

Tese apresentada à Divisão de Pós-Graduação do Instituto Tecnológico de Aeronáutica como parte dos requisitos para obtenção do título de Doutor em Ciência no Curso de Engenharia Eletrônica e Computação na Área de Sistemas e Controle.

Walter Fetter Lages

Controle e Estimação de Posição e Orientação de Robôs Móveis

Tese aprovada em sua versão final pelos abaixo assinados

Prof. Dr. Elder M. Hemerly
Orientador

Prof. Dr. Paulo Afonso de Oliveira Soviero
Chefe da Divisão de Pós-Graduação

Campo Montenegro

São José dos Campos, SP, Brasil

1998

Controle e Estimação de Posição e Orientação de Robôs Móveis

Walter Fetter Lages

Composição da Banca Examinadora:

Prof. Dr. Cairo Lúcio Nascimento Jr. **Presidente** - ITA - IEES
Prof. Dr. Elder M. Hemerly **Orientador** - ITA - IEES
Prof. Dr. Luiz Carlos Sandoval Góes..... ITA - IEMP
Prof. Dr. Márcio Rillo USP - PEE
Prof. Dr. Paulo Roberto Gardel Kurka..... UNICAMP - FEM

Índice Geral

Capítulo I

Introdução.....	1
-----------------	---

Capítulo II

Modelagem Cinemática e Dinâmica	9
II.1. Modelo Cinemático de Postura.....	10
II.1.1. Descrição das Rodas	11
II.1.1.1. Rodas Convencionais Fixas.....	12
II.1.1.2. Rodas Convencionais Orientáveis Centradas.....	14
II.1.1.3. Rodas Convencionais Orientáveis Não-centradas.....	14
II.1.1.4. Rodas Suecas.....	15
II.1.2. Restrições à Mobilidade do Robô	17
II.1.3. Modelo Cinemático de Postura no Espaço de Estados	21
II.2. Modelo Cinemático de Configuração	24
II.3. Modelo Dinâmico de Configuração.....	26
II.4. Modelo Dinâmico de Postura	30
II.5. Conclusão.....	31

Capítulo III

Controle Linearizante.....	32
III.1. Linearização Entrada-Saída por Realimentação Estática de Estados.....	33
III.2. Ponto de Referência.....	36
III.3. Realimentação Linearizante para a Classe (3,0).....	37
III.4. Realimentação Linearizante para a Classe (2,0).....	39
III.5. Realimentação Linearizante para a Classe (2,1).....	41

III.6. Realimentação Linearizante para a Classe (1,1).....	43
III.7. Realimentação Linearizante para a Classe (1,2).....	43
III.8. Resultados de Simulação	43
III.9. Resultados de Tempo Real	53
III.10. Conclusão	57
 Capítulo IV	
Controle por Transformação Descontínua	59
IV.1. Transformação de Coordenadas	59
IV.2. Leis de Controle	62
IV.2.1. Classe (3,0)	62
IV.2.2. Classe (2,0)	64
IV.2.3. Classe (2,1)	66
IV.2.4. Classe (1,1)	67
IV.2.5. Classe (1,2)	68
IV.2.6. Resultados de Simulação	70
IV.3 <i>Backstepping</i> Utilizando PID	73
IV.3.1. Motivação	73
IV.3.2. Resultados de Tempo Real	74
IV.4. <i>Backstepping</i> Utilizando Controle Linearizante	78
IV.5. Conclusão	78
 Capítulo V	
Controle Adaptativo	80
V.1. Versão Adaptativa do Controle Linearizante	82
V.1.1. Parametrização do Modelo de Robôs Móveis	85
V.1.2. Resultados de Simulação	89

V.1.3. Resultados de Experimentais	92
V.2. Versão Adaptativa do Controle por Transformação Descontínua.....	95
V.2.1. Parametrização do Modelo	96
V.2.2. Lei de Adaptação	98
V.2.3. Resultados de Simulação	104
V.3. Conclusão	109
Capítulo VI	
Estimação de Posição e Orientação.....	110
VI.1. Método de <i>Dead-Reckoning</i>	115
VI.2. Visão Computacional	118
VI.2.1. Fusão de Dados.....	121
VI.2.2. Resultados de Simulação	126
VI.3. Bússola Digital	126
VI.3.1. Resultados de Simulação	128
VI.3.2. Resultados Experimentais	130
V.4. Conclusão	131
Capítulo VII	
Implementação	132
VII.1. Descrição do <i>Hardware</i> do IEEE-Rover	132
VII.2. Descrição do <i>Software</i> do IEEE-Rover.....	137
VII.3. O Robô Móvel Twil.....	143
VII.4. Conclusão.....	147
Capítulo VIII	
Conclusões	148
Referências Bibliográficas	152

Índice de Figuras

Fig. II-1	Definição dos sistemas de coordenadas	11
Fig. II-2	Definição do sistema de coordenadas da roda convencional fixa.....	13
Fig. II-3	Roda convencional orientável não-centrada	15
Fig. II-4	Definição do sistema de coordenadas para roda sueca	16
Fig. III-1	- Diagrama de blocos do controle linearizante.....	36
Fig. III-2	Sistema de coordenadas do ponto de referência para controle do robô	36
Fig. III-3	Simulação da trajetória realizada pelo robô com acesso completo aos estados.....	44
Fig. III-4	Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover.....	45
Fig. III-5	Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover.....	45
Fig. III-6	Resposta de orientação para o robô IEEE-Rover.....	46
Fig. III-7	Trajetória de referência (sólida) e trajetória realizada pelo robô (tracejada) com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$	47
Fig. III-8	Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$	47
Fig. III-9	Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$	48
Fig. III-10	Resposta de orientação para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$	48
Fig. III-11	Torque aplicado na roda direita (sólida) e torque aplicado na roda esquerda (tracejada) com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$	49

Fig. III-12 Trajetória de referência (sólida) e trajetória realizada pelo robô (tracejada) com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$	50
Fig. III-13 Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$	51
Fig. III-14 Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$	51
Fig. III-15 Resposta de orientação para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$	52
Fig. III-16 Torque aplicado na roda direita (sólida) e torque aplicado na roda esquerda (tracejada) com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$	52
Fig. III-17 Trajetória espacial em tempo real para o robô Twil em tarefa de <i>parking</i>	53
Fig. III-18 Resposta em tempo real de $x_r(t)$ para o robô Twil em tarefa de <i>parking</i>	54
Fig. III-19 Resposta em tempo real de $y_r(t)$ para o robô Twil em tarefa de <i>parking</i>	54
Fig. III-20 Resposta em tempo real de orientação para o robô Twil em tarefa de <i>parking</i>	55
Fig. III-21 Trajetória espacial de referência (traço ponto), trajetória de tempo real (sólida) e trajetória simulada (tracejada) em tarefa de rastreamento	55
Fig. III-22 Referência (traço ponto), resposta de tempo real (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô Twil em tarefa de rastreamento	56
Fig. III-23 Referência (traço ponto), resposta de tempo real (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô Twil em tarefa de rastreamento	56
Fig. III-24 Resposta de tempo real (sólida) e resposta simulada (tracejada) de orientação para o robô Twil em tarefa de rastreamento	57
Fig. IV-1 Definição das variáveis em coordenadas polares	60

Fig. IV-2	Trajetória espacial para um robô de cada classe	71
Fig. IV-3	Resposta de $\theta(t)$ para um robô de cada classe.....	71
Fig. IV-4	Resposta de $x_c(t)$ para robô classe (1,2)	72
Fig. IV-5	Resposta de $y_c(t)$ para robô classe (1,2)	72
Fig. IV-6	Resposta de $\theta(t)$ para robô classe (1,2)	76
Fig. IV-7	Controlador com <i>backstepping</i>	74
Fig. IV-8	Trajetória espacial em tempo real (contínua) e simulada (tracejada) para um robô classe (2,0)	76
Fig. IV-9	Resposta de $\theta(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0)	76
Fig. IV-10	Resposta de $\eta_1(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0)	77
Fig. IV-11	Resposta de $\eta_2(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0)	77
Fig. VI-12	<i>Backstepping</i> com controlador linearizante	78
Fig. V-1	Diagrama de blocos do controle linearizante adaptativo	89
Fig. V-2	Saída do modelo de referência (tracejada) e posição simulada do robô na direção X_0 (sólida)	90
Fig. V-3	Saída do modelo de referência (tracejada) e posição simulada do robô na direção Y_0 (sólida)	90
Fig. V-4	Evolução simulada de $\bar{\Theta}_1(t)$ (tracejada) e $\bar{\Theta}_2(t)$ (sólida).....	91
Fig. V-5	Evolução simulada da diagonal da matriz de covariância	91

Fig. V-6	Saída do modelo de referência (tracejada), posição do robô na direção X_0 com controlador adaptativo (sólida) e posição do robô na direção X_0 com controlador não-adaptativo (pontilhada).....	93
Fig. V-7	Saída do modelo de referência (tracejada), posição do robô na direção Y_0 com controlador adaptativo (sólida) e posição do robô na direção Y_0 com controlador não-adaptativo (pontilhada).....	93
Fig. V-8	Evolução de $\bar{\Theta}_1(t)$ e $\bar{\Theta}_2(t)$ em tempo real.....	94
Fig. V-9	Evolução da diagonal da matriz de covariância em tempo real.....	94
Fig. V-10	Diagrama de blocos do controle por transformação descontínua adaptativo.....	95
Fig. V-11	Simulação da trajetória espacial do robô utilizando controle por transformação descontínua.....	105
Fig. V-12	Posição simulada do robô na direção X_0 em função do tempo.....	105
Fig. V-13	Posição simulada do robô na direção Y_0 em função do tempo.....	106
Fig. V-14	Orientação simulada do robô em função do tempo.....	106
Fig. V-15	Evolução simulada de $\bar{\Theta}_1(t)$ (sólida) e $\bar{\Theta}_2(t)$ (tracejada).....	107
Fig. V-16	Evolução simulada de $\bar{\Theta}_3(t)$ (sólida) e $\bar{\Theta}_4(t)$ (tracejada).....	108
Fig. V-17	Evolução simulada de $P_{11}(t)$ (sólida) e $P_{22}(t)$ (tracejada).....	108
Fig. V-18	Evolução simulada de $P_{33}(t)$ (sólida) e $P_{44}(t)$ (tracejada).....	109
Fig. VI-1	Trajetória circular utilizada para desenvolvimento do algoritmo de <i>dead-reckoning</i>	117
Fig. VI-2	Trajetória do robô (pontilhada) e trajetória estimada por <i>dead-reckoning</i> (sólida).....	118
Fig. VI-3	<i>Landmark</i> utilizado pelo sistema de visão.....	119

