se a profundidade relativa em (III.39) e (III.40), chega-se a

$$r_y \dot{r}_x - r_x \dot{r}_y + f r_y \omega_y = 0 \quad (\text{III.41})$$

Como se sabe, o fluxo óptico é estimado a partir de uma seqüência de imagens reais, em geral ruidosas. Sejam \ddot{r}_x e \ddot{r}_y os valores reais (que são ruidosos) do fluxo óptico. O critério de mínimos quadrados pode ser usado em (III.41), resultando

$$\tilde{\omega}_y = \frac{1}{f} \frac{\sum_R (r_x r_y \ddot{r}_y - r_y^2 \ddot{r}_x)}{\sum_R (r_y^2)} \quad (\text{III.42})$$

onde $\tilde{\omega}_y$ é a rotação estimada a partir do fluxo, e o sub-índice R no somatório denota a região da imagem utilizada para a estimação do fluxo.

Por sua vez, a profundidade relativa (ou tempo de colisão) pode ser estimada de (III.39) ou (III.40). Considerando-se por exemplo (III.39), tem-se

$$\tau = \frac{P_z}{T_z} = \frac{r_x}{\ddot{r}_x + \tilde{\Omega}_x} \quad (\text{III.43})$$

III.8 Determinação da Postura Relativa para Guiamento do Robô Móvel

Por postura relativa, ou orientação, entende-se a posição relativa do robô em relação a um objeto, que nos casos em ambiente interno geralmente são as paredes ou obstáculos.

Seja o caso onde se queira guiar o robô móvel pelo centro de um corredor reto utilizando-se realimentação visual com processamento das imagens por fluxo óptico. O cenário é semelhante ao da Figura III.7.

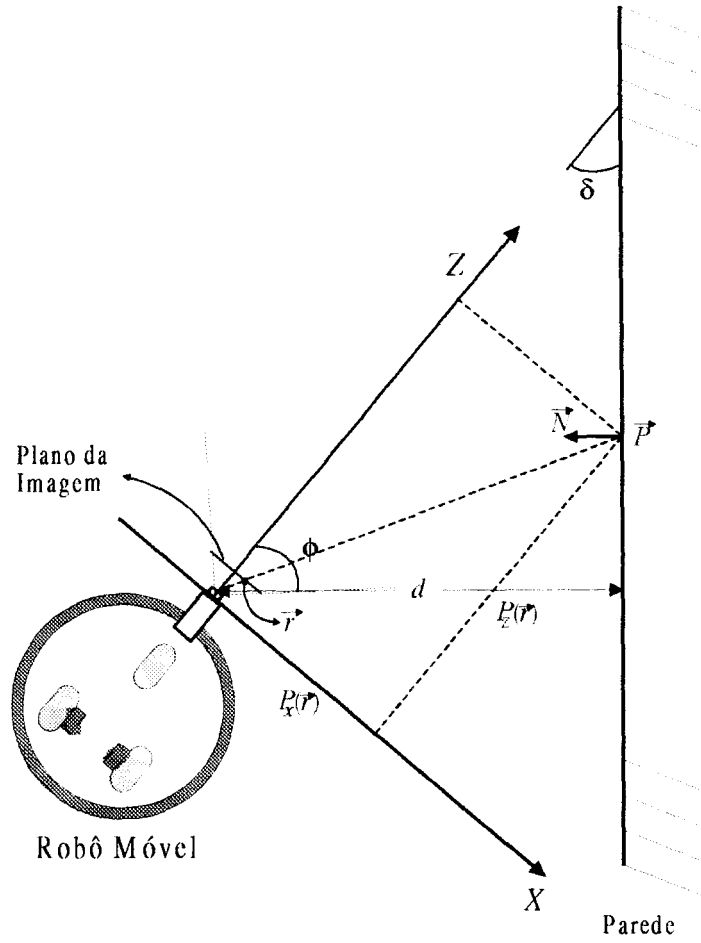


Figura III.8. Configuração para determinação da postura relativa usando fluxo óptico.

Em [9], estuda-se a variação temporal do fluxo óptico em uma região fixa da imagem, isto é,

$$H = \frac{\partial \vec{r}}{\partial t} \Big|_{\vec{r}=cte} \quad (III.44)$$

Derivando-se (III.39) temporalmente, e lembrando-se que \mathbf{r} é constante com o passar do tempo, decorre

$$H_x = (-\dot{r}_x - \Omega_x) \frac{\dot{P}_z}{P_z} \quad (III.45)$$

onde H_x é o componente x de H.

Agora, derivando-se (III.40), obtém H_y , o componente y de H,

$$H_y = (-\dot{r}_y - \Omega_y) \frac{\dot{P}_z}{P_z} \quad (\text{III.46})$$

ou em notação matricial,

$$H = \begin{bmatrix} -\dot{r}_x - \Omega_x \\ -\dot{r}_y - \Omega_y \end{bmatrix} \frac{\dot{P}_z}{P_z} \quad (\text{III.47})$$

A definição em (III.47) é útil na determinação do ângulo ϕ na Figura III.8, e consequentemente, do ângulo δ .

111.8.1 Determinação do Desvio Lateral e do Ângulo de Inclinação

Conforme a Figura III.8, a distância lateral d pode ser facilmente obtida como

$$d(t) = P_z(r) \cos \phi + P_x(r) \sin \phi \quad (\text{III.48})$$

e de (III.29) resulta

$$d(t) = P_z(r) \cos \phi + \frac{r_x P_z(r)}{f} \sin \phi \quad (\text{III.49})$$

O vetor normal $\vec{N} = (N_x \ N_y \ N_z)$ à superfície pode ser decomposto em componentes nas direções de X e Z de acordo com a Figura III.8, isto é,

$$\vec{N} = (-\sin \phi \ 0 \ -\cos \phi) \quad (\text{III.50})$$

onde o sinal negativo denota que o sentido de N é contrário ao do deslocamento. O módulo de \vec{N} é unitário.

Portanto, pode-se escrever

$$\vec{r} \cdot \vec{N} = (r_x \ r_y \ f) \cdot (-\sin \phi \ 0 \ -\cos \phi) = -r_x \sin \phi - f \cos \phi \quad (\text{III.51})$$

e de (III.49) e (III.51) resulta

$$P_z = \frac{-f d(t)}{\vec{r} \cdot \vec{N}} \quad (III.52)$$

derivando-se (III.49) e empregando-se (III.52), segue que

$$\frac{\dot{P}_z}{P_z} = \frac{-f \dot{d}}{P_z \vec{r} \cdot \vec{N}} - \frac{\vec{r} \cdot \dot{\vec{N}}}{\vec{r} \cdot \vec{N}} \quad (III.53)$$

As variações no comportamento do vetor normal \vec{N} e do vetor da distância lateral \vec{d} podem ser expressas em função dos parâmetros do movimento. Dado que o único componente da velocidade de translação atuante no movimento é idealmente a Z, conforme a restrição apresentada na Seção III.6, sua ação pode ser decomposta em dois componentes: a) na direção da normal \vec{N} e b) na direção perpendicular a ela. O componente na direção da normal \vec{N} é responsável pela variação na distância lateral entre o robô e a parede, e portanto, pode-se escrever

$$\dot{d} = \vec{N} \cdot \vec{T} \quad (III.54)$$

Por sua vez, à medida que o robô muda sua orientação, o vetor normal \vec{N} do corredor reto também sofre variação em sua direção no sistema de coordenadas do robô. Esta variação pode ser expressa pela equação

$$\dot{\vec{N}} = \vec{\omega} \times \vec{N} \quad (III.55)$$

O resultado obtido em (III.55) é facilmente verificado derivando-se a Equação (III.50).

Agora, substituindo-se (III.54) e (III.55) em (III.53), tem-se

$$\frac{\dot{P}_z}{P_z} = \frac{-f \vec{N} \cdot \vec{T}}{P_z \vec{r} \cdot \vec{N}} - \frac{\vec{r} \cdot \vec{\omega} \times \vec{N}}{\vec{r} \cdot \vec{N}} \quad (III.56)$$

Utilizando-se (III.56) em (III.47), obtém-se a relação entre a derivada temporal do fluxo óptico e a orientação do robô,

$$H = \begin{bmatrix} \dot{r}_x + \Omega_x \\ \dot{r}_y + \Omega_y \end{bmatrix} \frac{(f \vec{T}/P_z + \vec{r} \times \vec{\omega}) \cdot \vec{N}}{\vec{r} \cdot \vec{N}} \quad (\text{III.57})$$

Desenvolvendo-se o produto vetorial em (III.57), tem-se

$$\vec{r} \times \vec{\omega} = (r_x, r_y, f) \times (0, \omega_y, 0) = (-f\omega_y, 0, r_x\omega_y) \quad (\text{III.58})$$

Portanto, de (III.51), (III.58) e lembrando (III.50), chega-se a

$$H = \begin{bmatrix} \dot{r}_x + \Omega_x \\ \dot{r}_y + \Omega_y \end{bmatrix} \left(\frac{f\omega_y \sin \phi - r_x\omega_y \cos \phi - \frac{fT_z}{P_z} \cos \phi}{-r_x \sin \phi - f \cos \phi} \right) \quad (\text{III.59})$$

Dividindo-se agora o numerador e o denominador do lado direito de (III.59) por $\cos \phi$ e reordenando a equação de forma a se isolar ϕ , chega-se à orientação

$$\phi = -\arctan \left\{ \frac{fH_x r_x - r_x^2 \omega_y (\dot{r}_x + \Omega_x) - (\dot{r}_x + \Omega_x)^2}{r_x [H_x r_x + f\omega_y (\dot{r}_x + \Omega_x)]} \right\} \quad (\text{III.60})$$

Uma vez que se possui a orientação do veículo, pode-se obter a distância relativa do mesmo à parede. Pela Equação (III.49) não se consegue obter diretamente o valor de d , pois não se conhece $P_z(r)$. Portanto, dividindo-se (III.49) pelo valor da velocidade de translação T_z , resulta

$$\frac{d(t)}{T_z} = \tau \cos \phi + \frac{r_x}{f} \tau \sin \phi \quad (\text{III.51})$$

Como H , \dot{r}_x , Ω_x e τ podem ser estimados a partir da imagem e de sensores auxiliares, a determinação aproximada da orientação e da distância relativa do robô são possíveis, o que permite a aplicação de uma lei de controle adequada para o guiamento deste robô em um ambiente semi-estruturado como o descrito. Convém ressaltar que as estimativas anteriormente citadas são degradadas por ruídos, havendo pelo menos duas maneiras de se

atenuar esta dificuldade: a) utilização do conhecimento da estrutura da imagem (por exemplo, no caso do corredor, para estimação seriam utilizadas regiões nas laterais), b) utilização de um valor médio das estimativas dos parâmetros H , \dot{r}_x , Ω_x e τ em uma certa região.

III.9 Conclusões

Neste capítulo foram apresentados procedimentos para se utilizar fluxo óptico na guiagem de robôs móveis. Há a necessidade de se ter maior domínio sobre o ambiente que cerca o experimento, como distribuição uniforme da iluminação, cenas com textura adequada e processamento das imagens a uma taxa suficiente para evitar grandes deslocamentos entre quadros consecutivos em uma sequência de imagens. A presença de ruídos é também um fator crítico, podendo ser minimizado por algoritmos como o apresentado na Seção III.3 ou através de métodos de filtragem da imagem.

Os resultados para atualização da estimativa do fluxo óptico foram obtidos a uma taxa de 10 Hz . Comparando-se os resultados desta abordagem com os apresentados no Capítulo II, percebeu-se um maior custo computacional envolvido na estimativa do fluxo óptico, além de uma maior dependência do ambiente do experimento. Todavia, o método do fluxo óptico apresenta a estimativa de tempo de colisão e detecção de obstáculos, o que torna o método atraente em termos qualitativos.

Capítulo IV

Controle de Robôs Móveis

IV.1 Introdução

Os robôs móveis pertencem a uma importante classe de sistemas, os sistemas não-holonômicos. Estes sistemas são caracterizados pelo fato de que há restrições não integráveis impostas ao movimento. Mais precisamente, definindo-se $q \in \mathbb{R}^n$ como sendo o vetor de estado, tem-se restrição da forma $A(q)\dot{q} = 0$, isto é, o vetor de velocidades \dot{q} está restrito ao espaço nulo de $A(q)$ para todo q . Neste trabalho, esta restrição aparece na equação (IV.17).

Para o estudo do controle de robôs móveis é importante se definir o tipo de acionamento do robô e a que classe ele pertence. Existem basicamente três métodos de acionamento para robôs móveis: através de rodas, através de lagartas e através de pernas. O acionamento através de rodas é o mais comum. Além de ser de fácil construção, adapta-se bem às condições de operação em ambientes internos e ambientes externos. A utilização do acionamento através de lagartas ou através de pernas restringe-se à operação em ambientes externos, devido à maior facilidade em transpor obstáculos.

Com relação à classificação dos robôs móveis em classes, os robôs com rodas podem ser divididos em cinco classes diferentes, as classes (3,0), (2,0), (2,1), (1,1) e (1,2). Para detalhes e definições das respectivas classes, vide [32].

O robô usado neste estudo é o *MagellanTM Pro Compact Mobile Robot* da *IS Robotics* [35], com acionamento através de rodas, pertencendo à classe (2,0), pois possui duas rodas fixas em um eixo comum, operando em modo diferencial.

Quatro tipos de modelos podem ser estabelecidos para robô com acionamento através de rodas: a) modelo cinemático de postura, b) modelo cinemático de configuração, c) modelo dinâmico de configuração e d) modelo dinâmico de postura, vide [32] para detalhes.

Os modelos cinemáticos descrevem o robô em função da velocidade e orientação das rodas, enquanto que os modelos dinâmicos descrevem o robô em função das forças generalizadas aplicadas pelos atuadores, como os torques nas rodas.

Os modelos de postura consideram como estado apenas a posição e orientação do robô, ao passo que os modelos de configuração consideram além da postura outras variáveis internas, como o deslocamento angular das rodas. Do ponto de vista de controle da posição e orientação espacial do robô móvel, apenas os modelos de postura são necessários.

Neste capítulo, investigam-se leis de controle visando a implementação. Esta investigação considerará principalmente as leis de controle apresentadas em [10], [11] e [44], pois permitem tratar o modelo completo do robô móvel, incertezas paramétricas e perturbações não modeladas.

Tais leis foram sintonizadas para possibilitar, em trajetórias retilíneas ou do tipo circuito fechado, uma partida suave do robô, desenvolvendo um movimento uniformemente acelerado a partir do repouso até atingir sua velocidade nominal. Também é utilizado o modelo dinâmico completo do robô móvel, incluindo-se atrito de contato com a superfície e perturbações não modeladas, que incluem a dinâmica não modelada do tipo não estruturada.

Para tal modelo, utiliza-se uma estratégia de controle empregando redes neurais artificiais (*RNA's*), cujos pesos e termos de *bias* são atualizados *on-line* por um algoritmo de sintonização que garante pequenos erros de rastreamento e sinais de controle limitados. Simulações com os parâmetros de um robô real; *MagellanTM* [35], explicitam o bom desempenho das estratégias investigadas.

Este capítulo está dividido da seguinte forma: as Seções IV.2 e IV.3 apresentam os modelos cinemáticos e dinâmicos, respectivamente. A Seção IV.4 aborda o projeto do controlador, definindo os problemas básicos de navegação. A Seção IV.5 apresenta o controlador de rastreamento considerando-se o modelo cinemático de postura em coordenadas cartesianas, enquanto que na Seção IV.6 apresenta-se o controlador considerando-se o modelo cinemático de postura em coordenadas polares. Na Seção IV.7, as equações para o controle do modelo dinâmico e, na Seção IV.8, uma aplicação com o *backstepping integrator*. Finalmente, em IV.9, considera-se o controle de robôs móveis utilizando-se o modelo dinâmico em coordenadas cartesianas via redes neurais artificiais.

IV.2 Modelo Cinemático de Postura

Assume-se que o robô seja constituído por um corpo rígido, cujas rodas não sofram deformação e que se movimente exclusivamente no plano horizontal. Considerando-se o robô móvel da Figura IV.1, pode-se definir um vetor de postura pela tripla $(x_c \ y_c \ \theta)$, onde x_c e y_c são as coordenadas do centro de massa P_c , e θ é o ângulo de orientação do robô.