Fig. VI-4	Diagrama de blocos da fusão de dados obtidos por visão computacional e <i>encoders</i>	125
Fig. VI-5	Trajetória de referência (sólida) e trajetória estimada por fusão de dados (tracejada).....	126
Fig. VI-6	Diagrama de blocos da fusão de dados da bússola digital e <i>encoders</i>	128
Fig. VI-7	Simulação do odômetro e da fusão de dados odômetro/bússola.....	129
Fig. VI-8	Simulação da orientação	129
Fig. VI-9	Posição calculada pelo odômetro e pela fusão odômetro/bússola em tempo real.....	130
Fig. VI-10	Orientações obtidas em tempo real	131
Fig. VII-1	Fotografia do IEEE-Rover	132
Fig. VII-2	Diagrama de blocos do IEEE-Rover.....	133
Fig. VII-3	Disco de <i>encoder</i> e sinais herados	135
Fig. VII-4	Diagrama de temporização da bússola.....	138
Fig. VII-5	Diagrama de temporização dos <i>encoders</i>	139
Fig. VII-6	Períodos de amostragem da bússola e dos <i>encoders</i>	142
Fig. VII-7	Temporização da bússola e da fusão de dados.....	143
Fig. VII-8	- Fotografia do robô móvel Twil	144
Fig. VII-9	Diagrama de blocos do robô móvel Twil.....	145
Fig. VII-10	Amostra em tempo real dos sensores do robô Twil, acessado através da Internet utilizando-se um <i>browser</i> convencional.	147

Índice de Tabelas

Tab. II-I	Modelos cinemáticos de postura	22
Tab. II-II	Número mínimo de motores para cada classe de robô móvel.....	29
Tab. IV-I	Modelos em coordenadas polares	61

Relação de Símbolos

a_x	Semi-eixo horizontal do <i>landmark</i> ;
a_y	Semi-eixo vertical do <i>landmark</i> ;
α	Ângulo entre o segmento $\overline{P_0 P_w}$ e a direção X_c ;
α	Diferença entre a orientação do robô e o ângulo do erro de posição;
α_a	Ganho de adaptação;
α_i	Coeficientes do modelo de referência;
b	Distância entre as rodas e o eixo de simetria;
$B(x)$	Dinâmica do modelo cinemático de postura;
β	Ângulo das rodas orientáveis;
β_c	Ângulo das rodas orientáveis centradas;
β_{oc}	Ângulo das rodas orientáveis não-centradas;
d	Distância entre o centro da roda e o seu ponto de giro;
d	Distância entre P_o e P_c ;
$\dim(x)$	Dimensão de x ;
$\Delta_c(q)$	Distribuição espacial de q ;
$\Delta D(k)$	Deslocamento linear do robô;
$\Delta D_l(k)$	Deslocamento linear da roda esquerda;
$\Delta D_r(k)$	Deslocamento linear da roda direita;
$\Delta \theta(k)$	Deslocamento angular do robô;
$\Delta \theta_t(k)$	Deslocamento angular real;
e	Erro de posição;
e	Módulo do erro de posição em coordenadas polares;

e_f	Erro filtrado por $F(s)$;
e_1	Erro aumentado;
$E(\mathbf{x})$	Matriz composta por todas as $L_{g_i} L_f^{k-1} h_j(\mathbf{x})$;
$\bar{E}(\mathbf{x})$	Estimativa de $E(\mathbf{x})$;
$f(x)$	Função não linear de x ;
$F(s)$	Função de transferência de filtro;
φ	Deslocamento angular da roda;
ϕ	Ângulo do erro de posição em coordenadas polares;
Φ	Vetor de erros paramétricos;
$g(x)$	Função não linear de x ;
$G_m(s)$	Função de transferência do modelo de referência;
γ	Ângulo da direção de velocidade nula da roda sueca;
γ	Grau relativo de um processo;
γ_i	Coefficiente de ponderação > 0 ;
$\Gamma(e, \phi)$	Matriz de transformação para coordenadas polares;
h	Coefficiente de ponderação > 0 ;
$h(x)$	Função não linear de x ;
$\text{inv}(\Delta)$	Fechamento involutivo da distribuição Δ ;
I	Matriz identidade;
I	Momento de inércia gerado por m_c e m_w com relação ao eixo vertical;
I_w	Momento de inércia gerado pelo conjunto roda/motor,
$J(\Phi, e)$	Custo a minimizar;
l	Comprimento do segmento $\overline{P_0 P_w}$;
l_w	Braço de giro da roda do robô;

$L_f h_j(\mathbf{x})$	Derivada de Lie da saída $h_j(\mathbf{x})$ com relação à $f(\mathbf{x})$;
$\bar{E}_f^{\gamma_p} h_p(\mathbf{x})$	Estimativa de $L_f^{\gamma_p} h_p(\mathbf{x})$;
$L(s)$	Polinômio mônico de grau γ ;
λ	Coeficiente de Lagrange;
λ	Coeficiente de ponderação > 0 ;
λ	Distância focal;
m_c	Massa da plataforma sem considerar as rodas e os motores;
m_w	Massa de cada conjunto roda/motor;
M	Grau de não-holonomicidade;
M	Matriz de Inércias;
$M(s)$	Inverso de $L(s)$;
μ	Coeficiente de Lagrange;
N	Número de rodas;
$\mathcal{N}(\mathbf{x})$	Espaço nulo de \mathbf{x} ;
N_c	Número de rodas centradas;
N_f	Número de rodas convencionais fixas;
N_{oc}	Número de rodas não centradas;
N_{sw}	Número de rodas suecas;
$NP_l(k)$	Número de pulsos lidos do <i>encoder</i> esquerdo;
$NP_r(k)$	Número de pulsos lidos do <i>encoder</i> direito;
η	Variável de entrada do modelo cinemático;
P	Matriz de covariância;
P_0	Origem do sistema de coordenadas inercial;
P_c	Centro de gravidade do robô;

P_w	Projeção vertical do centro da roda do robô;
Π	Número de pulso por revolução dos <i>encoders</i> ;
r	Raio da roda;
r_l	Raio do <i>landmark</i> ;
${}^c\mathbf{R}_0$	Matriz de rotação entre os sistemas de coordenadas associados a P_c e P_0 ;
${}^w\mathbf{R}_c$	Matriz de rotação entre os sistemas de coordenadas associados a P_w e P_c ;
$\rho(x)$	<i>Rank</i> de x ;
q	Vetor de coordenadas de configuração;
$\text{sgn}(x)$	Função sinal de x ;
$\text{span}\{\text{col}(x)\}$	Espaço varrido pelas colunas de x ;
$S(q)$	Dinâmica do modelo cinemático de configuração;
δ_m	Grau de mobilidade;
δ_M	Grau de manobrabilidade;
δ_s	Grau de dirigibilidade;
$\delta\theta(k)$	Erro de orientação;
Θ	Orientação entre a câmera com relação ao <i>landmark</i> ;
Θ	Vetor de parâmetros desconhecidos;
$\bar{\Theta}$	Estimativa de Θ ;
θ	Orientação do robô;
$\theta_B(k)$	Orientação medida pela bússola;
$\theta_t(k)$	Orientação real do robô;
$\theta(k)$	Orientação medida por <i>dead-reckoning</i> ;
T	Energia cinética;
gT_c	Transformação homogênea entre o sistema inercial e o centro de massa;

vT_c	Transformação homogênea entre o centro de massa e a câmera;
vT_l	Matriz de transformação homogênea entre o <i>landmark</i> e câmera;
τ	Vetor de torque nas rodas;
τ_c	Torque aplicado para orientação das rodas centradas;
τ_ϕ	Torque aplicado para rotação da roda;
τ_{oc}	Torque aplicado para orientação da rodas não centrada;
u	Variável de entrada genérica de modelo no espaço de estados;
v	Vetor de entradas equivalentes;
V	Candidata a função de Lyapunov;
V	Vetor de forças centrífugas e de Coriolis;
$v(k)$	ruído de medida;
W	Matriz de regressão;
W_f	Matriz de regressão filtrada por $F(s)$
$W(k)$	Ruído de estado;
$\omega(k)$	Ruído de estado;
x	Variável de estado genérica;
x_c	Coordenada do centro de gravidade do robô na direção X;
x_r^c	Coordenada o ponto de referência do robô na direção X_c ;
x_r	Referência de posição na direção X;
X	Matriz de regressão filtrada pelo modelo de referência $G(s)$;
ξ	Vetor de coordenadas de postura;
ξ_0	Vetor de posição e orientação do robô descrito no sistema inercial;
ξ_c	Vetor de posição e orientação do robô descrito no sistema associado a P_c ;
ξ_r	Vetor de posição e orientação do ponto de referência do robô;

ξ_w	Vetor de posição e orientação do robô descrito no sistema associado a P_w ;
y	Vetor de saídas de um processo;
y_c	Coordenada do centro de gravidade do robô na direção Y;
y_r^c	Coordenada o ponto de referência do robô na direção y_c ;
y_i	i-ésimo componente do vetor y ;
y_i^γ	Derivada de ordem γ do i-ésimo componente de y ;
y_m	Saída do modelo de referência;
y_r	Referência de posição na direção Y;
ω_r	Velocidade da roda direita;
ω_l	Velocidade da roda esquerda;
ψ	Ângulo entre a direção de movimento da roda e a direção X_c ;
Z	Distância da câmera ao <i>landmark</i> ;
ζ	Derivada de β_c .

Sumário

O problema de estabilização de robôs móveis na posição e orientação desejadas é abordado neste trabalho. Inicialmente são apresentados os modelos cinemático e dinâmico de robôs móveis com rodas e mostra-se que são possíveis apenas cinco classes de modelos. A seguir, são propostas duas estratégias de controle. Uma baseada na linearização por realimentação do modelo do robô e outra baseada em uma transformação não contínua de coordenadas. A aplicabilidade destas estratégias de controle, para cada uma das cinco classes de robôs móveis, é discutida.