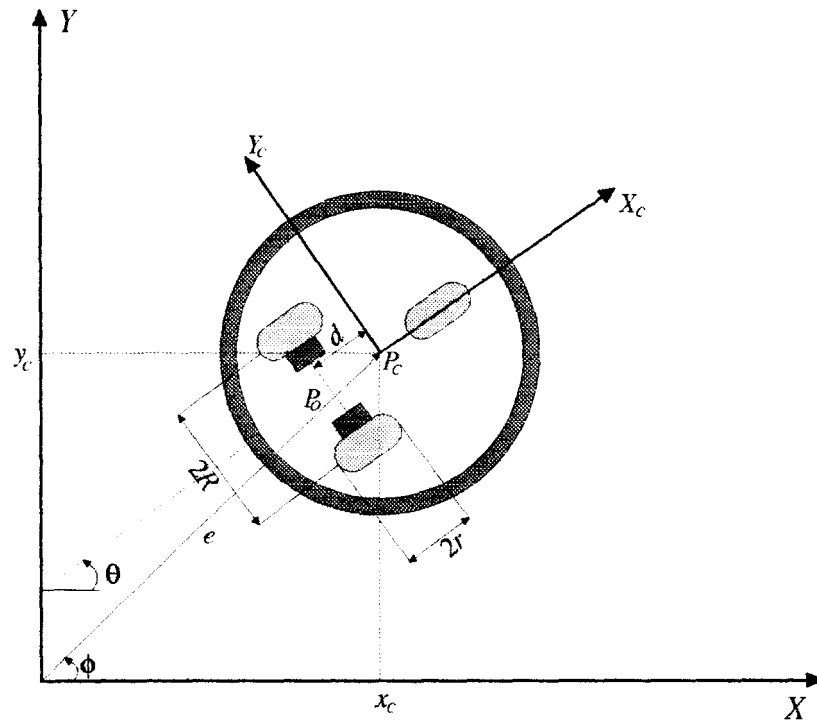


Figura IV. 1. Sistemas de coordenadas para o robô móvel.

A seguinte notação será usada:

P_c centro de massa do robô móvel;

P_o interseção do eixo de simetria com o eixo das rodas;

d distância entre P_o e P_c ;

$2R$ distância entre o centro das rodas;

r raio de cada roda;

m_c massa do robô sem as rodas e os motores;

m_w massa de cada conjunto roda e motor;

I_c momento de inércia do robô sem as rodas e os motores em relação a um eixo vertical através de P_o ;

I_m momento de inércia de cada conjunto roda e motor em relação ao eixo vertical através do diâmetro;

m $m = m_c + 2m_w$ - massa total do robô;

$I = I_c + 2m_w R^2 + 2I_m$ - momento de inércia total do robô.

Para o robô da Figura IV. 1, as velocidades podem ser descritas por

$$\begin{cases} \dot{x}_c = v \cos \theta - \omega d \sin \theta \\ \dot{y}_c = v \sin \theta + \omega d \cos \theta \\ \dot{\theta} = \omega \end{cases} \quad (\text{IV.1})$$

ou em notação matricial,

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -d \sin \theta \\ \sin \theta & d \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{IV.2})$$

onde v é a velocidade linear, e ω a velocidade angular.

A Equação (IV.2) é conhecida como modelo cinemático de postura. Este modelo pode ser também obtido para coordenadas polares [1], [32], [43], onde o vetor de postura é formado pela tripla $(e \ \phi \ \alpha)$, conforme Figura IV. 1. Relacionando-se as coordenadas na figura, obtém-se as Equações (IV.3)-(IV.7),

$$e = \sqrt{x_c^2 + y_c^2} \quad (\text{IV.3})$$

$$\phi = \text{atan}(y_c/x_c) \quad (\text{IV.4})$$

$$\alpha = \theta - \phi \quad (\text{IV.5})$$

$$x_c = e \cos \phi \quad (\text{IV.6})$$

e

$$y_c = e \sin \phi \quad (\text{IV.7})$$

Derivando-se (IV.3) e (IV.4) e substituindo-se o resultado em (IV.6) e (IV.7), obtém-se

$$\dot{e} = \dot{x}_c \cos \phi + \dot{y}_c \sin \phi \quad (\text{IV.8})$$

e

$$\dot{\phi} = \frac{1}{e} (-\dot{x}_c \sin \phi + \dot{y}_c \cos \phi) \quad (\text{IV.9})$$

Portanto, de (IV.2), (IV.8) e (IV.9) obtém-se o modelo cinemático de postura em coordenadas polares,

$$\begin{bmatrix} \dot{e} \\ \dot{\phi} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \frac{\cos \alpha}{\sin \alpha} & -\frac{d \sin \alpha}{d \cos \alpha} \\ \frac{e}{\sin \alpha} & \frac{e}{d \cos \alpha} \\ -\frac{\sin \alpha}{e} & 1 - \frac{d \cos \alpha}{e} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (\text{IV.10})$$

Para um caso mais geral, onde o vetor de referência de postura seja diferente de 0, a determinação da tripla $(e \ \phi \ \alpha)$ requer o uso do sistema de coordenadas mostrado na Figura IV.2.

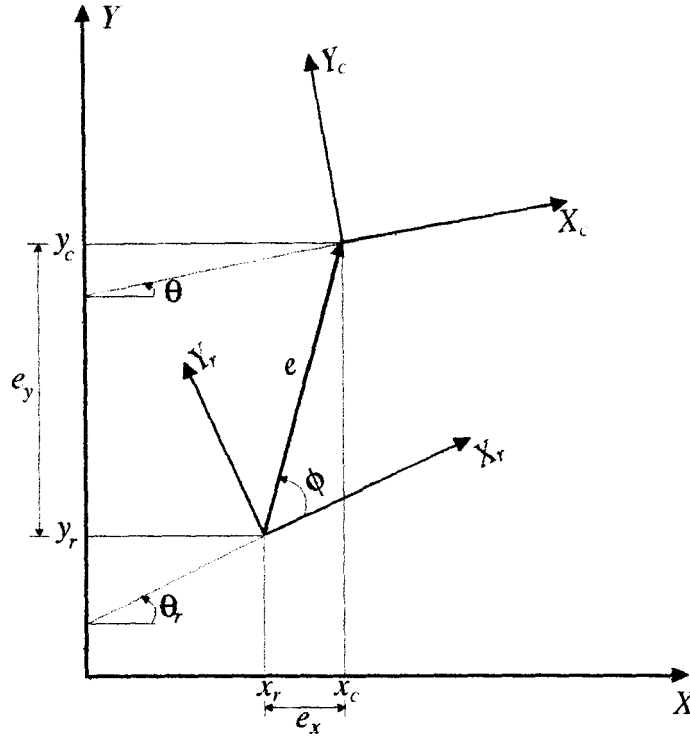


Figura IV.2. Sistemas de coordenadas considerando referência de postura diferente de zero.

Com base na Figura IV.2, tem-se,

$$e = \sqrt{e_x^2 + e_y^2} \quad (\text{IV. 11})$$

$$\phi = \text{atan}\left(\frac{-e_x \sin \theta_r + e_y \cos \theta_r}{e_x \cos \theta_r + e_y \sin \theta_r}\right) \quad (\text{IV.12})$$

e

$$\alpha = e_\theta - \phi \quad (\text{IV.13})$$

onde $(e_x \ e_y \ e_\theta) = (x_c \ y_c \ \theta) - (x_r \ y_r \ \theta_r)$, e os índices "r" representam a postura de referência.

IV.3 Modelo Dinâmico do Robô Móvel Não-Holonômico (*RMNH*)

Para se obter as equações dinâmicas do *RMNH* toma-se necessário considerar as restrições impostas ao movimento, conforme detalhado a seguir.

IV.3.1 Equações de Restrições

Há três restrições para o *RMNH* [48]. A primeira é a de que o robô se move apenas na direção normal ao eixo que une as rodas, isto é, na direção do eixo de simetria. Portanto,

$$\dot{y}_c \cos \theta - \dot{x}_c \sin \theta - d\dot{\theta} = 0 \quad (\text{IV.14})$$

As outras duas restrições asseguram que as rodas do robô não deslizem, resultando

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta + R\dot{\theta} = r\dot{\phi}_r \quad (\text{IV.15})$$

e

$$\dot{x}_c \cos \theta + \dot{y}_c \sin \theta - R\dot{\theta} = r\dot{\phi}_l \quad (\text{IV.16})$$

onde ϕ_r e ϕ_l são o deslocamento angular das rodas direita e esquerda, respectivamente..

Seja o vetor $q = (x_c \ y_c \ \theta \ \phi_r \ \phi_l)$. Portanto, as três restrições podem ser escritas na forma matricial

$$A(q)\dot{q} = 0 \quad (\text{IV.17})$$

onde

$$A(q) = \begin{bmatrix} -\sin \theta & \cos \theta & -d & 0 & 0 \\ -\cos \theta & -\sin \theta & -R & r & 0 \\ -\cos \theta & -\sin \theta & R & 0 & r \end{bmatrix} \quad (\text{IV-18})$$

IV.3.2 Equações Dinâmicas

Obtém-se agora as equações dinâmicas para o *RMNH*. As equações de Lagrange do movimento do robô, com os multiplicadores de Lagrange, são dados, conforme [48], por

$$m\ddot{x}_c - m_c d(\ddot{\theta} \sin \theta + \dot{\theta}^2 \cos \theta) - \lambda_1 \sin \theta - (\lambda_2 + \lambda_3) \cos \theta = 0 \quad (\text{IV.19})$$

$$m\ddot{y}_c + m_c d(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) + \lambda_1 \cos \theta - (\lambda_2 + \lambda_3) \sin \theta = 0 \quad (\text{IV.20})$$

$$-m_c d(\ddot{x}_c \sin \theta - \ddot{y}_c \cos \theta) + I\ddot{\theta} - d\lambda_1 + R(\lambda_3 - \lambda_2) = 0 \quad (\text{IV.21})$$

$$I_\omega \ddot{\phi}_r + \lambda_2 r = \tau_r \quad (\text{IV.22})$$

e

$$I_\omega \ddot{\phi}_l + \lambda_3 r = \tau_l \quad (\text{IV.23})$$

onde τ_r e τ_l são os torques no eixo das rodas gerados pelos motores direito e esquerdo, respectivamente. As Equações (IV.19)-(IV.23) podem ser reescritas na forma vetorial como

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} = E(q)\tau - A^T(q)\lambda \quad (\text{IV.24})$$

sendo $A(q)$ conforme (IV.18) e M, V e E dadas por

$$M(q) = \begin{bmatrix} m & 0 & -m_c d \sin \theta & 0 & 0 \\ 0 & m & m_c d \cos \theta & 0 & 0 \\ -m_c d \sin \theta & m_c d \cos \theta & I & 0 & 0 \\ 0 & 0 & 0 & I_\omega & 0 \\ 0 & 0 & 0 & 0 & I_\omega \end{bmatrix}$$

$$V(q, \dot{q})\dot{q} = \begin{bmatrix} -m_c d \dot{\theta}^2 \cos \theta \\ -m_c d \dot{\theta}^2 \sin \theta \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{e} \quad E(q) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Defina-se ainda uma matriz $S(q)$ tal que $S(q) \in \mathbb{R}^{5 \times 2}$, de forma a se representar a equação de movimento (IV.24) e a equação de restrições (IV.17) na forma de espaço de estados, tal que

$$A(q)S(q) = 0, \text{ onde } 0 \in \mathbb{R}^{3 \times 2} \quad (\text{IV.25})$$

cuja solução é

$$S(q) = \begin{bmatrix} c(R \cos \theta - d \sin \theta) & c(R \cos \theta + d \sin \theta) \\ c(R \sin \theta + d \cos \theta) & c(R \sin \theta - d \cos \theta) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$

onde a constante $c = r/2R$.

De (IV.17) tem-se que \dot{q} está no espaço nulo de $A(q)$, e como as duas colunas de $S(q)$ também estão no espaço nulo de $A(q)$, além de serem linearmente independentes, torna-se possível expressar \dot{q} como uma combinação linear das duas colunas de $S(q)$, conforme

$$\dot{q} = S(q)v \quad (\text{IV.26})$$

onde

$$v = \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} \quad (\text{IV.27})$$

A Equação (IV.26) representa o modelo cinemático de configuração. Vide [48] para detalhes.

Agora, derivando-se (IV.26), substituindo-se o resultado em (IV.24) e pré-multiplicando por S^T , resulta

$$S^T M \dot{S} \dot{v} + (S^T M \dot{S} + S^T V S) v = \tau \quad (\text{IV.28})$$

e fazendo-se

$$\bar{M} = S^T M S \quad \text{e} \quad \bar{V} = S^T M \dot{S} + S^T V S \quad \text{onde} \quad \bar{M} \in \mathbb{R}^{2 \times 2} \quad \text{e} \quad \bar{V} \in \mathbb{R}^{2 \times 2}, \quad \text{tem-se}$$

$$\bar{M} \dot{v} + \bar{V} v = \tau \quad (\text{IV.29})$$

A Equação (IV.29) representa a equação do movimento para o robô móvel (IV.24) e a equação das restrições (IV.17) na forma de espaço de estados, considerando-se os estados como velocidades angulares das rodas do robô. A Equação (IV.29) não considera o atrito de superfície e perturbações não modeladas no sistema, que serão tratados na Seção IV.7.

IV.4 Projeto do Controlador

Para se obter o conhecimento completo do comportamento do sistema há a necessidade de se especificar não apenas as características de guiamento cinemático, como suas velocidades, mas também o conhecimento do seu comportamento dinâmico, como torques aplicados nas rodas, e também, as perturbações atuantes sobre ele. A dinâmica modelada para o RMNH é descrita pelas Equações (IV.24) ou (IV.29). Neste capítulo serão consideradas estratégias de controle que consideram ambos os aspectos, o cinemático e o dinâmico, e serão apresentados e discutidos os resultados.

Há três situações básicas para o problema de controle: a) rastreamento de uma trajetória de referência, b) seguimento de um caminho e c) estabilização em torno de uma postura. A tarefa é selecionar uma velocidade de controle, representada por v_c , para o sistema de guiamento. Tais problemas são formulados conforme a seguir.

IV.4.1 Rastreamento

O problema do rastreamento da trajetória para veículos não-holonômicos é considerado da seguinte maneira: considerando-se a Figura IV.1, tem-se a seguinte referência:

$$\begin{cases} \dot{x}_r = v_r \cos \theta_r \\ \dot{y}_r = v_r \sin \theta_r \\ \dot{\theta}_r = \omega_r \\ p_r = (x_r, y_r, \theta_r) \\ v_r = (v_r, \omega_r) \end{cases} \quad (IV.30)$$

onde p_r representa o vetor de posturas de referência e v , o vetor das velocidades linear e angular de referência.

O objetivo é encontrar um controlador suave de velocidade,

$$v_c(t) = f_c(e_p, v_r, K) \text{ tal que } \lim_{t \rightarrow \infty} (p_r - p) = 0$$

onde e_p, v_r e K são o erro de rastreamento, conforme em (IV.31), o vetor de velocidade referencial e o vetor de ganho do controlador, respectivamente. Portanto, basta se calcular o torque τ em (IV.24), tal que force $v \rightarrow v_c$ quando $t \rightarrow \infty$.

IV.4.2 Seguimento de Caminho

Dado um caminho P no plano e a velocidade linear do robô móvel $v(t)$, o objetivo para esta situação é encontrar um controlador de velocidade,

$$v_c(t) = f_c(e_\theta, v, d, K) \text{ tal que } \lim_{t \rightarrow \infty} e_\theta = 0 \text{ e } \lim_{t \rightarrow \infty} d(t) = 0$$

onde e_θ é o erro de orientação e $d(t)$ a distância no plano horizontal entre um ponto de referência no robô e o caminho P . Portanto, deve-se determinar o torque τ em (IV.24) tal que $v \rightarrow v_c$ quando $t \rightarrow \infty$.

IV.4.3 Estabilização em uma Postura

Dada uma postura arbitrária p_r , tomada como referência, o objetivo está em se determinar o controlador

$$v_c(t) = f_c(e_p, v_r, K, t) \text{ tal que } \lim_{t \rightarrow \infty} (p_r - p) = 0$$

Para tanto, calcula-se o torque τ em (IV.24) tal que $v \rightarrow v_c$ quando $t \rightarrow \infty$. Para a estabilização em torno de uma postura, considera-se v_r e ω_r nulos.

Convém lembrar que o problema de estabilização em uma postura não pode ser resolvido mediante sinal de controle suave. Este problema não constitui caso de interesse para esta tese. Para maiores detalhes vide [47].

IV.5 Controle de *RMNH* considerando-se o Modelo Cinemático de Postura em Coordenadas Cartesianas

O modelo cinemático do robô móvel em coordenadas cartesianas é dado por (IV.2). Para o caso do robô móvel rastreando uma trajetória de referência, como na Seção IV.4.1, considere-se o erro de postura e_p , ou *tracking error*, como a diferença entre a postura de referência e a postura atual do robô, com a devida transformação de coordenadas. Assim, tem-se

$$e_p = T_e(p_r - p),$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (\text{IV.31})$$

onde T_e é a matriz de rotação entre o sistema de coordenada inercial $(X \ Y \ Z)$ e o sistema de coordenadas localizado em um ponto do robô $(X_r \ Y_r \ Z_r)$.

IV.5.1 0 Controlador de Rastreamento

Para o sistema de controle de rastreamento, a entrada do sistema é a postura de referência p_r e as velocidades de referência, $v_r = (v_r \ \omega_r)^T$, que são variantes no tempo. A saída do sistema é a postura atual p . O objetivo deste controlador é obter a convergência do erro de postura e_p para zero. O diagrama de blocos da Figura IV.3, conforme [27], mostra a estrutura deste sistema de controle.

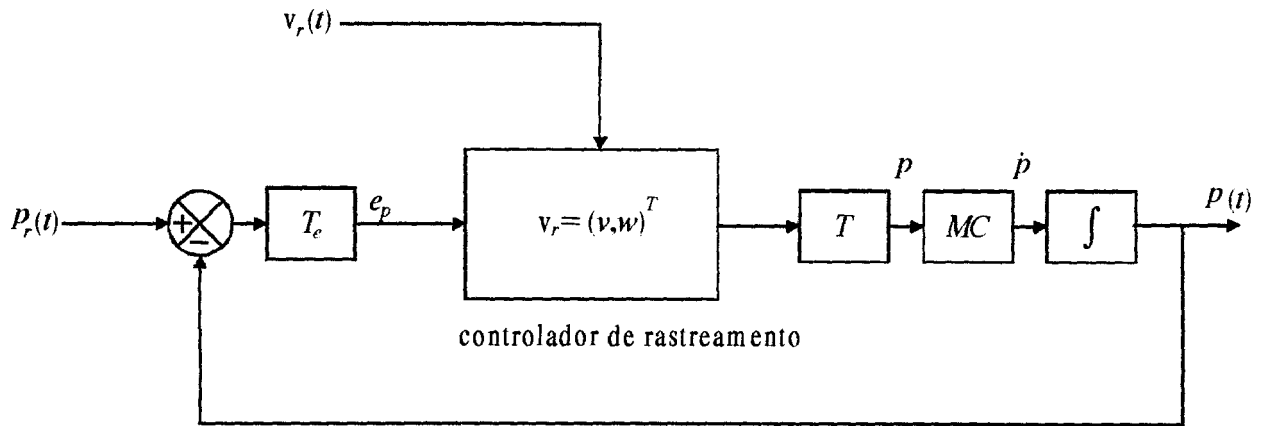


Figura IV.3. Estrutura do sistema de controle para rastreamento.

No diagrama da Figura IV.3, o bloco designado por T representa a função de transferência que relaciona a velocidade de cada roda com a tensão aplicada no motor correspondente. Para a situação de controle onde se considera apenas o modelo cinemático do robô, assume-se "rastreamento perfeito de velocidade" fazendo $T=1$, o que não ocorre na prática. Para situações reais como em implementação das estratégias de controle em robôs, tal função de transferência é determinada experimentalmente, como em [32].

O bloco identificado por MC representa a matriz do modelo cinemático do robô em coordenadas cartesianas, dada na Equação (IV.2).

A regra de controle proposta originariamente por [27], para o rastreamento de veículos não-holonômicos tem a forma

$$v_c = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v(e_p, p_r) \\ \omega(e_p, p_r) \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \quad (IV.32)$$

com $v_c(t) = f_c(e_p, v_r, K, t)$ e $K = (k_1 \ k_2 \ k_3)^T$, onde os parâmetros de projeto k_i 's (ganhos do controlador) são todos positivos. A regra de controle em (IV.32) apresenta estabilidade assintótica em torno de $e_p = 0$. Em [27] apresenta-se a prova através do segundo método de Lyapunov.

No projeto do controlador, os ganhos k_i s são ajustados experimentalmente de forma a se manter um compromisso entre a velocidade de resposta do controlador e o fato de ser implementável. Como os ganhos afetam diretamente as velocidades de controle, conforme (IV.32), é necessário ajustá-los de forma a não se ultrapassar os limites de *hardware* do robô. Outro fator importante no projeto de controladores de rastreamento implementáveis é o cuidado com a geração da trajetória de referência. Deve-se ter o cuidado de se gerar trajetórias contínuas e suaves para que o rastreamento seja possível.

Seguem as simulações para o controlador cinemático de postura em coordenadas cartesianas para o robô móvel *MagellanTM* [35], cujos parâmetros constam na Tabela IV.1.

As simulações foram implementadas em MATLABTM.

A Figura IV.4 mostra a trajetória definida como desejada para o robô e seu rastreamento segundo a lei de controle exibida em (IV.32). A postura de partida do robô foi atribuída como $p = (2 \ 1 \ 0)$ e os ganhos do controlador ajustados experimentalmente como $K = (3 \ 2 \ 1)$. Os retângulos desenhados sobre as trajetórias servem para se determinar a orientação instantânea do robô. A trajetória de referência tem $p_r = (2 \ 2 \ \pi/4)$ como postura de partida. Para esta simulação, gerou-se uma trajetória de referência retilínea, com

$v_r = 1\text{ m/s}$ e $\omega_r = 0\text{ rad/s}$. Para se evitar descontinuidades na lei de controle, considerou-se um movimento retilíneo uniformemente acelerado para os dois primeiros segundos para a trajetória desejada para o robô móvel. Este procedimento permite que o robô tenha um início de movimento contínuo e suave, resultando em uma lei de controle que pode ser implementada, pois respeita as velocidades máximas permitidas para o robô.

A Figura IV.5 apresenta os sinais de controle para v e ω requeridos para que o robô atinja a trajetória predeterminada. A monitoração dos sinais de controle mostram que este rastreamento é possível de ser aplicado na prática, dado que não supera os limites de *hardware* (do robô) para velocidades linear e angular máximas. A Figura IV.6 apresenta os erros de postura, isto é $e_1 = x_r - x$; $e_2 = y_r - y$ e $e_3 = \theta_r - \theta$. Pelo comportamento dos erros de postura, conclui-se que o desempenho do sistema de controle é adequado.

Tabela IV.1. Parâmetros físicos do robô *MagellanTM*.

	<i>Parâmetros do MagellanTM</i>
$m_c(Kg)$	9.1
$I_c(Kg \cdot m^2)$	0.2234
$R(m)$	0.18
$r(m)$	0.057
$d(m)$	0
$v_{\max}(m/s)$	2.5
$\omega_{\max}(^\circ/s)$	270

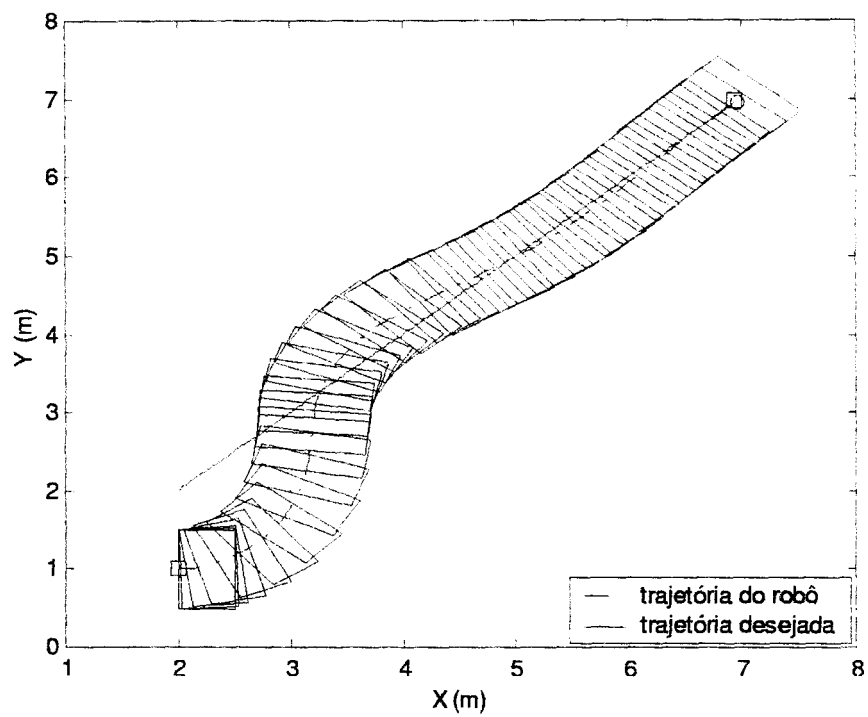


Figura IV.4. Trajetória desejada e trajetória obtida para o robô segundo o controlador de rastreamento apresentado em (IV.32).

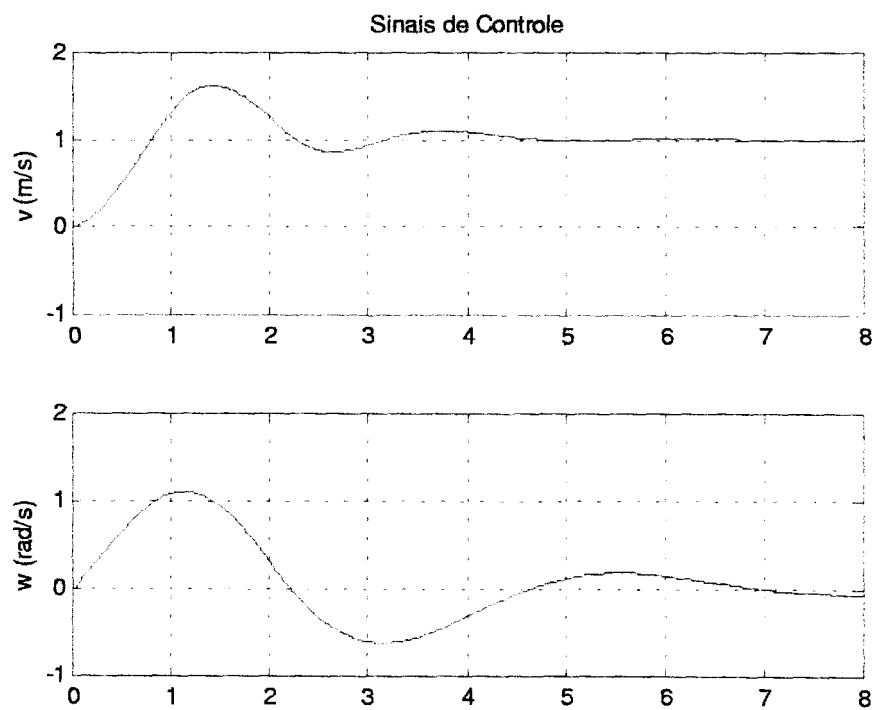


Figura IV.5. Sinais de controle para as velocidades linear e angular, obtidas via (IV.32).

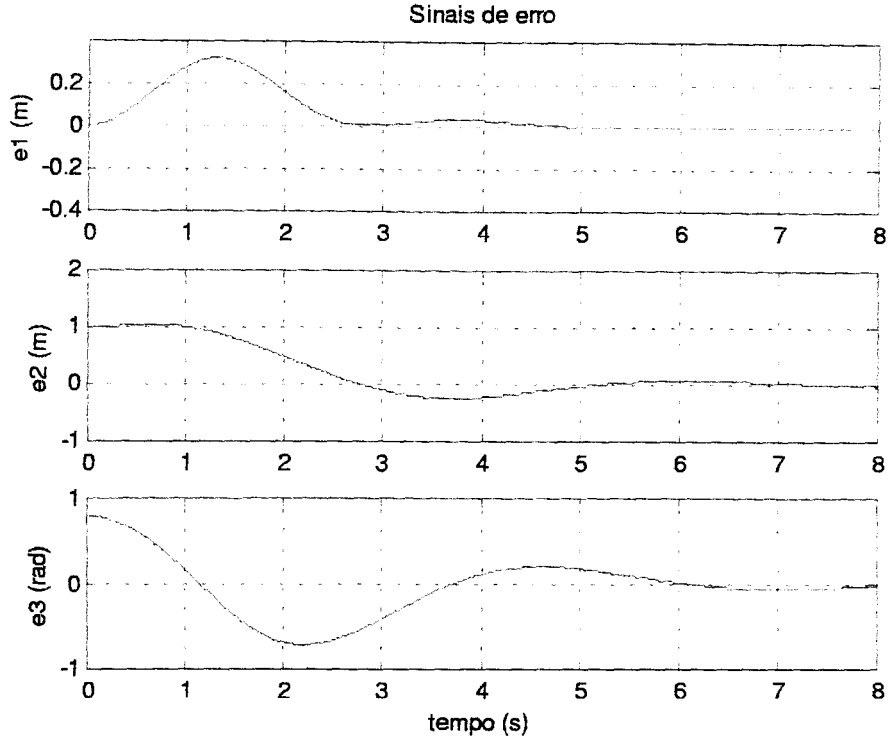


Figura IV.6. Erros de postura, obtidos via (IV.31).

O tempo de amostragem (passo de integração da simulação digital) considerado nas simulações foi de $T = 0.01$ s, conforme [48].

As Figuras IV.7, IV.8 e IV.9 apresentam o resultado obtido para uma trajetória de referência tipo circuito fechado, isto é, partindo de um ponto e retornando a ele. A postura de partida do robô foi atribuída como $p = (1 \ 0 \ 0)$ e os ganhos do controlador cinemático ajustados como $K = (5 \ 4 \ 3)$. Os retângulos auxiliares na Figura IV.7 servem para indicar a orientação do robô. A trajetória de referência tem $p_r = (1 \ 1 \ 0)$ como postura inicial. Considerou-se uma trajetória onde $v_r = 1 \text{ m/s}$ durante todo o trajeto e velocidade angular diferente de zero nas curvas. Para se evitar mudanças bruscas na velocidade angular, o que resultaria em descontinuidades na lei de controle, adotou-se que ao se iniciar a curva o robô possuiria velocidade angular nula, que iria aumentando gradativamente até $\omega_r = 1 \text{ rad/s}$, e a

partir deste valor, diminuindo até sair da curva com velocidade angular nula novamente. Para se evitar que seja requerido um sinal de controle muito elevado nos primeiros instantes do movimento, a trajetória definida considera o robô partindo do repouso e atingindo a velocidade nominal em 3 s. Esta implementação respeita os limites de hardware do robô, conforme pode ser visto nas Figuras IV.8 dos sinais de controle e IV.9 dos sinais de erro.