Mostra-se que a linearização por realimentação de estados permite o controle da posição e orientação de apenas uma das classes de robôs móveis. Utilizando-se esta estratégia, duas outras classes podem ter apenas a posição controlada. Para as demais classes esta estratégia é praticamente inútil. A estratégia utilizando transformação descontínua de coordenadas, por outro lado, permite o controle de robôs pertencentes a todas as classes

Como os parâmetros do modelo dinâmico do robô podem ser variantes no tempo, desenvolve-se versões adaptativas dos controladores propostos. Em particular, o controlador com *backstepping* adaptativo permite que a prova de estabilidade em malha fechada, isto é considerando o processo, o controle e a adaptação, seja feita em conjunto.

Para permitir a implementação dos controladores propostos, são apresentadas técnicas de estimação de posição e orientação através da fusão de dados obtidos a partir de *encoders* incrementais, visão computacional e bússola digital.

A validação dos métodos desenvolvidos é realizada através de simulações e da implementação de dois protótipos.

Abstract

The problem of mobile robot stabilization on a desired position and orientation is dealt with in this work. Initially, the kinematic and dynamic models of wheeled mobile robots are presented and it is shown that only five model classes are possible. Then, two control strategies are proposed. The first one is based on the feedback linearization of the robot model, while the other is based on a discontinuous coordinate transform. The applicability of both control strategies to each one of the five robot classes is discussed.

It is shown that only one robot class can have its position and orientation controlled by the state feedback linearization. Two other classes can have only the position controlled. For the remaining classes, this method is practically useless. On the other hand, the discontinuous transform strategy can control the position and orientation of robots belonging to all classes.

Since the parameters of the robot dynamic model can be time-varying, adaptive versions of the proposed controllers are developed. In particular, the adaptive backstepping controller can have its closed-loop stability proved by considering the process, the control and the adaptation as a whole.

In order to enable the implementation of the proposed controllers, position and orientation estimation methods using data fusion are presented. Data from incremental encoders, computer vision and a digital compass are used.

The validation of the proposed methods is done by simulation and by the implementation of two prototypes.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Elder M. Hemerly pela dedicação, apoio e incentivo, tanto no desenvolvimento deste trabalho quanto no curso de mestrado e principalmente, por seus esforços no sentido de que eu não fosse tão dispersivo.

Aos colegas Aguinaldo Cardozo Costa Filho, Hilton Abilio Gründling, Laurizete dos Santos Camargo, Luis Fernando Alves Pereira e Neusa Maria Franco de Oliveira, pela agradável convivência durante o curso.

À CAPES pela bolsa de estudos.

Capítulo I

Introdução

Embora robôs industriais estejam em uso há aproximadamente 20 anos, a área de robôs de serviço está recém emergindo como um campo de aplicação[63]. As aplicações de robôs de serviço incluem: entrega de correspondência, aspiração de pó, lavagem externa de aviões, assistência a pessoas com deficiências, aragem de terra para plantio etc. Pesquisas direcionadas aos elementos necessários para a implementação de robôs de serviço tem sido desenvolvidas [2]. Em [63] é sugerido que os robôs industriais evoluíram para os robôs de serviço, que por sua vez evoluirão para os robôs pessoais. Ter-se-ia, desta forma, com os robôs uma evolução similar à que ocorreu com os computadores. Parece razoável supor que a disseminação dos robôs pessoais terá um impacto maior do que a disseminação dos computadores pessoais.

Atualmente estão sendo desenvolvidos em conjunto pela NASA, MIT e CMU vários protótipos de robôs para exploração planetária. Vide [73] para um resumo. Estes robôs serão utilizados especificamente nos projetos *Mars Surveyor*, para explorar o planeta Marte e *Mission to Planet Earth*, para explorar locais de difícil acesso, tais como crateras de vulcões aqui na Terra.

Uma variante destes robôs exploradores que ultimamente tem recebido bastante atenção são os robôs submarinos [80][81], projetados para desenvolverem tarefas a profundidades impossíveis de serem atingidas por mergulhadores humanos.

Um requisito comum a todas estas classes de robôs é a capacidade de locomoção. Em geral, é necessário que esta locomoção seja feita de forma autônoma. A capacidade de

locomoção autônoma torna-se atrativa basicamente devido a dois fatores: 1) necessidade de interação com um ambiente pouco estruturado; 2) necessidade de operação sem supervisão.

A falta de estrutura do ambiente decorre do tipo de tarefa que um robô de serviço ou explorador deve executar. Um robô industrial interage basicamente com outras máquinas e portanto é possível construir-se um ambiente relativamente estruturado. Um robô de serviço vai interagir com pessoas, que não costumam manter um ambiente estruturado à sua volta. Um robô explorador vai operar em um ambiente inóspito e provavelmente desconhecido. Portanto, não é razoável supor que seja estruturado.

A necessidade de operação sem supervisão pode surgir devido à falta de um supervisor treinado nas proximidades do robô e/ou por dificuldades de comunicação, como nos casos de exploração espacial, onde o atraso de comunicação é significativo ou nos caso de operação submersa, onde a água dificulta a utilização de ondas eletromagnéticas para comunicação.

Outra aplicação possível seria em rodovias inteligentes, onde os veículos viajariam sem a necessidade de operação humana. Os projetos experimentais neste sentido [18][24] indicam que a operação automática dos veículos permitirá um aumento no fluxo de veículos com a conseqüente diminuição dos congestionamentos e da poluição gerada.

Atualmente, os principais problemas associados a implementação de robôs móveis dizem respeito à determinação de modelos cinemáticos e dinâmicos adequados, à estimação da posição e orientação do robô, ao controle do robô e ao planejamento de trajetórias a serem rastreadas. Aqui, pretende-se basicamente investigar técnicas para o controle de robôs móveis com rodas. Em especial, deseja-se obter métodos de controle que possam ser implementados em tempo real. No entanto, o problema de controle está intimamente relacionado com a modelagem. Conseqüentemente, a obtenção de modelos adequados para os propósitos de controle também será abordada. Além disso, como será mostrado nos capítulos seguintes, os métodos de modelagem e controle propostos necessitam de informação precisa sobre a

posição e a orientação do robô. Quando se deseja obter resultados de simulação, isto não representa problema, pois todas as variáveis do sistema estão disponíveis. Por outro lado, em um robô real, nem sempre todas as variáveis, em especial as de posição e orientação, estarão diretamente disponíveis. Neste caso, se estas variáveis forem necessárias para a implementação do controle, elas terão que ser estimadas.

Várias técnicas estão disponíveis para a obtenção da posição e da orientação do robô, utilizando-se *encoders*, imagens de vídeo [74][76] e de sinais de sonar [11][32][49].

É conhecido [25] que o método de *dead-reckoning* [10] não é um método adequado para estimação da posição de robôs móveis, visto que há erros sistemáticos devido à assimetria nas rodas, escorregamentos, etc. Os sensores baseados em processamento de imagens, por outro lado, não sofrem estes problemas e podem fornecer uma boa estimativa para a posição. No entanto, o tempo requerido para o processamento da imagem é grande, o que dificulta a sua implementação em tempo real.

Dados provenientes de sonares de ultra-som podem ser processados em um tempo menor, mas a precisão obtida, especialmente da orientação do robô é menor do que com sensores de visão. Os problemas encontrados para se obter uma estimativa da posição e orientação do robô através de medidas de distância por ultra-som devem-se basicamente aos efeitos de reflexão, difração e ângulo de abertura a que estão sujeitas as ondas sonoras [39]. Um método para estimação de posição e orientação de robôs móveis é apresentado em [26]. As dificuldades inerentes ao uso de sensores de ultra-som podem ser evitadas utilizando-se medidas de distância obtidas a partir de um *scanner* a laser, como em [71]. Basicamente, os mesmos métodos utilizados com sonares podem ser utilizados com *scanner* a laser, com a vantagem dos efeitos de reflexão, difração e ângulo de abertura do feixe serem minimizados. No entanto, o custo de um *scanner* a laser é bem superior ao de um sistema de sonares.

As dificuldades apresentadas por cada um destes sensores podem ser contornadas através de um método de estimação da posição que considere os dados provenientes dos vários sensores, como em [26].

Um robô móvel é um sistema não-holonômico [38], isto é, existem restrições não integráveis. Embora a dinâmica destes sistemas tenham sido bastantes estudadas, apenas recentemente começaram a aparecer estudos sobre o controle destes sistemas [8]. As dificuldades decorrentes das restrições não-holonômicas a que está sujeito o sistema tem sido contornadas através do uso de controladores com acoplamento cruzado [17], controladores com lógica nebulosa [37], controladores baseados em comportamento [41] e redes neurais [54].

As dificuldades para o controle de postura (posição e orientação) de robôs móveis devem-se basicamente às limitações expostas por Brockett [12] onde é mostrado que um sistema não-holonômico sem deriva não pode ser estabilizado em um ponto através de uma realimentação suave invariante no tempo. Deste resultado originam-se os principais métodos para controle de postura de robôs móveis.

Alguns métodos [70][72] abandonam a idéia de estabilização em um ponto e procuram obter convergência para uma trajetória. Neste caso, o problema de movimentação de um ponto a outro de seu espaço de trabalho é resolvido em duas etapas: primeiro uma trajetória que leva de um ponto a outro é computada *off-line* e em seguida a lei de controle é utilizada para fazer com que o robô se movimente seguindo a trajetória planejada. Em [70] é mostrado que com esta técnica pode-se, inclusive, obter uma lei de controle linearizante. Um método para planejamento de trajetórias para sistemas não-holonômicos é apresentado em [45]. Em [38] é apresentada uma abordagem bastante ampla dos métodos existentes para planejamento de trajetórias para robôs, sujeitos ou não a restrições não-holonômicas. Uma vantagem deste tipo de abordagem é que facilita a consideração de ambientes contendo obstáculos.

Outros métodos mantêm a exigência de convergência para um ponto às custas de leis de controle mais complexas. Em [8] obtém-se estabilização empregando uma realimentação não suave, porém contínua por partes. Uma outra estratégia com lei de controle contínua por partes é apresentada em [15], onde a trajetória resultante é formada por arcos de circunferência. Pode ser provado [58] que a trajetória mais curta entre dois pontos no plano, dadas as orientações iniciais e finais do robô é formado por arcos de circunferência e segmentos de reta.

Utilizando-se uma lei de controle suave, é possível obter-se convergência para um ponto se esta lei de controle for variante no tempo, como em [43][45]. Embora em [43] seja obtida convergência exponencial, leis de controle variantes no tempo, em geral, produzem trajetórias demasiadamente oscilantes.

Também são utilizadas leis de controle híbridas [51]. Estas leis diferenciam-se das demais por combinarem leis de controle de tempo contínuo, como as discutidas nos parágrafos anteriores e leis de controle de tempo discreto. Na maioria das vezes, ocorre chaveamento entre uma lei de controle não suave e uma lei de controle variante no tempo. Leis de controle híbridas também tendem a produzir trajetórias bastante oscilantes.

A técnica de linearização por realimentação de estados [28] mostra-se útil para controle de robôs móveis. Em [77] é utilizado um controlador linear clássico para controlar o sistema linearizado. Um controlador tipo modelo de referência é empregado em [36]. Esta técnica de controle é bastante interessante do ponto de vista de estimação da posição e orientação do robô móvel. Tendo-se um modelo linear, pode-se utilizar o filtro de Kalman convencional para obtenção das estimativas de posição e orientação. No entanto, com esta estratégia pode-se obter linearização completa apenas para robôs omnidirecionais (podem se mover em qualquer direção sem reorientação) . No caso geral, não é possível linearizar todas as

variáveis de posição e orientação. Por outro lado, existem robôs como o RAMSIS [26] onde a orientação do corpo do robô pode ser controlada por outros meios.

Todos estes trabalhos têm em comum o fato de que a descrição do veículo é feita em coordenadas cartesianas. Porém, como sugerido em [42][43], o uso de transformações de coordenadas não suaves seguido de uma realimentação suave é um procedimento promissor, uma vez que possibilita o desenvolvimento de leis de controle suaves, invariantes no tempo, que estabilizem o sistema e não estejam sujeitas às limitações impostas por Brockett. A idéia nesta abordagem é fazer uma transformação de coordenadas de forma que o sistema transformado seja descontínuo [3].

A abordagem utilizada neste trabalho é semelhante a [1]. Uma transformação de coordenadas faz com que o modelo do sistema torne-se descontínuo na origem. Obtém-se, através de uma análise *Lyapunov-like*, uma lei de controle suave invariante no tempo. Quando sujeito a esta realimentação, o sistema torna-se contínuo também na origem.

As principais contribuições deste trabalho são:

- 1) Desenvolvimento de um procedimento de linearização por realimentação estática de estados para diversas configurações de robôs móveis com rodas;
- 2) Extensão da lei de controle suave, baseadas em transformação para coordenadas polares, proposta por [1], para todas as configurações possíveis de robôs móveis com rodas;
- 3) Proposição de versões adaptativas das leis de controle consideradas nos dois itens anteriores;
- 4) Proposição e implementação de estratégias para estimação de posição e orientação de robôs móveis;

- 5) Implementação de dois protótipos de robô móvel, viabilizando a aplicação em tempo real e avaliação de desempenho das estratégias de estimação e controle propostas.

Este trabalho está organizado da seguinte forma: no capítulo II são apresentada a modelagem cinemática e dinâmica de robôs móveis com rodas. Mostra-se que são possíveis apenas cinco classes de modelos. Os modelos obtidos são utilizados nos capítulos seguintes onde são apresentadas estratégias de controle.

O capítulo III apresenta uma estratégia de controle baseada em linearização do modelo dinâmico do robô semelhante a proposta em [77]. Procura-se generalizar esta técnica para todas as classes de robôs móveis e as suas limitações são analisadas.

No capítulo IV são propostas duas técnicas de controle utilizando-se uma transformação descontínua e *backstepping* [19]. A idéia do controlador de *backstepping* é abordar de forma separada o modelo cinemático e o modelo dinâmico do robô. Com base em [1], projeta-se um controlador para a cinemática do robô utilizando-se uma transformação de variáveis descontínua e um método Lyapunov-like. Como a técnica proposta em [1] aplica-se apenas a robôs de uma das classes, propõe-se transformações adicionais que permitem o controle de robôs pertencentes a qualquer classes. São propostos dois tipos de controlador de *backstepping*: um utilizando um controlador PID e outro utilizando um controlador linearizante, semelhante ao discutido no capítulo III.

Considerando-se que os parâmetros de massa e inércia do robô são de difícil determinação e que podem variar durante a operação, no capítulo V são propostas versões adaptativas das estratégias de controle discutidas nos capítulos II e III. Adota-se uma estratégia de adaptação baseada no método dos mínimos quadrados, proposta por [52]. Estende-se os resultados apresentados em [52] para sistemas tipo MIMO com grau relativo 1.

Métodos para estimação da posição e orientação de robôs móveis são discutidos no capítulo VI.

A implementação dos protótipos de dois robôs móveis, utilizados para obtenção de dados experimentais nos capítulos anteriores, é detalhada no capítulo VII.

Finalmente, no capítulo VIII são apresentadas conclusões e sugestões para trabalhos futuros.

Capítulo II

Modelagem Cinemática e Dinâmica

Existem basicamente três métodos para acionamento de robôs móveis: a) através de rodas, b) através de lagartas e c) através de pernas. O acionamento através de rodas é o mais utilizado, por ser de fácil construção e adaptar-se relativamente bem às condições de operação em ambientes *in-door* e ambientes *out-door* pavimentados, onde as irregularidades do solo não são muito severas. A utilização do acionamento através de lagartas ou através de pernas restringe-se à operação em ambientes *out-door* hostis, devido à sua maior facilidade em transpor obstáculos.

A utilização limitada do acionamento através de pernas deve-se à sua construção mecânica mais complexa em relação aos outros métodos de acionamento. Além disso, geralmente necessita um maior número de atuadores e o projeto do sistema de controle torna-se mais complicado, pois é necessário se considerar o equilíbrio do robô. Um exemplo de robô acionado por pernas é o Ambler [64] desenvolvido pela NASA.

O acionamento através de lagartas possui uma construção relativamente simples. No entanto, a operação normal deste tipo de acionamento baseia-se intencionalmente no deslizamento das lagartas. Isto faz com que métodos de determinação de postura, como o *dead-reckoning* (vide capítulo VI) tenham um desempenho muito ruim. Em [10] é apresentado um método para melhorar o desempenho do *dead-reckoning* deste tipo de robô. Porém, este método apresenta limitações para aplicação em ambientes inóspitos, onde um acionamento através de lagartas seria desejável.

Neste trabalho, seguindo a formulação de [14] serão considerados apenas robôs com acionamento através de rodas. Os modelos para robôs acionados por rodas podem ser

divididos em quatro tipos: a) modelo cinemático de postura, b) modelo cinemático de configuração, c) modelo dinâmico de configuração e d) modelo dinâmico de postura.

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

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

A maioria dos trabalhos apresentados na literatura [8][31][44] descreve o robô em coordenadas cartesianas e utiliza apenas um modelo cinemático de postura. Um número reduzido de autores consideram também um modelo dinâmico [77]. Em [44] é utilizado um modelo cinemático de postura em coordenadas polares. A modelagem em coordenadas polares permite escrever um modelo com propriedades interessantes do ponto de vista de controle. Estas propriedades serão exploradas no capítulo IV.

A seguir serão apresentados os modelos utilizados neste trabalho, seguindo a formulação semelhante a [14]. Embora apenas os modelos de postura sejam necessários para controle do robô, são apresentados também os modelos de configuração, pois o modelo dinâmico de postura é obtido a partir do modelo dinâmico de configuração.

II.1. Modelo Cinemático de Postura

Neste trabalho assume-se um robô móvel constituído de um corpo rígido cujas rodas não sofrem deformação e que movimenta-se no plano horizontal. A posição e a orientação do robô são descritas em relação a um sistema inercial $\{X_0, Y_0\}$. Um sistema de coordenadas $\{X_c, Y_c\}$ é associado ao corpo do robô, conforme detalhado na Fig. II-1.

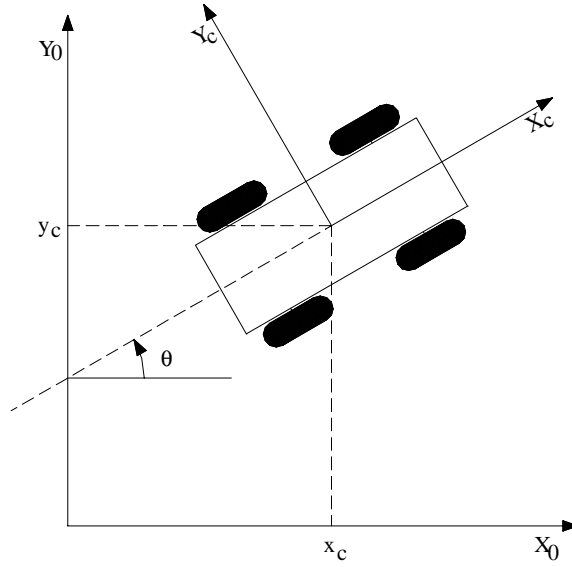


Fig. II-1 - Definição dos sistemas de coordenadas.

Com isto, as coordenadas de postura do robô descritas no sistema $\{X_0, Y_0\}$ serão

$$\xi_0 = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} \quad (\text{II.1})$$

e

$${}^c\mathbf{R}_0 = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.2})$$

será a matriz de rotação entre $\{X_c, Y_c\}$ e $\{X_0, Y_0\}$.

II.1.1. Descrição das Rodas

O modelo desenvolvido para as rodas assume que o plano de cada roda permaneça vertical durante o movimento e que a rotação se dá em torno de um eixo horizontal cuja orientação em relação os sistema de coordenadas $\{X_c, Y_c\}$ pode ser fixa ou variável.

Considera-se duas classes de rodas: rodas convencionais e rodas suecas. Em ambos os casos, assume-se que o ponto de contato com o solo é reduzido a um único ponto.

As rodas convencionais satisfazem à condição de rolamento sem deslizamento, portanto a velocidade do ponto de contato com o solo é zero. Ou seja, tanto as componentes de velocidade paralela e ortogonal ao plano da roda são nulas.

Para uma roda sueca apenas um dos componentes de velocidade do ponto de contato é zero. A direção desta componente é arbitrária, mas fixa em relação à orientação da roda.