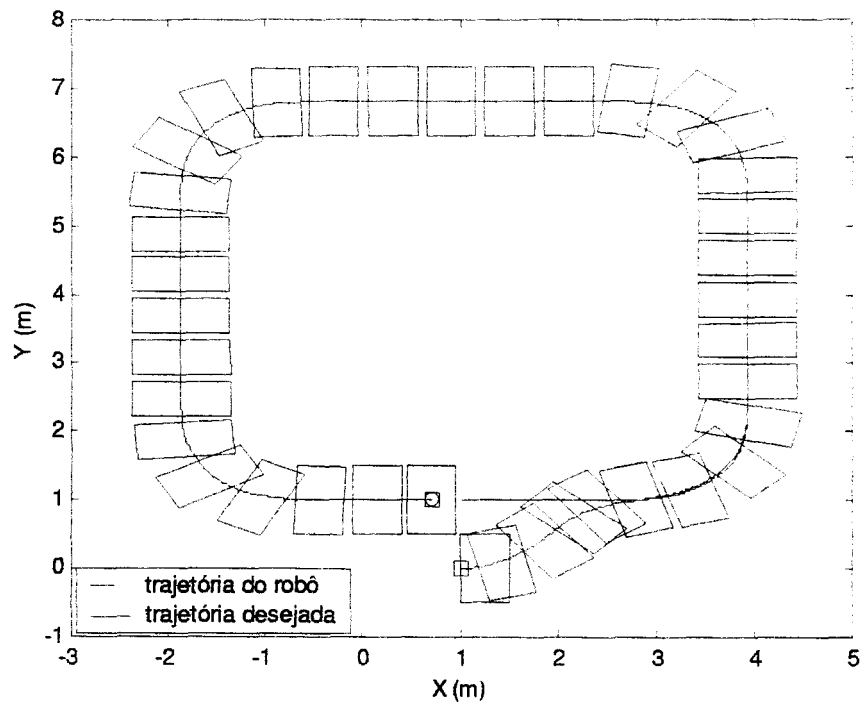


Figura IV.7. Trajetória desejada e trajetória obtida para o robô, com controlador de rastreamento apresentado em (IV.32), para trajeto tipo circuito fechado.

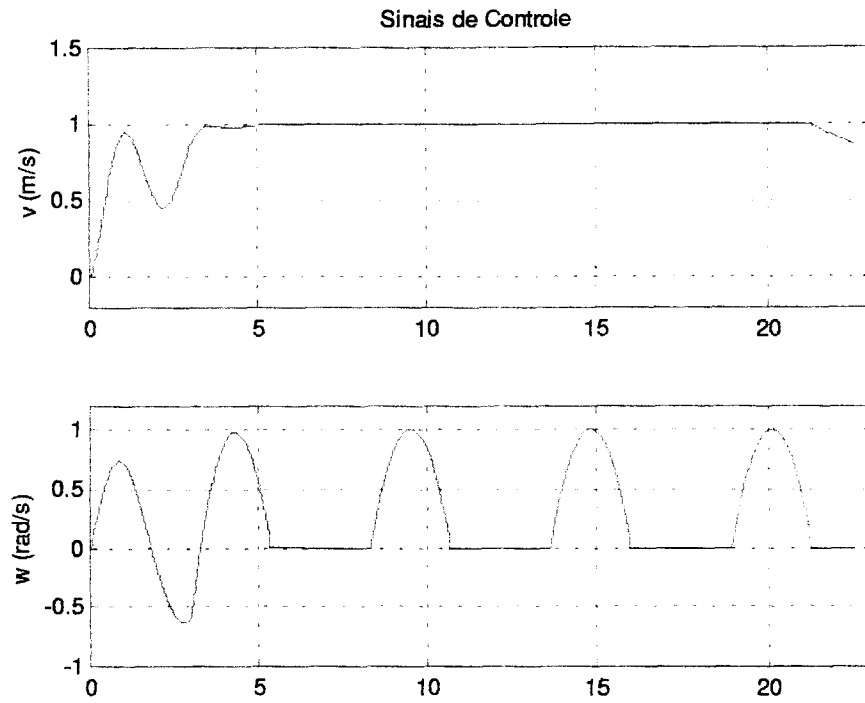


Figura IV.8. Sinais de controle para as velocidades linear e angular, obtidas via (IV.32), para trajeto tipo circuito fechado.

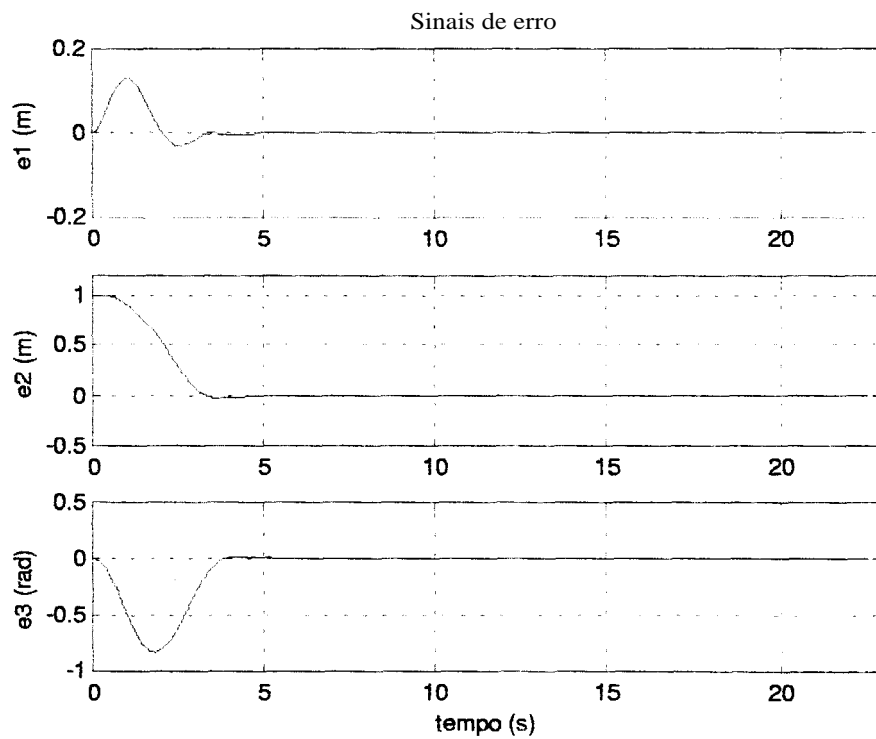


Figura IV.9. Erros de postura obtidos via (IV.32), para trajeto tipo circuito fechado.

IV.6 Controle de *RMNH* considerando-se o Modelo Cinemático de Postura em Coordenadas Polares

O modelo cinemático do *RMNH* em coordenadas polares é dado pela Equação (IV.10). A lei de controle proposta para o modelo cinemático em coordenadas polares se baseia em análise empregando o segundo método de Lyapunov e lema de Barbalat. Em [1], [32], [43] propõe-se a seguinte candidata a função de Lyapunov para o caso onde $d=0$,

$$V(e, \phi, \alpha) = \frac{1}{2}(e^2 + \alpha^2 + h\phi^2) \quad (IV.33)$$

onde h é uma constante positiva. Tal função apresenta a derivada

$$\dot{V}(e, \phi, \alpha) = e \cos \alpha + \alpha \left[\omega - \frac{\sin \alpha}{e \alpha} (\alpha - h\phi) v \right] \quad (IV.34)$$

Considerando-se

$$v = -\gamma_1 e \cos \alpha \quad (IV.35)$$

e

$$\omega = -\gamma_2 \alpha - \gamma_1 \cos \alpha \frac{\sin \alpha}{\alpha} (\alpha - h\phi) \quad (IV.36)$$

resulta

$$\dot{V}(e, \phi, \alpha) = -\gamma_1 e^2 \cos^2 \alpha - \gamma_2 \alpha^2 \leq 0 \quad (IV.37)$$

Dado que a candidata a função de Lyapunov é contínua e $\dot{V} \leq 0$, garante-se a estabilidade do sistema em malha fechada. Além disto, considerando-se que V é limitada inferiormente, é não-crescente no tempo (pois $\dot{V} \leq 0$), V é uniformemente contínua (pois V é limitada), tem-se, pelo lema de Barbalat, que V tende a zero, garantindo-se que e, ϕ e α tendem para zero [1], [32] e [45].

As simulações para o modelo cinemático de postura em coordenadas polares são bastante semelhantes às aquelas apresentadas para o modelo cinemático de postura em coordenadas cartesianas, razão pela qual serão omitidas aqui.

Uma das vantagens em se usar o modelo cinemático em coordenadas polares está no fato de não haver necessidade de se conhecer as velocidades de referência para o controle, como pode ser visto nas Equações (IV.35) e (IV.36). Adicionalmente, pode-se efetuar o controle mesmo quando as velocidades de referência são nulas, o que é impossível em coordenadas cartesianas, conforme a Equação (IV.32).

IV.7 Controle de *RMNH* considerando-se o Modelo Dinâmico

Como já foi citado anteriormente, os modelos dinâmicos descrevem o robô em função das forças generalizadas aplicadas pelos atuadores. Portanto, considera informações adicionais ao modelo cinemático para determinar um sistema de controle mais eficiente.

Seja a Equação (IV.24). De modo a se considerar uma descrição mais realista para o robô móvel, incorpora-se ao modelo o atrito de contato do robô com a superfície além de perturbações no sistema que não são modeladas. Desta forma, a equação dinâmica para o *RMNH* dada em (IV.24) é reescrita como

$$M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F(\dot{q}) + \tau_d = E(q)\tau - A^T(q)\lambda \quad (IV.38)$$

onde os termos adicionais em (IV.38), em relação a (IV.24), são $F(q)$, atrito de contato com a superfície, e τ_d , perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada. A Equação (IV.29), na forma de espaço de estados, por sua vez, pode ser escrita como

$$\bar{M}\dot{v} + \bar{V}v + \bar{\mu} = \tau \quad (IV.39)$$

onde $\bar{\mu}$ representa a soma de $F(\dot{q})$ e τ_d .

Defina-se, agora, o erro da velocidade de rastreamento como

$$e_c = \Gamma \dot{v}_c - v \quad (\text{IV.40})$$

onde $\dot{v}_c \in \mathbb{R}^{2 \times 1}$ é a lei de controle para velocidade aplicada ao robô, $v \in \mathbb{R}^{2 \times 1}$ é o vetor em (IV.27), e $\Gamma \in \mathbb{R}^{2 \times 2}$ é a matriz que relaciona as velocidades de deslocamento angular das rodas com as velocidades linear e angular do robô. Esta relação é dada por

$$v = \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} = \Gamma \begin{bmatrix} v \\ \omega \end{bmatrix}, \text{ onde } \Gamma = \begin{bmatrix} 1 & R \\ r & r \\ 1 & -R \\ r & r \end{bmatrix} \quad (\text{IV.41})$$

Portanto, pode-se reescrever (IV.39) em função do erro da velocidade de rastreamento, conforme [11],

$$\bar{M} \dot{e}_c = -\bar{V} e_c + f(x) - \bar{\mu} - \tau \quad (\text{IV.42})$$

onde

$$f(x) = \bar{M} \Gamma \dot{v}_c + \bar{V} v \quad (\text{IV.43})$$

Portanto, para o cálculo de $f(x)$ é necessário o conhecimento de \dot{v}_c , v e, pela Equação (IV.41), de \dot{v}_c . Logo, o vetor requerido para o cálculo de $f(x)$ pode ser definido como

$$x \equiv \begin{pmatrix} v^T & v_c^T & \dot{v}_c^T \end{pmatrix} \quad (\text{IV.44})$$

IV.8 Controle de *RMNH* considerando-se o *Backstepping Integrator*

Considere-se o sistema dado por

$$\begin{cases} \dot{\eta} = f(\eta) + g(\eta)\xi \\ \dot{\xi} = u \end{cases} \quad (\text{IV.45})$$

onde $(\eta^T, \xi)^T \in \mathbb{R}^{n+1}$ é o estado e $u \in \mathbb{R}$ é a entrada de controle. As funções $f: D \rightarrow \mathbb{R}^n$ e $g: D \rightarrow \mathbb{R}^n$ são suaves no domínio $D \subset \mathbb{R}^n$, que contém $\eta = 0$ e $f(0) = 0$. O objetivo para este sistema é se projetar um controle de realimentação de estado para estabilizar a origem $(\eta = 0, \xi = 0)$, assumindo-se f e g conhecidas.

Considerando-se ainda o modelo cinemático de configuração em (IV.26) e o controle *tipo backstepping integrator*, tem-se

$$\begin{cases} \dot{q} = S(q)v \\ \dot{v} = u \end{cases} \quad (\text{IV.46})$$

e deseja-se converter o controle de velocidade $v(t)$ em controle de torque $r(t)$ que considera parâmetros do robô real além de atrito de contato com a superfície e perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada.

O sistema de controle de guiamento suave, dado por v , pode ser encontrado pelas técnicas de controle cinemático em coordenadas cartesianas ou polares, conforme exposto anteriormente.

O objetivo agora é se obter $u(t)$ e $z(t)$ convenientes para uma lei cinemática de controle $v, (t)$ específica que controla o sistema de guiamento dado em (IV.46).

Seja o controlador de rastreamento descrito pelas Equações (IV.31) e (IV.32). Suas respectivas derivadas são dadas por [10], [25], [27]

$$\dot{e}_p = \begin{bmatrix} v_2 e_2 - v_1 + v_r \cos e_3 \\ -v_2 e_1 + v_r \sin e_3 \\ \omega_r - v_2 \end{bmatrix}, \text{ onde } v_1 = v \text{ e } v_2 = \omega \quad (\text{IV.47})$$

e

$$\dot{v}_c = \begin{bmatrix} \dot{v}_r \cos e_3 \\ \dot{\omega}_r + k_2 \dot{v}_r e_2 + k_3 \dot{v}_r \sin e_3 \end{bmatrix} + \begin{bmatrix} k_1 & 0 & -v_r \sin e_3 \\ 0 & k_2 v_r & k_3 v_r \cos e_3 \end{bmatrix} \dot{e}_p \quad (\text{IV.48})$$

A entrada de controle de realimentação não-linear proposta em [101, modificada para comportar a matriz que relaciona velocidades de deslocamento angular das rodas com velocidade de deslocamento do robô, é dada por

$$u = \dot{v}_c + K_4(v_c - v) \quad (IV.49)$$

onde K_4 é uma matriz diagonal positiva definida, dada por

$$K_4 = k_4 I \quad (IV.50)$$

A prova de estabilidade assintótica uniforme da origem e $p = 0$ pode ser encontrada em [10]. A estrutura do sistema de controle é resumida na Figura TV. 10.

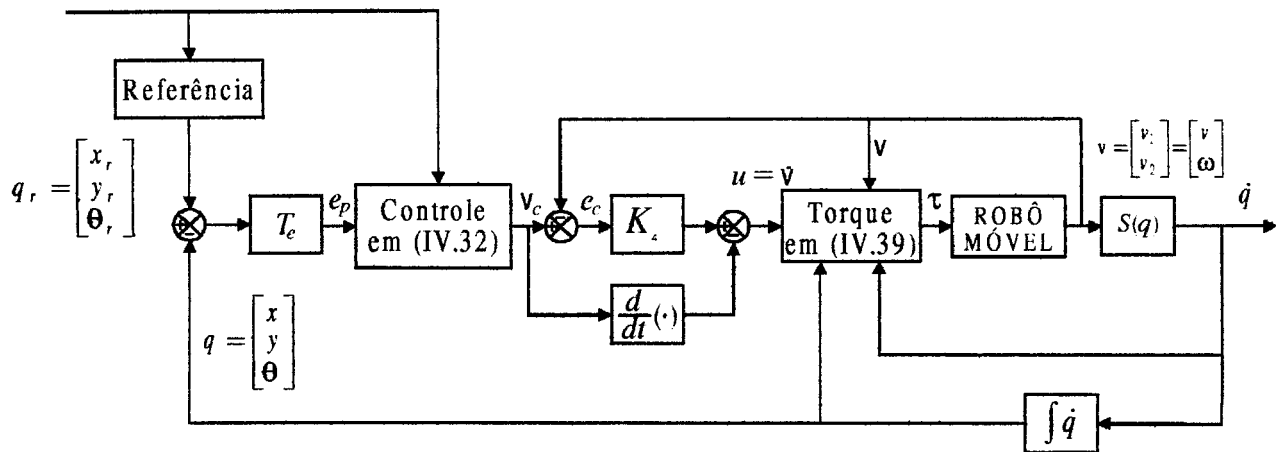


Figura TV. 10. Estrutura do sistema de controle tipo *backstepping integrator*.

A Figura IV. 11 mostra a trajetória desejada para o robô e a obtida utilizando-se *backstepping integrator* como em (IV.46)-(IV.49). A trajetória desejada foi gerada considerando-se uma velocidade angular constante de $\omega_r = 0.1 \text{ rad/s}$ e a velocidade linear uniformemente variada nos primeiros 2 s, para se evitar picos na lei de controle, e no restante do movimento assumiu-se $v_r = 1 \text{ m/s}$. A trajetória de referência tem $p_r = (0 \ 1 \ 0)$ como postura inicial, e a postura de partida para o robô é $p = (0 \ 0 \ 0)$, conforme pode ser visto na Figura IV. 11. Os ganhos do controlador cinemático são ajustados experimentalmente como

$K = \begin{pmatrix} 4 & 2 & 1 \end{pmatrix}$, e $k_4 = 1$ em (IV.50). Para a simulação das perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada considera-se um sinal de ruído com distribuição normal $N \sim (0,10)$ em unidades de N.m, e para o atrito de superfície, atribuiu-se também uma distribuição aleatória normal $N \sim (0,1)$ em unidades de N.m. A Figura IV.12 apresenta os sinais de controle v e ω necessários para que o robô atinja a trajetória desejada. Note-se mais uma vez que são respeitados os limites de *hardware* do robô, e a Figura TV. 13 apresenta os erros de postura. Mesmo considerando-se as perturbações, que incluem a dinâmica não modelada do tipo não estruturada do robô, o atrito com a superfície e as incertezas associadas a essas grandezas, pode-se observar o desempenho satisfatório do sistema controlado. A Figura IV.14 apresenta os torques aplicados às rodas do robô, e a Figura IV.15, as posturas referenciais e obtidas para o robô.

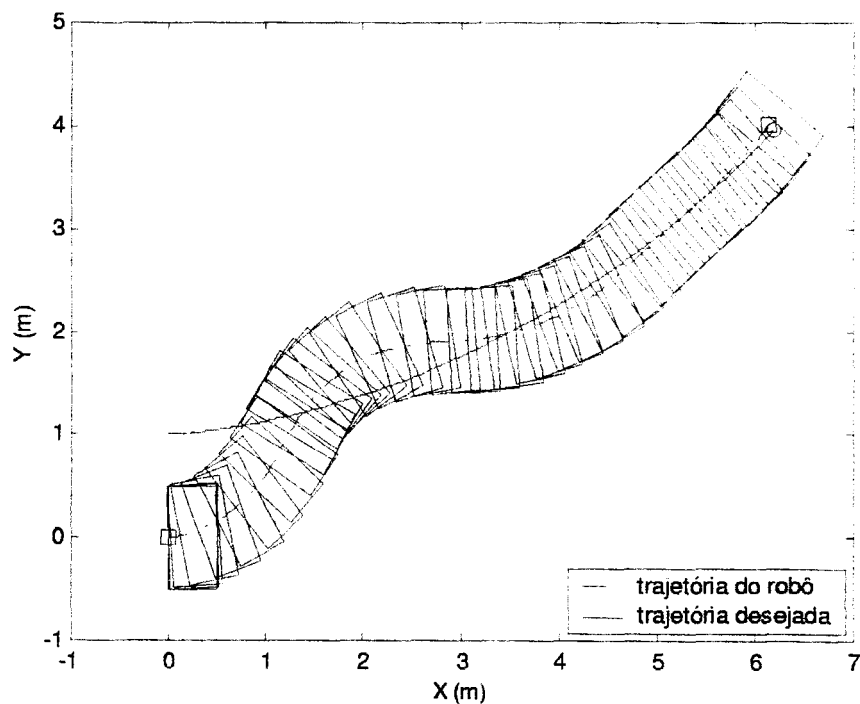


Figura IV. 11. Trajetórias desejada e obtida para o robô, empregando-se o controlador tipo *backstepping integrator*.

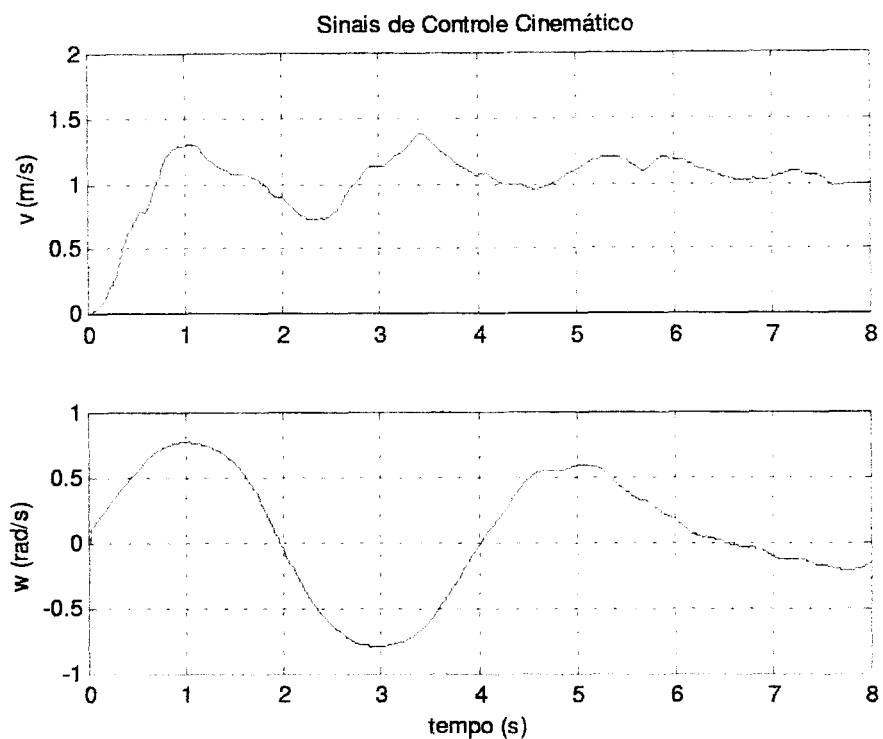


Figura IV.12. Sinais de controle para as velocidades linear e angular, obtidas para o controlador *tipo backstepping integrator*.

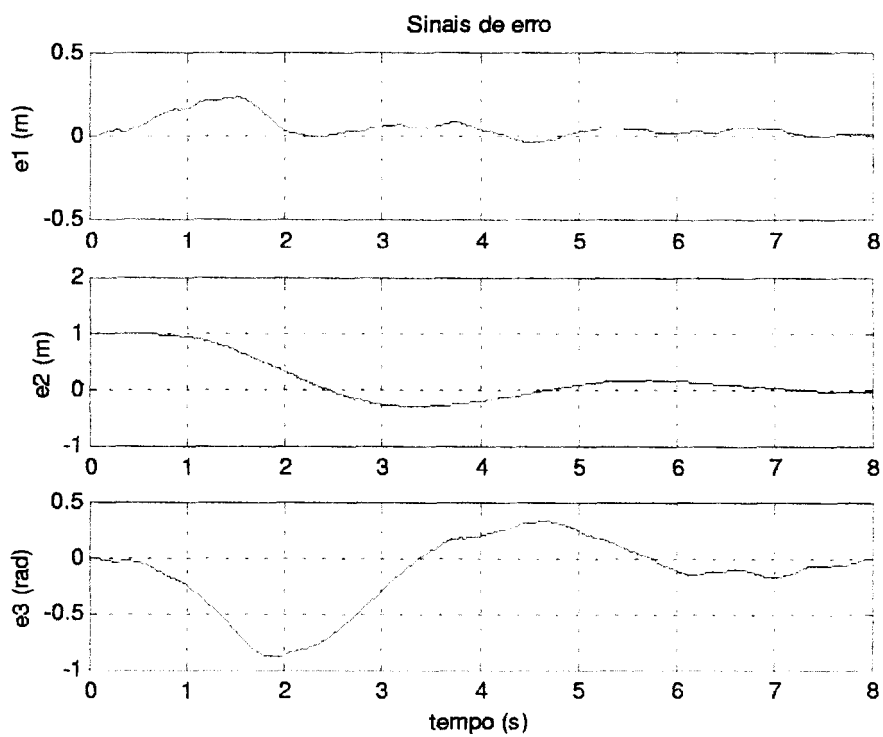


Figura IV.13. Erros de postura, obtidos com o controlador *tipo backstepping integrator*.

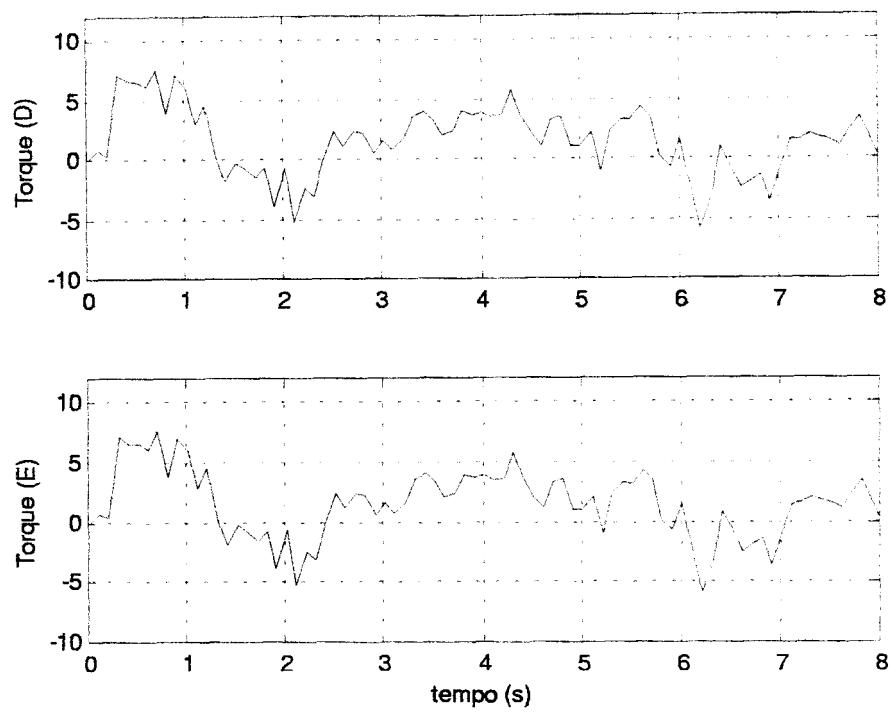


Figura IV. 14. Torques aplicados nas rodas direita e esquerda, obtidos via (IV.39).

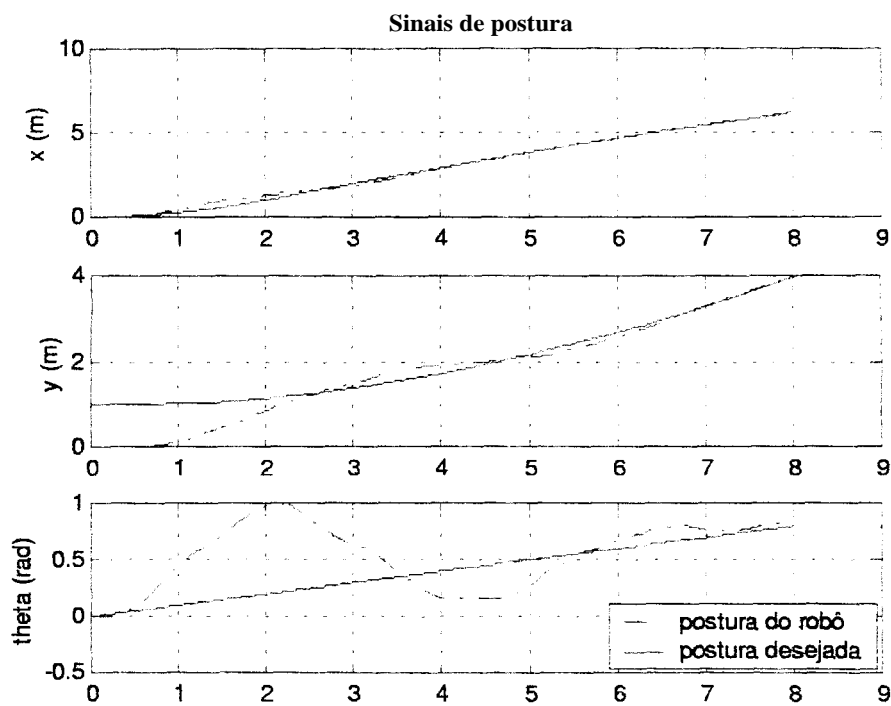


Figura IV.15. Evolução das posturas desejadas e obtidas durante o movimento do robô.

Convém ressaltar que diversas outras simulações foram efetuadas, e resultados apresentados neste capítulo devem ser considerados como resultados típicos, sob as mesmas condições de teste em [10], [11] e [44].

IV.9 Controle de *RMNH* considerando-se o Modelo Dinâmico em

Coordenadas Cartesianas utilizando-se Redes Neurais Artificiais

Controladores baseados em *RNA* 's constituem a ênfase nesta seção, pois conforme já explicitado em [10], [11] e [43] permitem a obtenção de bom desempenho mediante o tratamento do modelo completo do robô, incertezas paramétricas e perturbações não modeladas.

IV.9.1 Capacidade de Aproximação Universal das *RNA* 's

A área de controle de sistemas não-lineares, tais como o *RMNH*, é uma das maiores responsáveis pela aplicação de *RNA* 's. Sua principal motivação está baseada na habilidade das *RNA* 's de aproximar qualquer mapeamento contínuo.

A teoria da aproximação tem sido extensivamente estudada e segundo o teorema de Stone-Weierstrass, de 1976, qualquer função contínua de várias variáveis pode ser aproximada através da superposição de funções de uma variável. O teorema de Cybenko, de 1989, desempenhou papel definitivo na popularização do uso das *RNA* 's como aproximadores universais.

IV.9.2 Topologia das *RNA* 's

De acordo com a topologia, uma *RNA* pode ser classificada em *feedforward* (sem realimentação local) ou *feedback* (com realimentação local ou recorrente) [17],[39], [49]. Em uma *RNA* tipo *feedforward*, uma unidade envia sua saída apenas para unidades das quais ela não receba nenhuma entrada direta ou indiretamente, isto é, em uma rede tipo *feedforward* não existem laços de realimentação.

Na aplicação deste trabalho para controle de *RMNH*, utiliza-se redes tipo *feedforward*.

A Figura IV. 16 representa uma *RNA feedforward* com três camadas.

A vantagem em se utilizar camadas escondidas está no fato de que o poder de representação da rede é bastante aumentado. A capacidade de representação da rede está relacionado à complexidade da função que se deseja aproximar via *RNA*. Geralmente não se utiliza mais de duas camadas escondidas na rede, pois em geral a capacidade de representação de duas camadas escondidas é suficiente para a tarefa, e o tempo de treinamento da rede aumenta proporcionalmente ao número de conexões.

Para os objetivos deste trabalho, considera-se uma rede neural com a estrutura mostrada na Figura IV.16.