Considerando-se as suposições acima, pode-se deduzir as restrições às quais está sujeito o movimento das rodas:

II.1.1.1. Rodas Convencionais Fixas

Da Fig. II-1 e das expressões (II.1) e (II.2), tem-se que a velocidade do robô em relação ao sistema de coordenadas inercial, descrita no sistema de coordenadas $\{X_c, Y_c\}$ será

$$\dot{\xi}_c = {}^c\mathbf{R}_0 \dot{\xi}_0 \quad (\text{II.3})$$

Esta mesma velocidade, descrita em um sistema de coordenadas $\{X_w, Y_w\}$ associando ao centro da roda, conforme a Fig. II-2 tem a forma

$$\dot{\xi}_w = {}^w\mathbf{R}_c \dot{\xi}_c + \mathbf{I}_w \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^w\mathbf{R}_c \dot{\xi}_c \quad (\text{II.4})$$

onde \mathbf{I}_w é o braço de giro e ${}^w\mathbf{R}_c$ é a matriz de rotação entre os sistemas de coordenadas $\{X_w, Y_w\}$ e $\{X_c, Y_c\}$ dados por

$$\mathbf{I}_w = \begin{bmatrix} -1 \sin \beta \\ -1 \cos \beta \\ 0 \end{bmatrix} \quad (\text{II.5})$$

$${}^w\mathbf{R}_c = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.6})$$

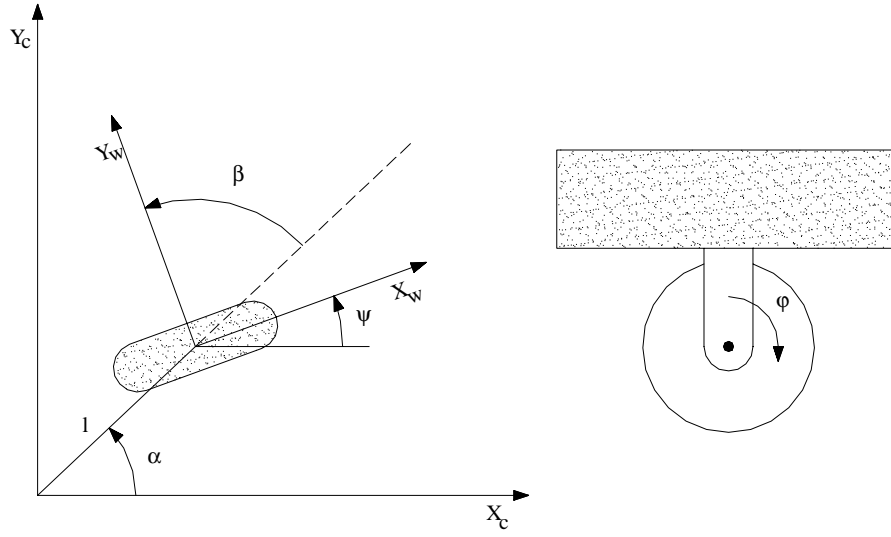


Fig. II-2 - Definição do sistema de coordenadas da roda convencional fixa.

como $\psi = \alpha + \beta - \pi/2$ pode-se escrever ${}^w\mathbf{R}_c$ na forma

$${}^w\mathbf{R}_c = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.7})$$

e utilizando-se (II.3), (II.4) pode ser reescrita como

$$\xi_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & 0 \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^c\mathbf{R}_0 \xi_0 + \begin{bmatrix} -1 \sin \beta \\ -1 \sin \beta \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^w\mathbf{R}_0 \xi_0 \quad (\text{II.8})$$

por outro lado, tem-se que o segundo termo de (II.8) é

$$\begin{bmatrix} -1 \sin \beta \\ -1 \sin \beta \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^w\mathbf{R}_0 \xi_0 = \begin{bmatrix} -1 \sin \beta \\ -1 \sin \beta \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \cos \beta \\ 0 & 0 & 1 \sin \beta \\ 0 & 0 & 0 \end{bmatrix} {}^c\mathbf{R}_0 \xi_0 \quad (\text{II.9})$$

logo tem-se

$$\xi_w = \begin{bmatrix} \sin(\alpha + \beta) & -\cos(\alpha + \beta) & -1 \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 1 \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^c\mathbf{R}_0 \xi_0 \quad (\text{II.10})$$

e como $\mathcal{E}_w = \begin{bmatrix} r\phi \\ 0 \\ \theta \end{bmatrix}$ chega-se a expressão

$$\begin{bmatrix} -\sin(\alpha + \beta) & \cos(\alpha + \beta) & 1 \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & 1 \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^c\mathbf{R}_0 \mathcal{E}_w + \begin{bmatrix} r\phi \\ 0 \\ -\theta \end{bmatrix} = 0 \quad (\text{II.11})$$

que representa as restrições a que está sujeito o movimento do robô devido à roda em questão.

Esta expressão pode ainda ser desmembrada em

$$\begin{bmatrix} -\sin(\alpha + \beta) & \cos(\alpha + \beta) & 1 \cos \beta \end{bmatrix} {}^c\mathbf{R}_0 \mathcal{E}_w + r\phi = 0 \quad (\text{II.12})$$

que representa a restrição de movimento ao longo do plano da roda e

$$\begin{bmatrix} \cos(\alpha + \beta) & \sin(\alpha + \beta) & 1 \sin \beta \end{bmatrix} {}^c\mathbf{R}_0 \mathcal{E}_w = 0 \quad (\text{II.13})$$

que representa a restrição de movimento ortogonal ao plano da roda.

II.1.1.2. Rodas Convencionais Orientáveis Centradas

Uma roda convencional orientável centrada é uma roda cujo plano pode ser rotacionado em torno de um eixo vertical passando pelo centro da roda. A descrição deste tipo de roda é a mesma utilizada para rodas fixas (caso anterior), porém o ângulo β não é constante, mas variável.

II.1.1.3. Rodas Convencionais Orientáveis Não-centradas

Este tipo de roda também pode ter sua orientação alterada. No entanto, neste caso a rotação do plano da roda ocorre em torno de um eixo vertical que **não** passa através do centro da roda, conforme a Fig. II-3. Neste caso, a expressão (II.5) torna-se

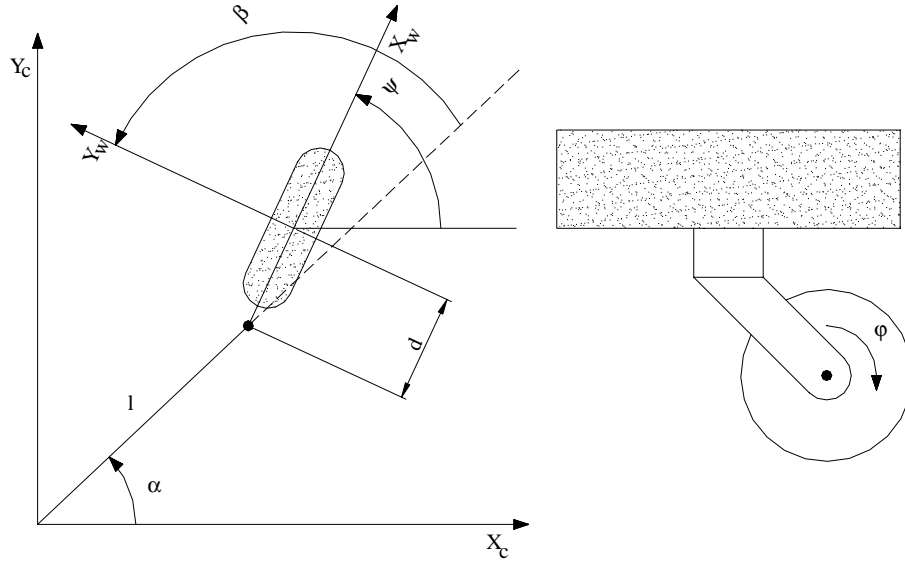


Fig. II-3 - Roda convencional orientável não-centrada.

$$\mathbf{I}_w = \begin{bmatrix} -d - l \sin \beta \\ -l \cos \beta \\ 0 \end{bmatrix} \quad (\text{II.14})$$

e portanto as expressões (II.11)-(II.13) tornam-se

$$\begin{bmatrix} -\sin(\alpha + \beta) & \cos(\alpha + \beta) & l \cos \beta \\ \cos(\alpha + \beta) & \sin(\alpha + \beta) & d + l \sin \beta \\ 0 & 0 & 1 \end{bmatrix} {}^c \mathbf{R}_0 \dot{\xi}_0 + \begin{bmatrix} r \dot{\phi} \\ 0 \\ -\dot{\phi} \end{bmatrix} = 0 \quad (\text{II.15})$$

$$[-\sin(\alpha + \beta) \quad \cos(\alpha + \beta) \quad l \cos \beta] {}^c \mathbf{R}_0 \dot{\xi}_0 + r \dot{\phi} = 0 \quad (\text{II.16})$$

$$[\cos(\alpha + \beta) \quad \sin(\alpha + \beta) \quad d + l \sin \beta] {}^c \mathbf{R}_0 \dot{\xi}_0 = 0 \quad (\text{II.17})$$

II.1.1.4. Rodas Suecas

A posição deste tipo de roda em relação ao sistema de coordenadas $\{X_c, Y_c\}$ é descrita de forma semelhante à roda convencional fixa. Porém, é necessário a introdução de mais um parâmetro para caracterizar a direção, com relação ao plano da roda, do componente nulo da velocidade do ponto de contato. A figura II-4 mostra a definição dos sistemas de coordenadas

utilizados para descrever este tipo de roda. Note-se que o eixo X_w está alinhado com a direção do componente nulo da velocidade do ponto de contato e não com a direção do plano da roda como nos casos anteriores.

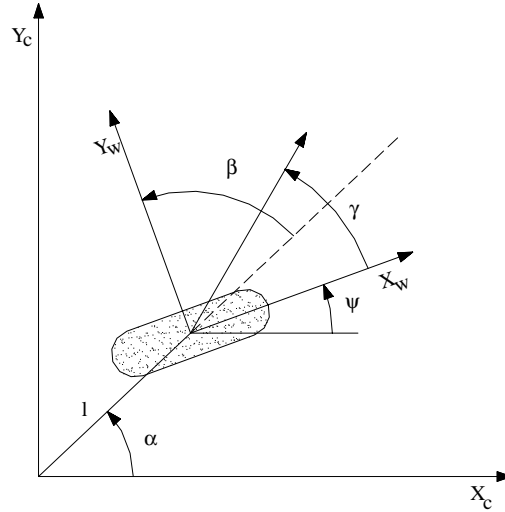


Fig. II-4 - Definição do sistema de coordenadas para roda sueca.