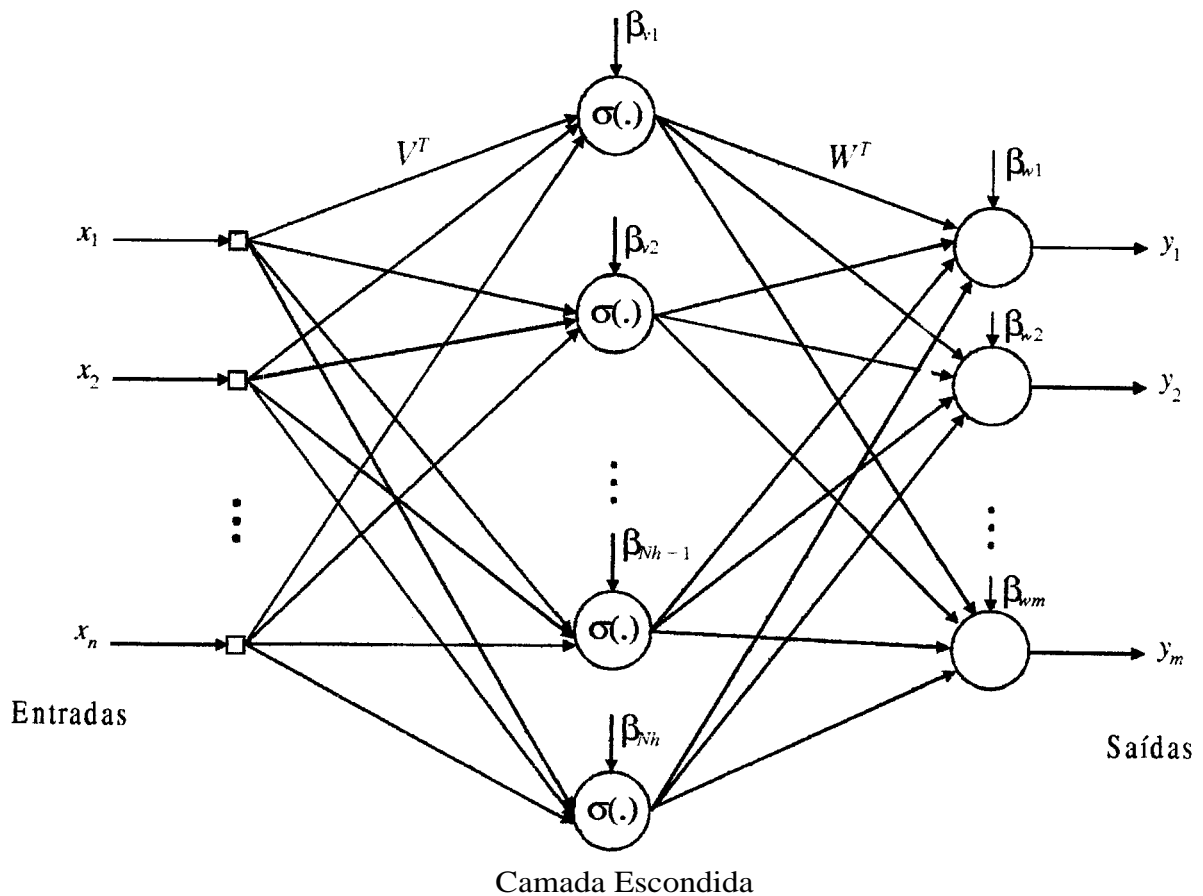


Figura IV.16. *RNA feedforward* com uma camada escondida.

Segundo a Figura IV.16, a rede possui um vetor de entrada $x \in \mathbb{R}^n$, e um vetor de saída $y \in \mathbb{R}^m$ dado por

$$y_i = \sum_{j=1}^{N_h} \left[w_{ij} \sigma \left(\sum_{k=1}^n v_{jk} x_k + \beta_{vj} \right) + \beta_{wi} \right] \text{ com } i = 1, \dots, m \quad (\text{IV.51})$$

onde $\sigma(\cdot)$ é a função de ativação. Neste caso, a função de ativação utilizada é a sigmoideal, isto é,

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (\text{IV.52})$$

N_h é o número de neurônios na camada escondida, v_{jk} são os pesos de conexão entre as entradas e a camada escondida, e w_{ij} os pesos de conexão entre os neurônios da camada escondida e as saídas da rede. Os termos de *bias* são dados por β_{vj} e β_{wi} . O termo de *bias* é o valor de limiar que pode ser interpretado como sendo um outro peso vindo de uma unidade cuja saída é sempre "1".

Agrupando-se os pesos v_{jk} e w_{ij} da RNA em matrizes de pesos V^T e W^T , (IV.51) pode ser reescrita na forma vetorial como

$$y = W^T \sigma(V^T x) \quad (\text{IV.53})$$

com o vetor da função de ativação definido por $\sigma(z) = [\sigma(z_1) \sigma(z_2) \dots \sigma(z_n)]^T$, onde $z \in \mathbb{R}^n$.

Os *bias* são incluídos como as primeiras colunas nas matrizes de peso. Para acomodá-los, os vetores x e $\sigma(\cdot)$ precisam ser aumentados, acrescentando-se "1" como seus primeiros elementos. Portanto, qualquer ajuste dos pesos V e W inclui o ajuste dos *bias*.

Como exposto na Seção IV.9. 1, uma função suficientemente suave pode ser aproximada por uma *RNA* convenientemente dimensionada. Portanto, uma função $f : \Re^n \rightarrow \Re^m$ pode ser escrita como

$$f(x) = W^T \sigma(V^T x) + \varepsilon(x) \quad (\text{IV.54})$$

onde $\varepsilon(x)$ é o erro de reconstrução da *RNA*. Para qualquer valor de ε_N positivo, pode-se encontrar pesos e bias de uma *RNA* tal que $\varepsilon < \varepsilon_N$ em um conjunto compacto. Portanto, uma estimativa de $f(x)$ pode ser dada por

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \quad (\text{IV.55})$$

onde \hat{V} e \hat{W} são as estimativas dos pesos da *RNA* que são calculados *on-line* por algoritmos de sintonização.

IV.9.3 Estrutura do Controlador para o *RMNH* via *RNA* 's

A Figura IV. 17 mostra a estrutura do controlador para o ajuste utilizando controle via redes neurais. Em aplicações práticas, a função não-linear $f(x)$ do robô é, pelo menos, parcialmente conhecida. Portanto, pode-se aplicar um controlador para rastreamento de referência, baseado no cálculo do torque a ser aplicado nos motores [101, [111,

$$\tau = \hat{f} + K_4 e_c - \gamma \quad (\text{IV.56})$$

onde $\hat{f}(x)$ é uma estimativa da função não-linear do robô, que nesta aplicação será obtida através de uma *RNA*, e $y(t)$ representa um termo de robustecimento no sinal de controle necessário para compensar distúrbios não-estruturados e não-modelados. O termo $y(t)$ pode

ser obtido por várias técnicas, inclusive *sliding control*. Neste trabalho, $y(t)$ é obtido conforme (IV.68).

Substituindo-se (IV.56) em (IV.42), o sistema em malha fechada resultante é

$$\bar{M}\dot{e}_c = -(K_4 + \bar{V})e_c + \tilde{f}(x) - \bar{\mu} + \gamma \quad (\text{IV.57})$$

onde $\tilde{f}(x) = f(x) - \hat{f}(x)$ é o erro de estimação de $f(x)$, que afeta o erro da velocidade de rastreamento.

Portanto, descreve-se o uso de uma *RNA* para obtenção da estimativa da função $f(x)$.

A maior vantagem em se usar *RNA* neste caso deve-se a sua propriedade de aproximação e interpolação.

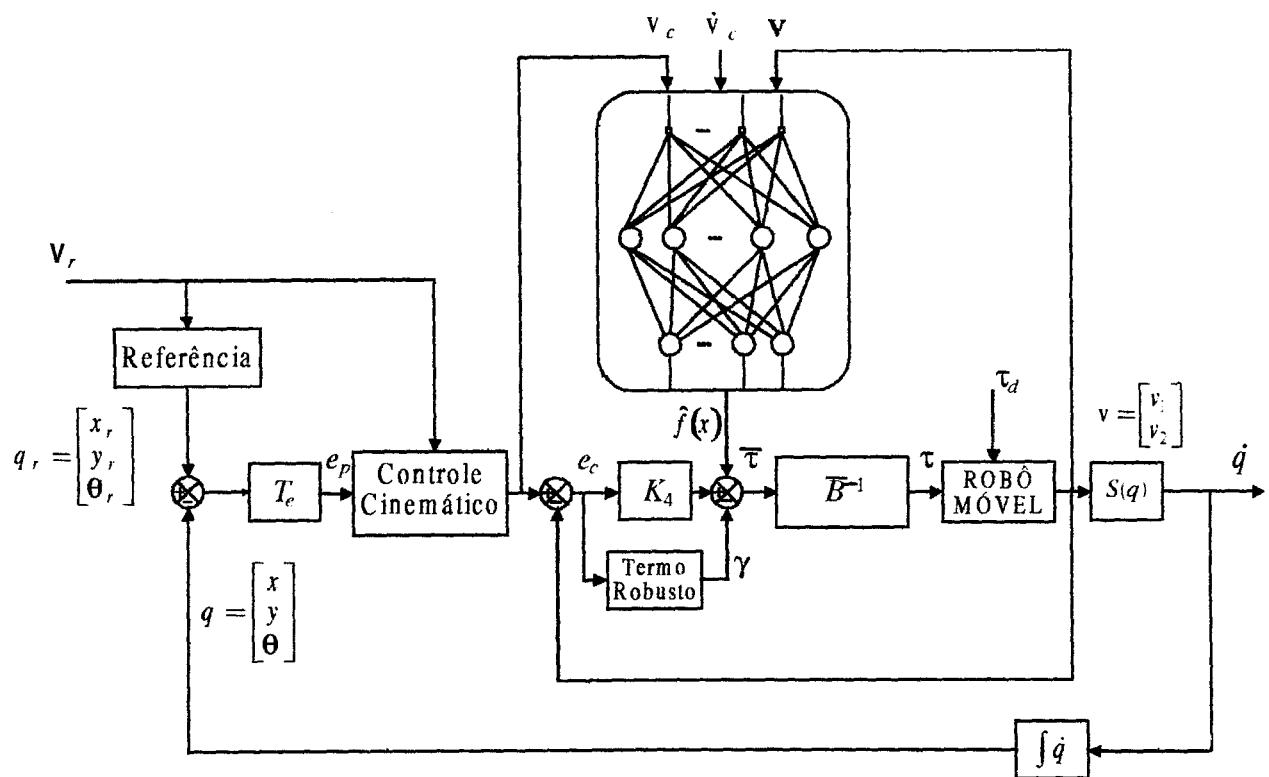


Figura IV.17. Estrutura do sistema de controle utilizando *RNA*.

Para o estabelecimento da lei de controle do *RMNH* através de *RNA's*, algumas definições são necessárias, conforme [11]:

Definição IM

Seja o erro de estimação da camada escondida para uma dada entrada x , definido por

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x) \quad (\text{IV.58})$$

A expansão em série de Taylor de $\sigma(x)$, em torno do ponto estimado z para um dado x , pode ser escrita como

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \sigma'(\hat{z})(V^T x - \hat{V}^T x) \Big|_{z=\hat{z}} + O(\tilde{V}^T x) \quad (\text{IV.59})$$

onde $\sigma'(z) \equiv \frac{\partial \sigma(z)}{\partial z} \Big|_{z=\hat{z}}$, sendo $\hat{z} = \hat{V}^T x$ a matriz Jacobiana e $O(\tilde{V}^T x)$ os termos de ordens

mais elevadas na série de Taylor, e $\tilde{V} = V - \hat{V}$. Definindo-se $\hat{\sigma}' = \sigma'(\hat{V}^T x)$, considerando-se (IV.58) e (IV.59), tem-se

$$\tilde{\sigma} = \sigma'(V^T x) \tilde{V}^T x + O(\tilde{V}^T x) \quad (\text{IV.60})$$

A expressão (IV.60) é importante pois substitui a expressão não-linear em \tilde{V} , dada em (IV.59) por uma expressão linear em V mais termos de ordens mais elevadas.

Utilizando-se uma RNA para aproximar $f(x)$ para o cálculo de controle em (IV.57), tem-se

$$\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_4 e_c - \gamma \quad (\text{IV.61})$$

Substituindo-se (IV.61) em (IV.57), obtém-se a equação para a dinâmica do erro de velocidade em malha fechada, isto é,

$$\bar{M} \dot{e}_c = -(K_4 + \bar{V}_m) e_c + W^T \sigma(V^T x) - \hat{W}^T \sigma(\hat{V}^T x) - \bar{\mu} + \varepsilon + \gamma \quad (\text{IV.62})$$

Adicionando-se e subtraindo-se $W^T \hat{\sigma}$ ao lado direito de (IV.62), obtém-se

$$\bar{M}\dot{e}_c = -(K_4 + \bar{V}_m)e_c + W^T \tilde{\sigma} - \tilde{W}^T \hat{\sigma} - \bar{\mu} + \varepsilon + \gamma \quad (\text{IV.63})$$

onde $\tilde{W} = W - \hat{W}$.

Agora, adicionando-se e subtraindo-se $\hat{W}^T \tilde{\sigma}$ ao lado direito de (IV.63), resulta

$$\bar{M}\dot{e}_c = -(K_4 + \bar{V}_m)e_c + \tilde{W}^T \tilde{\sigma} + \tilde{W}^T \hat{\sigma} + \hat{W}^T \tilde{\sigma} - \bar{\mu} + \varepsilon + \gamma \quad (\text{IV.64})$$

Substituindo-se (IV.60) em (IV.64), tem-se

$$\rho(t) = \varepsilon - \bar{\mu} + W^T O(\tilde{V}^T x) + \tilde{W}^T \hat{\sigma}' \cdot (V^T x) \quad (\text{IV.65})$$

onde $\rho(t)$ agrupa o erro de reconstrução da RNA $\varepsilon(x)$, os distúrbios e dinâmica não-modelada $\bar{\mu}$ e os termos de ordens elevadas na expansão em série de Taylor de $f(x)$. Deve-se notar que todos possuem exatamente a mesma influência: perturbações no sistema.

Lema IV.1

Conforme [11], os termos de perturbações em (IV.65) são limitados de acordo com

$$\|\rho(t)\| \leq \varepsilon_{\max} + \tau_{d,\max} + c_3 Z_m + c_6 Z_m \|\tilde{Z}\|_F + c_7 Z_m \|\tilde{Z}\|_F \|e_c\| \quad (\text{IV.66})$$

ou

$$\|\rho(t)\| \leq C_0 + C_1 \|\tilde{Z}\|_F + C_2 \|\tilde{Z}\|_F \|e_c\| \quad (\text{IV.67})$$

onde

ε_{\max} limitante do erro de reconstrução da RNA;

$\tau_{d,\max}$ limitante dos distúrbios dinâmicos, τ_d ;

\tilde{Z} $\tilde{Z} = Z - \hat{Z}$, é o erro de estimação de pesos Z , onde $Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix}$;

$\|\cdot\|$ norma de qualquer vetor ou matriz desejados. Quando houver $\|\cdot\|_F$, indica a norma de Frobenius, tal que, dada uma matriz $A = (a_{ij})$ a norma F (Frobenius) de A será

$$\|A\|_F^2 = \text{tr}(A^T A) = \sum_{i,j} a_{ij}^2 ;$$

Z_m limitante positivo de Z tal que $\|Z\|_F \leq Z_m$;

C_i 's constantes positivas conhecidas.

Somente resta agora determinar como se selecionar os termos de robustecimento $y(t)$ e os algoritmos de ajuste para os pesos da RNA que garantam estabilidade robusta e desempenho de rastreamento. A prova do Lema IV.1 é discutida em maiores detalhes em [34].

Definição IV.2

O termo de robustecimento y de (IV.57) é dado, conforme [11], por

$$\gamma(t) = -K_z (\|\hat{Z}\|_F + Z_M) e_c - e_c \quad (\text{IV.68})$$

onde $K_z > C_2$, com C_2 a constante conhecida em (IV.67).

Portanto, foram apresentados todos os subsídios necessários para enunciação do importante teorema para atualização dos pesos da RNA.

Teorema IV.1

Seja a equação dinâmica do *RMNH* descrita por (IV.38), a lei de controle do modelo cinemático de postura em coordenadas cartesianas em (IV.32) e a equação do cálculo do torque em (IV.56). Considerando-se o modelo da *RNA* descrita anteriormente, com as leis de atualização dos pesos conforme

$$\begin{cases} \dot{\hat{W}} = F \left[\hat{\sigma} e_c^T - \hat{\sigma} \hat{V}^T x e_c^T - (k \|e_c\| \hat{W})^T \right] \\ \dot{\hat{V}} = G (\hat{\sigma}^T W e_c x^T - k \|e_c\| \hat{V}) \end{cases} \quad (IV.69)$$

onde F e G são matrizes positivas definidas e $k > 0$, para um K_4 suficientemente grande em (IV.61), os erros de velocidade de rastreamento $e_c(t)$, o erro de posição $e_p(t)$ e as estimativas dos pesos da *RNA* \hat{V} e \hat{W} são uniformemente localmente limitados.

Prova: vide [11].

O resultado apresentado em [43] é semelhante ao discutido em [11], contudo com a diferença de que neste o controlador é projetado em coordenadas cartesianas, e naquele em coordenadas polares, cuja vantagem está em não se necessitar do conhecimento pontual da velocidade de referência para a estratégia de controle.

As simulações seguintes ilustram o procedimento para o projeto de um controlador implementável utilizando o modelo dinâmico em coordenadas cartesianas via redes neurais. As redes neurais desempenham o papel do ajuste fino para o rastreamento da trajetória pré-definida. As Figuras (IV.18)-(IV.23) apresentam o resultado para o estudo da trajetória tipo circuito fechado. A trajetória considerada apresenta velocidade linear constante igual a $v_r = 1 \text{ m/s}$ durante todo o trajeto e velocidade angular diferente de zero nas curvas, conforme apresentado na Seção IV.5.1. A Figura IV. 18 apresenta as trajetórias desejadas e obtidas para o robô. A trajetória de referência tem $p_r = (1 \ 1 \ 0)$ como postura inicial, e a postura de

partida para o robô é $p = (0 \ 0 \ \pi/2)$. Os ganhos do controlador cinemático em coordenadas cartesianas são ajustados experimentalmente como $K = (5 \ 4 \ 2)$, o ganho para o termo de robustecimento como $K_z = 0.001$, e $k_4 = 1.0$. A constante $k = 0.1$ é positiva e está incluída na lei de atualização dos pesos da *RNA*, $N_h = 9$ é o número de neurónios na camada escondida que apresentou os melhores resultados durante a implementação, $F = G = 0.5I_{10 \times 10}$ e $Z_M = 50$ é o limitante superior para a norma de Frobenius de Z . A Figura IV. 19 mostra o sinal de controle cinemático e a Figura IV.20 mostra o sinal de controle dinâmico, considerando os torques atuantes sobre o robe. A Figura IV.21 apresenta os sinais de erro, e a Figura IV.22 mostra a evolução das posturas desejadas e obtidas durante o movimento do robô. Finalmente, a Figura IV.23 mostra a evolução de alguns dos pesos da *RNA* durante o treinamento *on-line*.

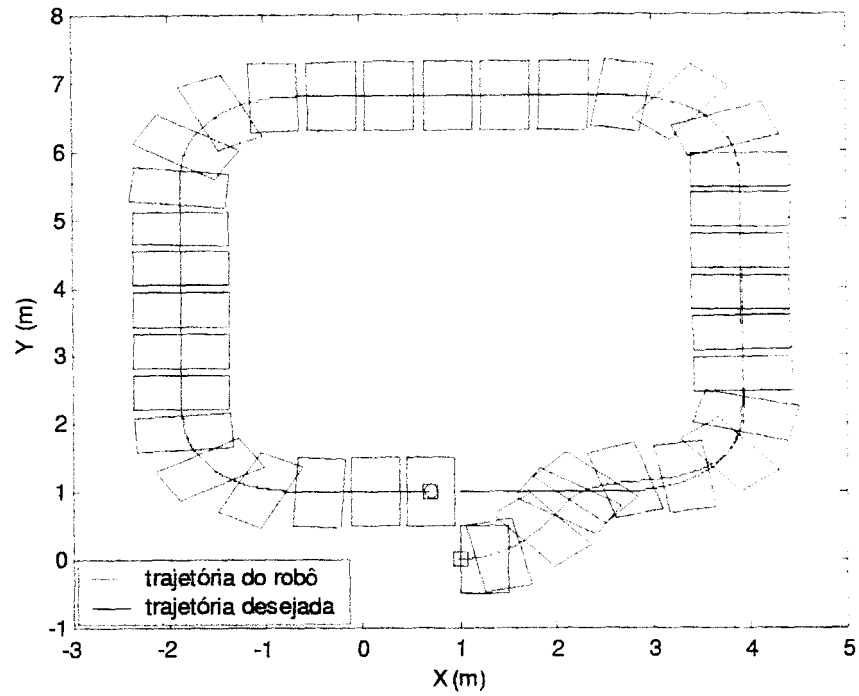


Figura IV.18. Trajetórias desejada e obtida para o robô, empregando-se o controlador utilizando redes neurais.

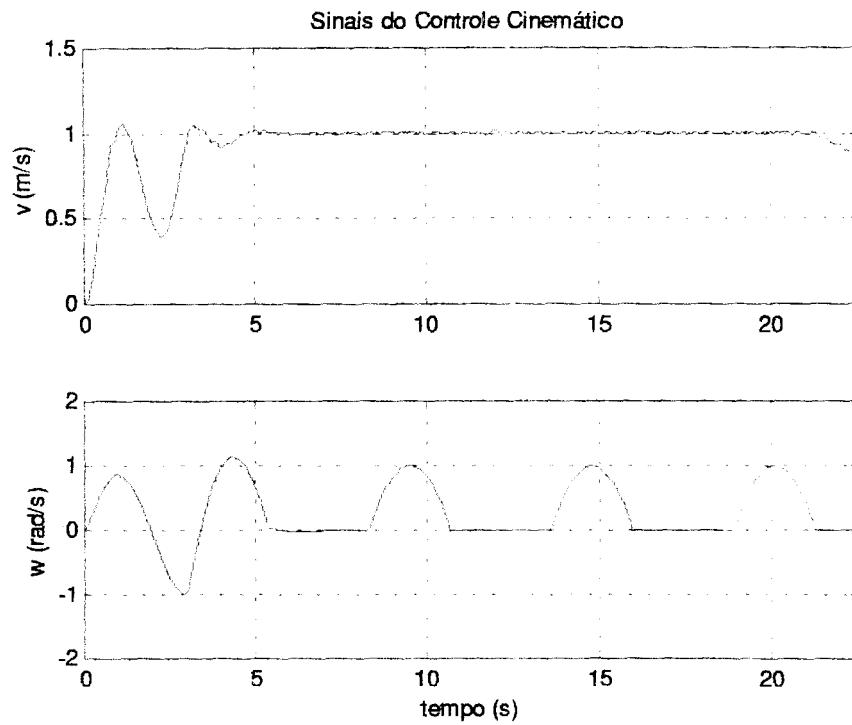


Figura IV.19. Sinais de controle para as velocidades linear e angular, obtidos conforme (IV.32).

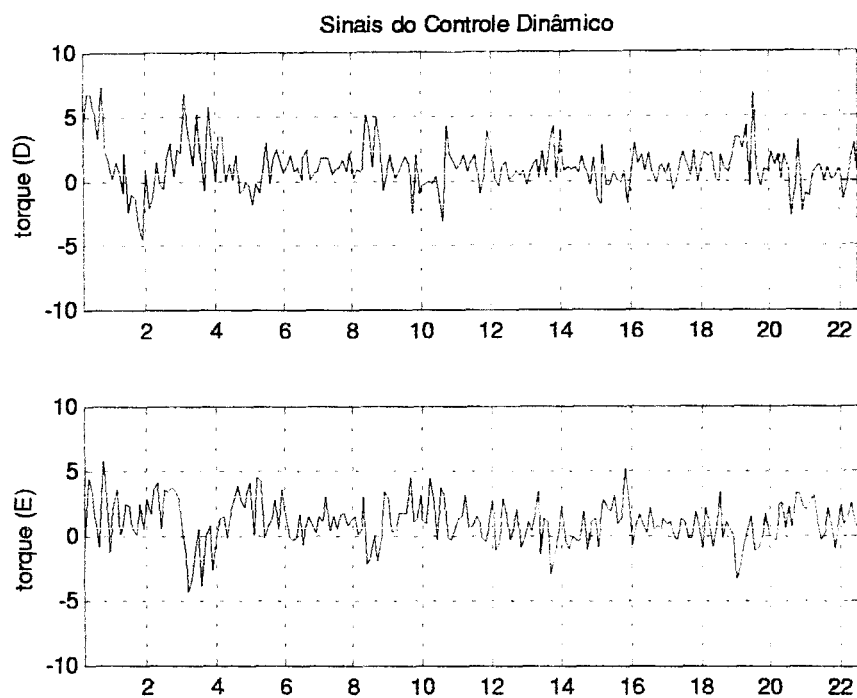


Figura IV.20. Sinais de controle dinâmico, considerando os torques atuantes sobre o robô.

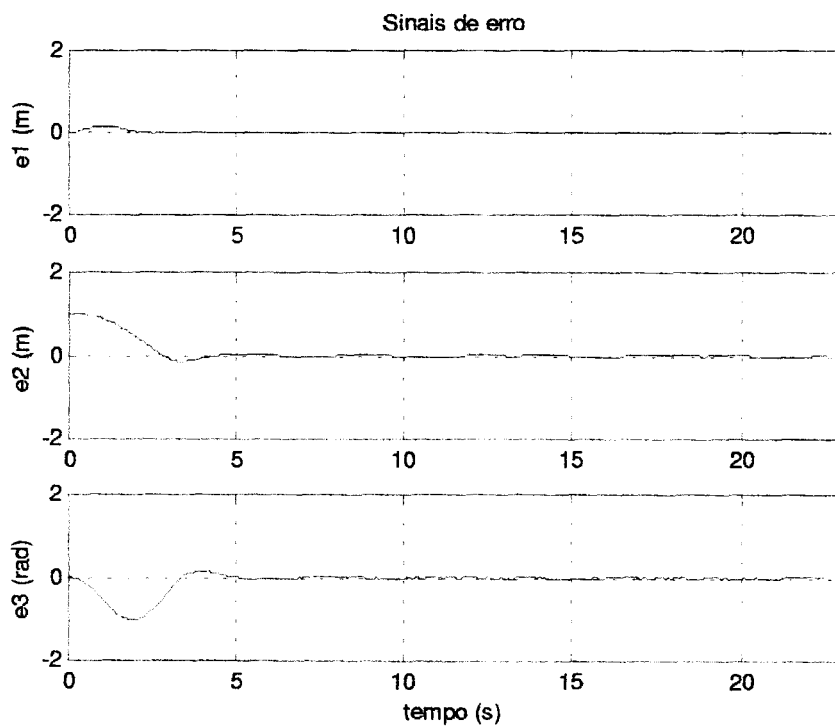


Figura IV.21. Sinais de erro resultantes da lei de controle aplicada ao robô.

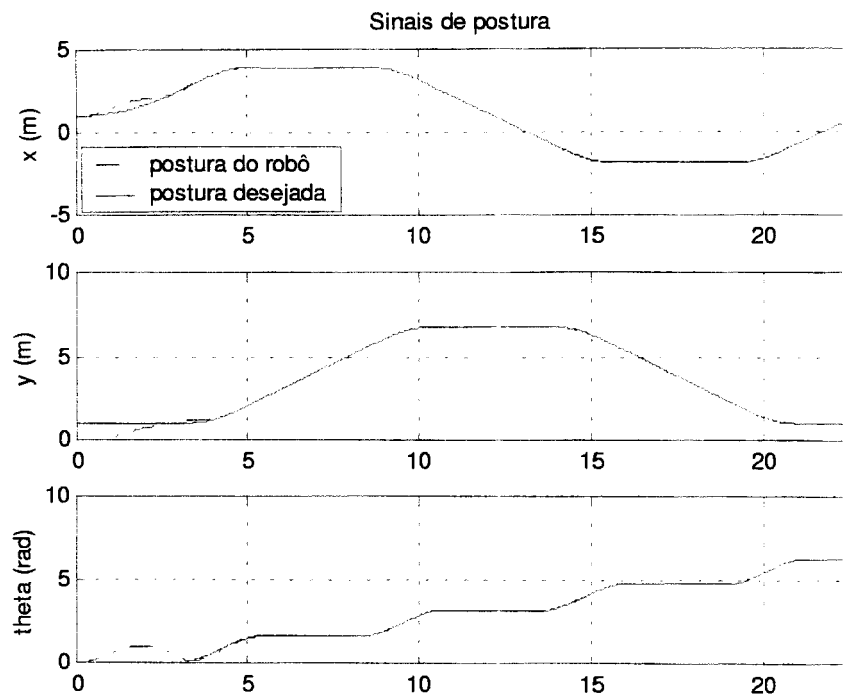


Figura IV.22. Evolução das posturas desejadas e obtidas, durante o movimento do robô.

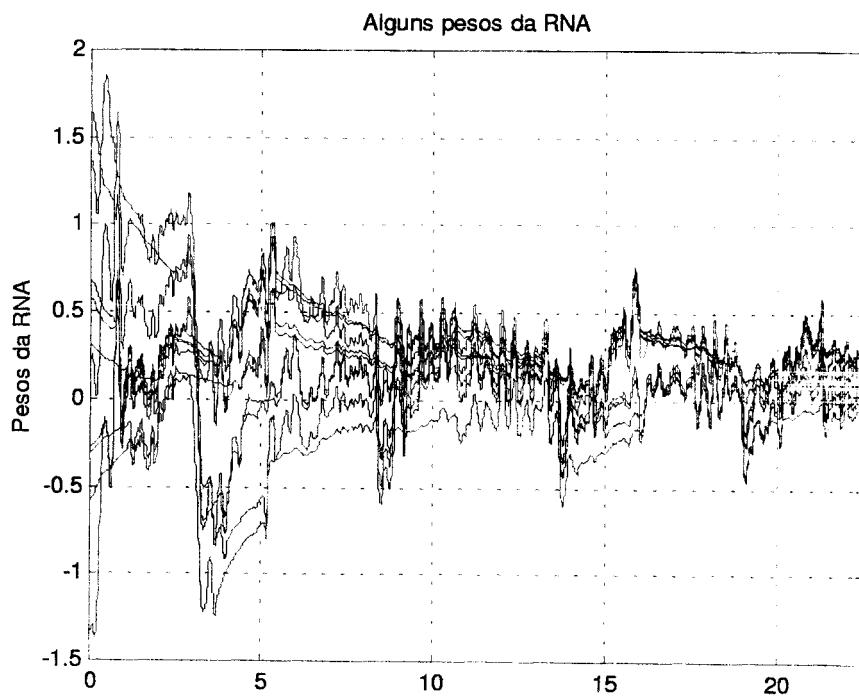


Figura IV.23. Evolução de alguns dos pesos da *RNA* durante o treinamento *on-line*.

A maior dificuldade verificada em se efetuar o controle se deve ao fato de se considerar os aspectos dinâmicos do *RMNH*, dado que o atrito de contato com a superfície e perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada são consideradas. A aproximação utilizando redes neurais permite um controle mais eficiente pois trata circunstâncias que o controle cinemático não considera, dadas suas simplificações como *tracking* perfeito de velocidade.

Adicionalmente, o controlador baseado em redes neurais tem um comportamento adaptativo bastante desejável, já que, conforme dito, pode tratar perturbações desconhecidas que incluem a dinâmica não modelada do tipo não estruturada. Algoritmos de atualização *on-line* dos pesos da *RNA*, que não requerem aprendizado *off-line*, garantem pequenos erros de rastreamento e sinais de controle limitados.

Os ganhos do controlador cinemático podem ser variados dentro de uma faixa de valores que não comprometa os valores máximos permitidos para os sinais de controle, conforme exposto na Seção IV.5.1. As matrizes positivas *F* e *G* definidas em (IV.69) não podem ter seus valores aumentados extremamente, pois a saída da rede neural seria ampliada, nem ter seus valores reduzidos drasticamente, o que causaria uma influência quase nula da rede neural.

O tempo de processamento é consideravelmente elevado em relação ao controle cinemático simples, devido a estimação *on-line* dos pesos da *RNA*.

IV.10 Comentários sobre a Integração das Técnicas de Visão

Computacional e Controle

A integração das técnicas de visão computacional, investigadas nos Capítulos II e III, com estratégia de controle baseada em *RNA* 's deste capítulo, vide Figura IV.17, pode ser efetuada mediante as seguintes considerações:

- As grandezas de entrada para o controle do Capítulo II são orientação (δ) e distância lateral (d) em relação à guia de navegação. Estas duas grandezas fornecem diretamente o erro de posição, que são dados em relação a θ e y na Figura IV.1. Isto sugere que o sistema de controle da Figura IV. 17, para rastreamento de uma trajetória, seja modificado para comportar os sinais de erro oriundos do procedimento descrito no Capítulo II. Os sinais de erro entrariam, na Figura IV. 17, onde está o vetor de erro de postura e_p , e o controle cinemático deveria ser revisto para os novos referenciais e uma análise deveria ser feita para o algoritmo de atualização dos pesos da *RNA*. Considerando-se o projeto do controlador para seguimento de caminho, conforme Seção IV.4.2, as informações oriundas do sistema de guiamento, orientação (δ) e distância lateral (d) em relação à guia de navegação são as informações necessárias para a entrada do controle $v_c(t)$.
- Com relação ao Capítulo III, as grandezas disponíveis para o controle são orientação (ϕ) e distância relativa (d/T_z). Neste caso, também tem-se o erro de posição em relação a θ e y na Figura IV.1. Apesar de o fluxo óptico não fornecer a distância absoluta em relação a uma parede ou obstáculo, pode-se monitorar o tempo de colisão em relação às paredes direita e esquerda para a situação onde se quer que o robô navegue no centro de um corredor, obtendo-se assim, sua posição em relação à referência. Portanto, os erros de posição em y e de orientação em θ podem ser fornecidos da mesma forma que no procedimento do Capítulo II, e a mesma análise pode ser feita com relação aos sinais de

erro, ao controlador cinemático, e ao algoritmo de atualização dos pesos da *RNA*, e para o projeto do controlador para seguimento de caminho, conforme Seção IV.4.2.