Com esta definição de sistemas de coordenadas, o braço de giro da roda passará a ser dado por

$$\mathbf{I}_w = \begin{bmatrix} -1 \sin(\beta + \gamma) \\ -1 \cos(\beta + \gamma) \\ 0 \end{bmatrix} \quad (\text{II.18})$$

e como agora tem-se $\psi = \alpha + \beta + \gamma - \pi/2$ a matriz de rotação entre os sistemas de coordenadas $\{X_w, Y_w\}$ e $\{X_c, Y_c\}$ será

$${}^w\mathbf{R}_c = \begin{bmatrix} \sin(\alpha + \beta + \gamma) & -\cos(\alpha + \beta + \gamma) & 0 \\ \cos(\alpha + \beta + \gamma) & \sin(\alpha + \beta + \gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{II.19})$$

Por outro lado, tem-se que a velocidade do robô descrita no sistema de coordenadas $\{X_w, Y_w\}$ é dada por

$$\xi_w = \begin{bmatrix} r\dot{\phi}\cos\gamma \\ \xi_{wy} \\ \dot{\phi} \end{bmatrix} \quad (\text{II.20})$$

Como se trata de uma roda sueca, o componente de velocidade normal é desconhecido, pois a velocidade do ponto de contato nesta direção não é nula. Com isto, a restrição de movimento da roda pode ser escrita como

$$\begin{bmatrix} -\sin(\alpha + \beta + \gamma) & \cos(\alpha + \beta + \gamma) & 1 \cos(\beta + \gamma) \end{bmatrix} {}^c\mathbf{R}_0 \xi_w + r \cos \gamma \dot{\phi} = 0 \quad (\text{II.21})$$

Assume-se que para uma roda sueca $\gamma \neq \frac{\pi}{2}$. Caso $\gamma = \frac{\pi}{2}$ a roda estaria sujeita a uma restrição idêntica à de uma roda convencional, perdendo o benefício de ser uma roda sueca.

II.1.2. Restrições à Mobilidade do Robô

Considere-se um robô que possua N rodas, sendo N_f rodas convencionais fixas, N_c rodas centradas, N_{oc} rodas não centradas e N_{sw} rodas suecas. Então, as equações que descrevem as restrições (II.12, II.13, II.16, II.17 e II.21) podem ser escritas na forma matricial:

$$\mathbf{J}_1(\beta_c, \beta_{oc}) {}^c\mathbf{R}_0 \xi_w + \mathbf{J}_2 \dot{\phi} = 0 \quad (\text{II.22})$$

$$\mathbf{C}_1(\beta_c, \beta_{oc}) {}^c\mathbf{R}_0 \xi_w + \mathbf{C}_2 \dot{\phi} = 0 \quad (\text{II.23})$$

com as seguintes definições:

$$\mathbf{J}_1(\beta_c, \beta_{oc}) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{J}_{1f} \\ \mathbf{J}_{1c}(\beta_c) \\ \mathbf{J}_{1oc}(\beta_{oc}) \\ \mathbf{J}_{1sw} \end{bmatrix} \quad (\text{II.24})$$

$$\mathbf{C}_1(\beta_c, \beta_{oc}) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{C}_{1f} \\ \mathbf{C}_{1c}(\beta_c) \\ \mathbf{C}_{1oc}(\beta_{oc}) \end{bmatrix} \quad (\text{II.25})$$

$$\mathbf{C}_2 \stackrel{\Delta}{=} \begin{bmatrix} 0 \\ 0 \\ \mathbf{C}_{2oc} \end{bmatrix} \quad (\text{II.26})$$

onde \mathbf{J}_{1f} , \mathbf{J}_{1c} , \mathbf{J}_{1oc} e \mathbf{J}_{1sw} são, respectivamente, matrizes de dimensões $(N_f \times 3)$, $(N_c \times 3)$, $(N_{oc} \times 3)$ e $(N_{sw} \times 3)$. \mathbf{J}_2 é uma matriz cuja diagonal são os raios das rodas ou, no caso de rodas suecas, o raios das rodas multiplicados por $\cos\gamma$. \mathbf{C}_{1f} , \mathbf{C}_{1c} e \mathbf{C}_{1oc} são matrizes de dimensões $(N_f \times 3)$, $(N_c \times 3)$, $(N_{oc} \times 3)$. \mathbf{C}_{2oc} é uma matriz cuja diagonal são iguais as distâncias d das N_{oc} rodas não centradas.

Considerando-se apenas as primeiras $(N_f + N_c)$ restrições de (II.23) tem-se

$$\mathbf{C}_1^*(\beta_c) {}^c\mathbf{R}_0 \mathbf{e} = 0 \quad (\text{II.27})$$

sendo

$$\mathbf{C}_1^*(\beta_c) \triangleq \begin{bmatrix} \mathbf{C}_{1f} \\ \mathbf{C}_{1c}(\beta_c) \end{bmatrix} \quad (\text{II.28})$$

e portanto ${}^c\mathbf{R}_0 \mathbf{e}$ pertence ao espaço nulo de $\mathbf{C}_1^*(\beta_c)$.

As limitações da mobilidade do robô estão relacionadas com o *rank* de $\mathbf{C}_1^*(\beta_c)$.

Definição II.1. Grau de Mobilidade

Define-se **grau de mobilidade** de um robô móvel como

$$\delta_m = \dim \mathcal{N}(\mathbf{C}_1^*(\beta_c)) \quad (\text{II.29})$$

■

É evidente que $\rho(\mathbf{C}_1^*(\beta_c)) \leq 3$. Se $\rho(\mathbf{C}_1^*(\beta_c)) = 3$, então ${}^c\mathbf{R}_0 \mathbf{e} = 0$ e a movimentação do robô é impossível, portanto deve-se ter $\rho(\mathbf{C}_1^*(\beta_c)) \leq 2$. Por outro lado, $\rho(\mathbf{C}_{1f}) = 2$ implica a existência de pelo menos 2 rodas fixas cujos planos não são paralelos. Se existirem mais de duas rodas fixas, os eixos de todas elas deverão concorrer para o mesmo centro de rotação instantâneo. Como este centro de rotação é fixo, devido aos planos das rodas não serem paralelos, o único movimento possível para o robô é a rotação em torno deste ponto. Logo, para eliminar esta limitação, impõe-se a condição de que $\rho(\mathbf{C}_{1f}) \leq 1$, ou seja, se o robô possui mais de uma roda convencional fixa, todas elas devem estar em um mesmo eixo. Tem-se

ainda que $\rho(C_1^*(\beta_c)) \leq \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$, mas a situação em que $\rho(C_1^*(\beta_c)) < \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$ corresponde a ter-se os centros das rodas orientáveis centradas sobre o eixo comum das rodas fixas. Esta situação faz com que as rodas orientáveis percam a sua capacidade de atuar sobre a alocação do centro de rotação instantâneo, portanto assume-se também que $\rho(C_1^*(\beta_c)) = \rho(C_{1f}) + \rho(C_{1c}(\beta_c))$.

Definição II.2. Grau de Dirigibilidade

O **grau de dirigibilidade** de um robô móvel é definido como o número de rodas convencionais orientáveis centradas que podem ser orientadas independentemente para dirigir o robô, e é dado por

$$\delta_s = \rho(C_{1c}(\beta_c)) \quad (\text{II.30})$$

■

Se o robô estiver equipado com mais de δ_s rodas convencionais orientadas centradas, o movimento de $N_c - \delta_s$ rodas deverá ser coordenado com as demais, de forma a garantir a existência do centro de rotação instantâneo.

Do exposto acima, conclui-se que a configuração das rodas de um robô móvel deve ser tal que as seguintes restrições sejam satisfeitas:

$$1 \leq \delta_m \leq 3 \quad (\text{II.31})$$

$$0 \leq \delta_s \leq 2 \quad (\text{II.32})$$

$$2 \leq \delta_m + \delta_s \leq 3 \quad (\text{II.33})$$

O limite inferior de (II.31) significa que serão considerados apenas os casos em que é possível o movimento. A restrição (II.32) indica que podem existir no máximo duas rodas orientáveis centradas independentes. As configurações em que $\delta_m + \delta_s = 1$ permitem apenas a

rotação em torno de um centro instantâneo fixo, sendo portanto desconsideradas. O limite superior de (II.33) é devido a condição de que $\rho(C_1^*(\beta_c)) \leq 2$, e portanto $\delta_s=2$ implica $\delta_m=1$.

Pode ser verificado que apenas cinco classes de robôs com rodas, caracterizados pelo par (δ_m, δ_s) satisfazem as restrições (II.31-II.33): as classes (3,0), (2,0), (2,1), (1,1) e (1,2).

Classe (3,0)

Nesta classe de robô tem-se $\rho(C_1^*(\beta_c)) = 0$, portanto não existem rodas fixas ou rodas orientáveis centradas. Estes robôs são denominados **omnidirecionais**, pois podem mover-se instantaneamente em qualquer direção e com qualquer reorientação.

Classe (2,0)

Como $\rho(C_1^*(\beta_c)) = \rho(C_{1f}) + \rho(C_{1c}(\beta_c)) = 1$ e $\rho(C_{1c}(\beta_c)) = 0$, esta classe de robô tem uma roda fixa ou várias rodas fixas em um eixo comum, pois é necessário que $\rho(C_{1f}) = 1$. Os robôs desta classe normalmente possuem rodas operando em modo diferencial.

Classe (2,1)

Esta classe de robô possui pelo menos uma roda orientável centrada e não possui rodas fixas, pois $\rho(C_{1f}) = 0$. Existindo mais de uma roda orientável centrada, o movimento destas rodas deverá ser coordenado de forma que $\rho(C_{1c}(\beta_c)) = 1$.

Classe (1,1)

Neste caso tem-se $\rho(C_{1f}) = 1$ e portanto deve-se ter uma roda fixa, ou várias rodas fixas em um eixo comum e uma ou mais rodas orientáveis centradas, cujo centro não deve estar sobre o eixo das rodas fixas. Havendo várias rodas orientáveis centradas, o seu movimento deve ser coordenado de forma que $\rho(C_{1c}(\beta_c)) = 1$.

Robôs construídos na forma de carros convencionais, com direção tipo Ackerman [43] pertencem a esta classe.

Em [29] é apresentado um método para representar qualquer robô desta classe por um modelo de bicicleta, com apenas uma roda fixa e uma roda orientável centrada.

Classe (1,2)

Robôs desta classe não possuem rodas fixas, pois $\rho(\mathbf{C}_{1f}) = 0$. Por outro lado, possuem duas rodas orientáveis centradas, ou mais, desde que seu movimento seja coordenado de forma que $\rho(\mathbf{C}_{1c}(\beta_c)) = 2$.

II.1.3. Modelo Cinemático de Postura no Espaço de Estados

De (II.27) tem-se que ${}^c\mathbf{R}_0\mathbf{\xi}$ pertence ao espaço nulo de $\mathbf{C}_1^*(\beta_c)$. Portanto, lembrando que ${}^c\mathbf{R}_0^{-1} = {}^c\mathbf{R}_0^T = {}^0\mathbf{R}_c$, pode-se escrever

$$\mathbf{\xi}^0 = {}^0\mathbf{R}_c \Sigma(\beta_c) \eta \quad (\text{II.34})$$

onde as colunas de $\Sigma(\beta_c)$ formam uma base do espaço nulo de $\mathbf{C}_1^*(\beta_c)$. Pode-se verificar facilmente que as dimensões de $\Sigma(\beta_c)$ e η serão sempre $3 \times \delta_m$ e $\delta_m \times 1$.

Definindo-se $\zeta = \beta_c$, a expressão (II.34) pode ser aumentada para

$$\begin{cases} \mathbf{\xi}^0 = {}^0\mathbf{R}_c \Sigma(\beta_c) \eta \\ \beta_c = \zeta \end{cases} \quad (\text{II.35})$$

que é o modelo no espaço de estados do sistema, com as coordenadas de postura ξ e as coordenadas angulares β_c como variáveis de estado. Como entradas lineares do sistema tem-se η e ζ . Se por outro lado, o robô não possui rodas orientáveis centradas ($N_c=0$), a matriz Σ não dependerá de β_c e a expressão (II.34) não precisará ser aumentada. Na Tab. II-I são mostrados os modelos cinemáticos de postura particularizados para cada uma das cinco classes de robôs móveis.

Tab. II-I - Modelos cinemáticos de postura.

Classe	Modelo
(3,0) omnidirecional	$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix}$
(2,0) diferencial	$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix}$
(2,1)	$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\beta}_c \end{bmatrix} = \begin{bmatrix} \cos(\theta + \beta_c) & 0 & 0 \\ \sin(\theta + \beta_c) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \zeta \end{bmatrix}$
(1,1) carro	$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\beta}_c \end{bmatrix} = \begin{bmatrix} d \cos \theta \sin \beta_c & 0 \\ d \sin \theta \cos \beta_c & 0 \\ \cos \beta_c & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \zeta \end{bmatrix}$
(1,2)	$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\beta}_{c1} \\ \dot{\beta}_{c2} \end{bmatrix} = \begin{bmatrix} L \left[\sin \beta_{c1} \cos(\theta + \beta_{c2}) + \sin \beta_{c2} \cos(\theta + \beta_{c1}) \right] & 0 & 0 \\ L \left[\sin \beta_{c1} \sin(\theta + \beta_{c2}) + \sin \beta_{c2} \sin(\theta + \beta_{c1}) \right] & 0 & 0 \\ \sin(\beta_{c2} - \beta_{c1}) & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \zeta_1 \\ \zeta_2 \end{bmatrix}$

Genericamente, o modelo cinemático de postura pode ser escrito na forma

$$\dot{x} = B(x)u \quad (\text{II.36})$$

$$\text{com } x = \begin{cases} \xi, N_c = 0 \\ \begin{bmatrix} \xi \\ \beta_c \end{bmatrix}, N_c \geq 0 \end{cases}$$

$$B(x) \overset{\Delta}{=} \begin{cases} {}^0R_c \Sigma, N_c = 0 \\ \begin{bmatrix} {}^0R_c \Sigma(\beta_c) & 0 \\ 0 & I \end{bmatrix}, N_c \geq 0 \end{cases}$$

$$u \overset{\Delta}{=} \begin{cases} \eta, N_c = 0 \\ \begin{bmatrix} \eta \\ \zeta \end{bmatrix}, N_c \geq 0 \end{cases}$$

Definição II.3. Grau de Manobrabilidade

O **grau de manobrabilidade** de um robô móvel é definido como

$$\delta_M = \delta_m + \delta_s \quad (II.37)$$

■

Pode-se observar que a dimensão do vetor de entradas é igual ao grau de manobrabilidade do robô. Este parâmetro indica quantos graus de liberdade podem ser manipulados através das entradas η e ζ . Em outras palavras, indica a possibilidade de alocar-se livremente o centro instantâneo de rotação. $\delta_M=3$ indica que o centro instantâneo de rotação pode ser livremente alocado no plano, diretamente através de η para robôs da classe(3,0) ou através da reorientação das rodas centradas para robôs das classes (1,2) e (2,1). Por outro lado, $\delta_M=2$ indica que a posição do centro instantâneo de rotação está restrita à pertencer a linha que passa pelo centro das rodas fixas. Esta posição será determinada diretamente por η para robôs da classe (2,0) ou através da orientação das rodas centradas para robôs da classe (1,1).

No entanto, deve-se ressaltar que o número de graus de liberdade que podem ser acessados diretamente é igual a δ_m , já que apenas os graus de liberdade acessados por η podem ser acessados diretamente. A ação de ζ nas coordenadas de postura é indireta, pois é obtida através da variável β_c , calculada pela integral de ζ . Fisicamente, isto pode ser verificado considerando-se que ζ modifica apenas a orientação das rodas centradas do robô e portanto, apenas influenciará as coordenadas de postura do robô se este se mover. Portanto,

para um mesmo valor de δ_M , robôs com δ_m maior são mais manobráveis. Como seria de esperar, o máximo de manobrabilidade é obtido com robôs omnidirecionais, quando tem-se $\delta_M = \delta_m = 3$.

Pode ser provado [14] que este modelo cinemático de postura é irreduzível, ou seja, não existe uma transformação de coordenadas tal que uma das coordenadas é identicamente nula.

II.2. Modelo Cinemático de Configuração

O modelo cinemático de postura foi obtido utilizando-se apenas um subconjunto das expressões (II.22) e (II.23) correspondentes às restrições (II.27) impostas pelas rodas fixas e pelas rodas orientáveis centradas. Utilizando-se as demais restrições, pode-se obter expressões para as velocidades angulares, $\dot{\beta}_{oc}$, e rotacionais, $\dot{\phi}$, não consideradas no modelo cinemático de postura.

De (II.22) e (II.23) obtém-se

$$\dot{\beta}_{oc} = -C_{2oc}^{-1} C_{1oc}(\beta_{oc})^c R_0 \xi \quad (II.37)$$

$$\dot{\phi} = -J_2^{-1} J_1(\beta_c, \beta_{oc})^c R_0 \xi \quad (II.38)$$

substituindo-se ξ de (II.34) resulta

$$\dot{\beta}_{oc} = D(\beta_{oc}) \Sigma(\beta_{oc}) \eta \quad (II.39)$$

$$\dot{\phi} = E(\beta_c, \beta_{oc}) \Sigma(\beta_{oc}) \eta \quad (II.40)$$

sendo $D(\beta_{oc}) = -C_{2oc}^{-1} C_{1oc}(\beta_{oc})$ e $E(\beta_c, \beta_{oc}) = -J_2^{-1} J_1(\beta_c, \beta_{oc})$.

Definindo-se o vetor de coordenadas de configuração

$$q = \begin{bmatrix} \xi \\ \beta_c \\ \beta_{oc} \\ \phi \end{bmatrix} \quad (II.41)$$

pode-se escrever as expressões (II.35), (II.39) e (II.40) na forma

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{u} \quad (\text{II.42})$$

com

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} {}^0\mathbf{R}_c \Sigma(\beta_c) & 0 \\ 0 & \mathbf{I} \\ \mathbf{D}(\beta_{oc}) \Sigma(\beta_c) & 0 \\ \mathbf{E}(\beta_c, \beta_{oc}) \Sigma(\beta_c) & 0 \end{bmatrix} \quad (\text{II.43})$$

$$\mathbf{u} = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (\text{II.44})$$

que se constitui o modelo cinemático de configuração.

Considerando-se a distribuição

$$\Delta_c(\mathbf{q}) \stackrel{\Delta}{=} \text{span}\{\text{col}(\mathbf{S}(\mathbf{q}))\} \quad (\text{II.45})$$

tem-se que

$$\delta_m + N_c = \dim(\Delta_c) \leq \dim(\text{inv}(\Delta_c)) \leq \dim(\mathbf{q}) = 3 + N_c + N_{oc} + N \quad (\text{II.46})$$

onde $\text{inv}(\Delta_c)$ denota o fechamento involutivo da distribuição Δ_c . Conseqüentemente, pelo teorema de Frobenius [28] tem-se que o modelo cinemático de configuração é redutível. O número de restrições de velocidades que não podem ser integradas, e portanto não podem ser eliminadas é representado pelo **grau de não holonomicidade** do robô.

Definição II.4. Grau de Não Holonomicidade

grau de não holonomicidade de um robô móvel é definido como

$$M = \dim(\text{inv}(\Delta_c)) - (\delta_m + N_c) \quad (\text{II.47})$$

■

Como o modelo cinemático de configuração é redutível tem-se que para todas as classes de robôs móveis $M > 0$. Ou seja, todas as classes de robôs móveis possuem restrições não holonômicas. Pode-se notar ainda que M depende da estrutura particular de cada robô, e portanto não tem o mesmo valor para todos os robôs de uma mesma classe.

Por outro lado, o número de restrições de velocidade que podem ser integradas e portanto eliminadas é dado pela diferença entre $\dim(\mathbf{q})$ e $\dim(\text{inv}(\Delta_1))$.

II.3. Modelo Dinâmico de Configuração

Os modelos apresentados nas seções anteriores descrevem o comportamento do robô em função das velocidades das rodas. No entanto, fisicamente, as variáveis de entrada de um robô móvel são os torques aplicados pelos motores.