Convém ressaltar que nos dois casos anteriores será necessário efetuar transformação de coordenadas de modo a tratar o fato de que o centro óptico da câmera não se encontra no ponto de guiamento do robô móvel. Com relação às hipóteses da Seção 11.6, elas serão violadas quando o robô executar rotação, devido a não coincidência dos centros de rotação do robô e do centro óptico da câmera. Mesmo com esta transformação, movimentos do robô podem causar deslocamentos do centro óptico que violam tais hipóteses.

Para aplicações que necessitem do conhecimento da postura global, duas alternativas poderiam ser utilizadas:

- a) como em [381], onde um subsistema de processamento de imagens estima a localização absoluta do veículo, utilizando uma câmera *onboard* e postes sinalizadores com códigos de barra ao longo da trajetória. Um computador armazena os padrões de código de postes e sua posição absoluta $(x \sim y \sim A)$. A câmera fixa ao veículo obtém as imagens continuamente e quando dois postes são detectados, o computador estima a posição e orientação do veículo por iriangularização. Este procedimento permite a obtenção do posicionamento global do veículo a partir das imagens obtidas pela câmera, com o auxílio de postes sinalizadores.
- b) Através da integração sensorial, onde as informações de outros sensores são consideradas, por exemplo, os sensores inerciais e/ou *GPS*.

Tais alternativas, porém, não constituem objeto de estudo deste trabalho.

Capítulo V

Conclusões

Esta tese investigou dois importantes assuntos concernentes à robótica móvel: guiamento e controle. Sobre guiamento foram apresentados algoritmos de processamento de imagens em tempo real, sendo a informação visual utilizada para detecção de características necessárias para posterior controle. Duas diferentes técnicas de guiamento visual foram investigadas, uma baseada na detecção de guias, modeladas por um polinômio de segunda ordem, e outra na técnica de fluxo óptico, que relaciona as velocidades aparentes entre cena e observador. No que se refere ao controle, foram investigadas estratégias que consideram tanto as características cinemáticas quanto dinâmicas do modelo, incluindo atrito com a superfície e perturbações que incluem a dinâmica não modelada do tipo não estruturada.

Para o algoritmo proposto de detecção de guias, baseado no *KIF*, obteve-se bons resultados, porém sem se considerar os eventuais efeitos do sistema de controle, uma vez que a integração não foi efetuada. Estabeleceu-se assim um procedimento com bom desempenho em condições de teste semelhantes a [41]. Esta estratégia mostrou uma vantagem em relação a [41], uma vez que requer menor intervenção humana e necessita de menos informação inicial, fator bastante desejado em navegação de robôs. Todo o processamento do algoritmo de detecção de guias é feito em tempo real, com imagens sendo capturadas sem considerar a

malha de controle e guiamento do robô, diferentemente de [23], onde as imagens são estáticas ou em [28], onde se requer processamento *off-line*. O processamento *on-line* sugere maior flexibilidade ao algoritmo, dado que poderá informar ao sistema de controle a postura atual e assim poder-se-á tomar as ações adequadas. O algoritmo recursivo proposto nesta tese para a atualização das estimativas dos coeficientes das parábolas foi o *KIF*, que apresentou vantagens de desempenho e velocidade computacional em relação ao *WRLS*, proposto em [41].

Em relação à técnica apresentada no Capítulo III, o algoritmo de detecção de guias para navegação do Capítulo II apresenta as seguintes vantagens: a) uma taxa de processamento de imagens maior, pois envolve a extração de informação local, representada pelas estimativas dos coeficientes das parábolas que interpolam as guias. Este processamento não necessita armazenar uma sequência de imagens em memória, sendo efetuado a cada nova imagem; b) lida melhor com ambientes pouco estruturados do tipo estudado, isto é, guias próximas do plano $Y = -H$, do que a técnica que utiliza fluxo óptico, pois neste caso, requer-se iluminação adequada e ambiente com boa textura, além de se considerar que as superfícies em questão apresentem padrão de brilho constante, o que na prática constitui hipótese restritiva; c) a dependência da correlação entre as imagens é menor, pois a cada nova imagem o processamento atualiza os parâmetros que caracterizam as guias, e no caso do fluxo óptico deve haver uma correlação forte entre dois quadros em sequência para que o fluxo seja adequadamente estimado. Mais precisamente, o algoritmo do Capítulo 11 não está limitado a um deslocamento de poucos pixels entre quadros.

Em relação ao controle de robôs móveis, foi investigado o projeto dos controladores visando sua implementação, e as leis de controle propostas em [11] e [44] foram analisadas tendo-se o cuidado de se ajustar os ganhos do controlador, de forma a se manter um compromisso entre a velocidade de resposta do controlador e o fato de ser implementável.

Outro fator importante no projeto de controladores de rastreamento implementáveis foi o cuidado com a geração da trajetória de referência. Foram geradas trajetórias contínuas e suaves para que o rastreamento fosse possível. Os parâmetros físicos usados nas simulações foram do robô *MagellanTM*. Para o controle dinâmico utilizando *RNA's*, todos os pesos e bias também foram ajustados *on-line*, sem necessidade de um treinamento anterior, o que eleva a autonomia do controle.

Como sugestões para futuros trabalhos, convém mencionar:

1. Simulação da integração das técnicas de visão computacional e de controle, objetivando avaliar a influência dos parâmetros de projeto, tanto do sistema de visão quanto do sistema de controle;
2. Implementação e integração das estratégias apresentadas neste trabalho no robô *MagellanTM* para aferição prática dos resultados obtidos;
3. Estudo da integração de outros sensores (*dead-reckoning*, bússola, GPS, *laser range-finder*, etc.) com o sensor de visão;
4. Estudo da utilização de visão estéreo (duas câmeras ao invés de uma), para se elevar a qualidade e confiabilidade das informações visuais; e
5. Estudo de novas leis de controle para o modelo dinâmico de robôs móveis, considerando-se um número menor de restrições.

Capítulo VI

Referências Bibliográficas

- [1] AICARDI, M.; CASALINO, G.; BICCHI, A., BALESTRINO, A. Closed loop steering of unicycle-like vehicles via Lyapunov techniques. *IEEE Robotics & Automation Magazine*, v. 2, n. 1, p. 27-35, mar. 1995.
- [2] ASAMI, S. Robots in Japan: present and future. *IEEE Robotics & Automation Magazine*, v. 1, n. 2, p. 22-26, jun. 1994,
- [3] ANDERSON, B. D. **0.**; **MOORE, J. B.** *Optimal filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [4] BARRON, J. L.; FLEET, D. J.; BEAUCHEMIN, S. S. Performance of optical flow techniques. *International Journal of Computer Vision*, v. 12, n. 1, p. 43-77, 1994.
- [5] BIERMAN, G. J. *Factorization methods for discrete sequential estimation*. New York: Academic Press, 1977.
- [6] BROWN, R. G.; HWANG, P. Y. C. *Introduction to random signals and applied Kalman filtering*. USA: John Wiley & Sons Inc., 1992.

- 171 CARELLT, R.; SECCHI, H.; MUT, V.; NASIST, O. Algoritmos estables para la navegacion de robots moviles en passilos usando el flujo optico, In: 8th WORKSHOP ON INFORMATION PROCESSING AND CONTROL, 1999, Mar del Plata. **Anais...** Argentina, sep. 1999, v. 2, p. 79-7 a 86-7.
- [8] CHEN, G.Y.; TSAI, W.H. An incremental-learning-by-navigation approach to vision-based autonomous land vehicle guidance in indoor environments using vertical line information and multiweighted generalized Hough transform technique. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*. v. 28, n. 5, p. 740-748, oct 1998.
- [9] DEV, A.; **KROSE, B.**; GROEN, F. Navigation of a mobile robot on the temporal development of the optic flow. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS'97), 1997, Grenoble. **Anais...** France: IEEE, sep. 1997, p. 558-563.
- [10] FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot: backstepping kinematics into dynamics. *Journal of Robotic Systems*, v. 14(3), p. 149-163, 1997.
- [11] FIERRO, R. LEWIS, F. L. Control of a nonholonomic mobile robot using neural networks. *IEEE Transactions on Neural Networks*, v. 9, n. 4, p. 589-600, jul 1998.
- [12] FOLEY, J. D.; DAM, A.V. *Fundamentals of interactive computer graphics*. Reading, MA: Addison-Wesley, 1982.
- [13] FU, K. S.; GONZALEZ, R. C.; LEE, C. S. G. *Robotics: control, sensing, vision and intelligence*. New York: McGraw-Hill Inc., 1987.
- [14] GIACHETTI, A.; CAMPANI, M.; **TORRE, V.** The use of optical flow for road navigation. *IEEE Transactions on Robotics and Automation*, v. 14, n. 1, p. 34-48, feb. 1998.

- [15] GOODWIN, G. C.; PAYNE, R. L. *Dynamic system identification: experiment design and data analysis*. New York: Academic Press, 1977.
- [16] GOODWIN, G. C.; SIN, K. S. *Adaptive filtering prediction and control*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [17] HAYKIN, S. *Neural networks - a comprehensive foundation*. Upper Saddle River, NJ: Prentice-Hall, 1991.
- [18] HEMERLY, E. M. *Controle por computador de sistemas dinâmicos*. São Paulo, SP: Edgard Blucher LTDA. , segunda edição, 2000.
- [19] HEMERLY, E. M.; RODRIGUES, C.C. Guiagem de veículos autônomos utilizando sensor de visão. In: 10^o CONGRESSO BRASILEIRO DE AUTOMATICA, 1994, Rio de Janeiro. **Anais...** Rio de Janeiro: CBA, 1994, p. 873-878.
- [20] HORN, B. K. P. *Robot vision*. Massachusetts: The MIT Press. McGraw-Hill Inc., 1986.
- [21] HORN, B. K. P.; SCHUNCK, B.G., "Determining optical flow": a retrospective. *Artificial Intelligence*, v. 59, p. 81-87, 1993.
- [22] HORN, B. K. P.; SCHUNCK, B.G. Determining optical flow, *Artificial Intelligence*, v. 17, p. 185-203, 1981.
- [23] RU, Z.; UCHIMURA, K. Action-based road horizontal shape recognition. *SBA Controle & Automação*, v. 10, n. 2, p. 83-88, 1999.
- [24] JAIN, R.; KASTURI, R.; SCHUNCK, B. G. *Machine vision*. New York: McGraw-Hill Inc., 1995.
- [25] JIANG, Z. P.; N IJMEIJER, H. Tracking control of mobile robots: a case study in backstopping. *Automatica*, v 33, n. 7, p. 1393-1399, 1997.

- [26] JAZWINSKI, A. H. *Stochastic processes and filtering theory*. New York: Academic Press, 1970.
- [27] KANAYAMA, Y.; KIMURA, Y.; MIYAZAKI, F.; NOGUCHI, T. A stable tracking control method for an autonomous mobile robot. In: INTERNATIONAL CONFERENCE OF ROBOTICS AND AUTOMATION, 1990. **Anais...** IEFE, 1990. v.39, p. 384-389.
- [28] KANBARA, T.; MIURA, J.; SHIRAI, Y. Selection of efficient landmarks for an autonomous vehicle. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS-PART 2. **Anais..**- jul. 1993, p. 1332-1338.
- [29] KHALIL, H. K. *Nonlinear systems*, 2nd. ed. Upper Saddle River, NJ: Prentice Hall, 1996.
- [30] KLUGE, K.; THORPE, C. Representation and recovery of road geometry in YARF In: PROCEEDINGS OF THE INTELLIGENT VEHICLES' 92 SYMPOSIUM. **Anais...** 1993, p. 114-119.
- [31] KREUCHER, C.; LAKSHMANAN, S. LANA: A lane extraction algorithm that uses frequency domain features. *IEEE Transactions on Robotics and Automation*, v. 15, n. 2, p. 343-350, apr. 1999.
- [32] LAGES, W. F. *Controle e estimação de posição e orientação de robôs móveis*. (Tese de Doutorado), São Jose dos Campos, Instituto Tecnológico de Aeronáutica -- ITA. 1998.
- [33] LAI, S. H.; VEMURI, B. C. Robust and efficient computation of optical flow.. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION **Anais...** IEEE, nov. 1995, p. 455-460.

- [34] LEWIS, F. L.; YESILDIREK, A.; LILT, K. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Transactions on Neural Networks*, v. 7, n. 2, p. 388-399, mar. 1996.
- [35] *MAGELLANIM pro compact mobile robot user's guide*. IS Robotics Inc., Real World Interface Division, New Hampshire: USA, 2000.
- [36] MENÉNDEZ, J.M.; GARCIA, N.; SALGADO, L.; RENDÓN, E. An algorithm for FOE localization. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING. **Anais...** IEEE, v. 3, sep. 1996, p. 811-814.
- [37] MORF, M.; KAILATH, T. Square root algorithms for least-squares estimation. *IEEE Transactions on Automatic Control*, v. AC-20, n. 4, p. 487-497, aug. 1975.
- [38] MURATA, S.; HIROSE, T. Onboard locating system using real-time image processing for a self-navigating vehicle. *IEEE Transactions on Industrial Electronics*, v. 40, n. 1, p. 145-154, feb. 1993.
- [39] NASCIMENTO JR., C. L.; YONEYAMA, T. *Inteligência artificial em automação e controle*. Sao Paulo, SP: Edgard Blucher LTDA. , 2000.
- [40] PIMENTEL, J. C.; HEMERLY, E. M. An algorithm to detect lanemarks for mobile robots navigation. In: IV INDUSTRY APPLICATIONS CONFERENCE, 2000, Porto Alegre. **Anais...** Porto Alegre: INDUSCON, 2000 [no prelo].
- [41] SCHNEIDERMAN, H.; NASHMAN, M. A discriminating feature tracker for vision-based autonomous driving. *IEEE Transactions on Robotics and Automation*, v. 10, n. 6, p. 769-775, dec. 1994.
- [42] SCHRAFT, R. D. Mechatronics and robotics for service applications. *IEEE Robotics & Automation Magazine*, v. 1, n. 4, p. 31-35, dec. 1994.

- [43] SOUSA, JR, C.; HEMERLY, E. M. Controle de robôs móveis em coordenadas polares via rede neural. In: XIII CONGRESSO BRASILEIRO DE AUTOMATICA, 2000, Florianópolis. **Anais...** Florianópolis: CBA, 2000 [no prelo].
- [44] SOUSA, JR, C. *Controle adaptativo de robôs móveis*. (Tese de Mestrado), São José dos Campos, Instituto Tecnológico de Aeronáutica – ITA. 2000.
- [45] SLOTINE, J-J. E.; LI, W. *Applied nonlinear control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [46] WALTON, J. M. *Terrain mapping from a mobile platform through optical flow techniques*. (Tese de Mestrado), Massachussets, Massachusetts Institute of Technology - MIT. 1995.
- [47] WIT , C. C.; SORTALEN, O. J. Exponential stabilization of mobile robots with nonholonomic constraints. *IEEE Transactions on Automatic Control*, v. 37, n. 2, p. 1326-1332, nov. 1992.
- [48] YAMAMOTO, Y.; YUN, X. Coordinating locomotion and manipulation of a mobile. manipulator. *IEEE Transactions on Automatic Control*, v. 39, n. 6, p. 1326-1332, jun. 1994.
- [49] ZURADA, J. M. *Introduction to artificial neural systems*. St. Paul: West Publishing Company, 1992.