Utilizando-se o formalismo de Lagrange, tem-se que a dinâmica de um robô móvel com rodas é dada por [14]

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} = {}^0\mathbf{R}_c \mathbf{J}_1^T(\beta_c, \beta_{oc}) \lambda + {}^0\mathbf{R}_c \mathbf{C}_1^T(\beta_c, \beta_{oc}) \mu \quad (\text{II.48})$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_{oc}} \right) - \frac{\partial T}{\partial \beta_{oc}} = \mathbf{C}_2^T(\beta_c, \beta_{oc}) \mu + \tau_{oc} \quad (\text{II.49})$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) - \frac{\partial T}{\partial \phi} = \mathbf{J}_2^T \lambda + \tau_{\phi} \quad (\text{II.50})$$

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_c} \right) - \frac{\partial T}{\partial \beta_c} = \tau_c \quad (\text{II.51})$$

onde τ_{ϕ} e τ_c τ_{oc} são os torques aplicados para rotação das rodas, orientação das rodas centradas e orientação das rodas não centradas, respectivamente. T representa a energia cinética, λ é o coeficiente de Lagrange associado à restrição (II.22) e μ é o coeficiente de Lagrange associado à restrição (II.23).

Pré-multiplicando-se as expressões (II.48), (II.49) e (II.50) por ${}^c\mathbf{R}_0$, $\mathbf{D}^T(\beta_{oc})$ e $\mathbf{E}^T(\beta_c, \beta_{oc})$, respectivamente, e somando-as, obtém-se

$$\begin{aligned}
& {}^c\mathbf{R}_0 \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} \right) + \mathbf{D}^T(\beta_{oc}) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_{oc}} \right) - \frac{\partial T}{\partial \beta_{oc}} \right) + \mathbf{E}^T(\beta_c, \beta_{oc}) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) - \frac{\partial T}{\partial \phi} \right) = \\
& = (\mathbf{J}_1(\beta_c, \beta_{oc}) + \mathbf{J}_2 \mathbf{E}(\beta_c, \beta_{oc}))^T \lambda + (\mathbf{C}_1(\beta_c, \beta_{oc}) + \mathbf{C}_2 \mathbf{D}(\beta_{oc}))^T \mu + \\
& + \mathbf{D}^T(\beta_{oc}) \tau_{oc} + \mathbf{E}^T(\beta_c, \beta_{oc}) \tau_\phi
\end{aligned} \quad (II.52)$$

porém, das definições de $\mathbf{D}^T(\beta_{oc})$ e $\mathbf{E}^T(\beta_c, \beta_{oc})$ tem-se

$$\mathbf{C}_{1oc}(\beta_{oc}) + \mathbf{C}_{2oc} \mathbf{D}(\beta_{oc}) = 0 \quad (II.53)$$

$$\mathbf{J}_1(\beta_c, \beta_{oc}) + \mathbf{J}_2 \mathbf{E}(\beta_c, \beta_{oc}) = 0 \quad (II.54)$$

e portanto, os coeficientes de Lagrange são eliminados da expressão (II.52), resultando

$$\begin{aligned}
& {}^c\mathbf{R}_0 \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\xi}} \right) - \frac{\partial T}{\partial \xi} \right) + \mathbf{D}^T(\beta_{oc}) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\beta}_{oc}} \right) - \frac{\partial T}{\partial \beta_{oc}} \right) + \mathbf{E}^T(\beta_c, \beta_{oc}) \left(\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\phi}} \right) - \frac{\partial T}{\partial \phi} \right) = \\
& = \mathbf{D}^T(\beta_{oc}) \tau_{oc} + \mathbf{E}^T(\beta_c, \beta_{oc}) \tau_\phi
\end{aligned} \quad (II.55)$$

A energia cinética de um robô móvel pode ser expressa por

$$T = \frac{1}{2} \dot{\xi}^T \mathbf{R}_c [\mathbf{M}(\beta_{oc}) {}^c\mathbf{R}_0 \dot{\xi} + 2\mathbf{V}(\beta_{oc}) \dot{\beta}_{oc} + 2\mathbf{W} \dot{\beta}_c] + \frac{1}{2} \dot{\beta}_{oc}^T \mathbf{I}_{oc} \dot{\beta}_{oc} + \frac{1}{2} \dot{\phi}^T \mathbf{I}_\phi \dot{\phi} + \frac{1}{2} \dot{\beta}_c^T \mathbf{I}_c \dot{\beta}_c \quad (II.56)$$

onde $\mathbf{M}(\beta_{oc})$, $\mathbf{V}(\beta_{oc})$, \mathbf{W} , \mathbf{I}_{oc} , \mathbf{I}_ϕ e \mathbf{I}_c são funções dos parâmetros de massa e inércia dos vários corpos rígidos que formam o robô.

Substituindo-se (II.56) em (II.51) e (II.55) e eliminando-se as velocidades $\dot{\xi}$, $\dot{\beta}_{oc}$, $\dot{\phi}$ e $\dot{\beta}_c$ e as acelerações $\ddot{\xi}$, $\ddot{\beta}_{oc}$, $\ddot{\phi}$ e $\ddot{\beta}_c$ com a utilização das expressões (II.35), (II.39) e (II.40) e suas derivadas obtém-se

$$\mathbf{H}_1(\beta_c, \beta_{oc}) \ddot{\eta} + \Sigma^T(\beta_c) \mathbf{V}(\beta_{oc}) \ddot{\xi} + \mathbf{f}_1(\beta_c, \beta_{oc}, \eta, \xi) = \Sigma^T(\beta_c) [\mathbf{D}^T(\beta_{oc}) \tau_{oc} + \mathbf{E}^T(\beta_c, \beta_{oc}) \tau_\phi] \quad (II.57)$$

com

$$\begin{aligned}
\mathbf{H}_1(\beta_c, \beta_{oc}) &= \Sigma^T(\beta_c) [\mathbf{M}(\beta_{oc}) + \mathbf{D}^T(\beta_{oc}) \mathbf{V}^T(\beta_{oc}) + \mathbf{V}(\beta_{oc}) \mathbf{D}(\beta_{oc}) + \mathbf{D}^T(\beta_{oc}) \mathbf{I}_{oc} \mathbf{D}(\beta_{oc}) + \\
& + \mathbf{E}^T(\beta_c, \beta_{oc}) \mathbf{I}_\phi \mathbf{E}(\beta_c, \beta_{oc})] \Sigma(\beta_c)
\end{aligned}$$

e

$$\mathbf{V}^T(\beta_{oc}) \Sigma(\beta_c) \ddot{\eta} + \mathbf{I}_c \ddot{\xi} + \mathbf{f}_2(\beta_c, \beta_{oc}, \eta, \xi) = \tau_c \quad (II.58)$$

que juntamente com (II.35), (II.39) e (II.40) formam o modelo dinâmico de configuração de um robô móvel com rodas genérico. Neste modelo genérico, os torques τ_ϕ , τ_{oc} e τ_c representam os torques que podem ser aplicados para rotação e orientação das rodas. No entanto, em um robô real, apenas alguns destes torques são efetivamente aplicados, pois geralmente utiliza-se o número mínimo de motores necessários. Pode-se facilmente concluir que cada roda centrada deve necessariamente possuir pelo menos um motor para sua orientação, pois caso contrário comportar-se-ia como uma roda fixa. Conseqüentemente, o vetor τ_c não pode possuir componentes identicamente nulos. Por outro lado, os vetores τ_ϕ e τ_{oc} podem possuir componentes identicamente nulos, desde que a rotação e a orientação das rodas às quais estes estejam associados possa ser obtida através do acionamento das demais rodas. Tem-se então, que o vetor dos torques fornecidos para rotação e orientação das rodas não centradas τ_m pode ser obtido de

$$\begin{bmatrix} \tau_{oc} \\ \tau_\phi \end{bmatrix} = \mathbf{P} \tau_m \quad (\text{II.59})$$

onde \mathbf{P} é uma matriz $(N_{oc}+N) \times N_m$ que seleciona os componentes de $[\tau_{oc} \ \tau_\phi]^T$ que são efetivamente utilizados como entradas de controle. A expressão (II.57) pode portanto, ser reescrita como

$$\mathbf{H}_1(\beta_c, \beta_{oc}) \dot{\eta} + \Sigma^T(\beta_c) \mathbf{V}(\beta_{oc}) \dot{\xi} + \mathbf{f}_1(\beta_c, \beta_{oc}, \eta, \xi) = \mathbf{B}(\beta_c, \beta_{oc}) \mathbf{P} \tau_m \quad (\text{II.60})$$

sendo $\mathbf{B}(\beta_c, \beta_{oc}) \stackrel{\Delta}{=} \Sigma^T(\beta_c) [\mathbf{D}^T(\beta_{oc}) \ \mathbf{E}^T(\beta_c, \beta_{oc})]$

É importante observar que a matriz $\mathbf{B}(\beta_c, \beta_{oc}) \mathbf{P}$ deve possuir *rank* completo para quaisquer valores de β_c e β_{oc} . Se esta condição não for satisfeita, existirão valores de β_c e β_{oc} para os quais o robô tornar-se-á sub-atuado, ou seja, não existirão graus de liberdade suficientes nas entradas de controle para determinar a alocação do centro instantâneo de

rotação do robô. Esta condição portanto, determina o número mínimo de motores que devem ser utilizados para um determinada classe de robô móvel, conforme explicitado na Tab. II-II.

Tab. II-II - Número mínimo de motores para cada classe de robô móvel.

Classe	Motores
(3,0)	3 ¹ ou 4 ²
(2,0)	2
(2,1)	3
(1,1)	2
(1,2)	4

O modelo dinâmico de configuração pode ser escrito de forma mais simples como

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{u} \\ \mathbf{H}(\boldsymbol{\beta})\ddot{\mathbf{q}} + \mathbf{f}(\boldsymbol{\beta}, \mathbf{u}) = \mathbf{F}(\boldsymbol{\beta})\boldsymbol{\tau}_0 \end{cases} \quad (\text{II.61})$$

com as seguintes definições

$$\boldsymbol{\beta}^{\Delta} = \begin{bmatrix} \beta_c \\ \beta_{oc} \end{bmatrix}$$

$$\mathbf{q}^{\Delta} = \begin{bmatrix} \xi \\ \beta \\ \varphi \end{bmatrix}$$

$$\mathbf{u}^{\Delta} = \begin{bmatrix} \eta \\ \zeta \end{bmatrix}$$

$$\mathbf{H}(\boldsymbol{\beta})^{\Delta} = \begin{bmatrix} \mathbf{H}_1(\beta_c, \beta_{oc}) & \boldsymbol{\Sigma}^T(\beta_c)\mathbf{V}(\beta_{oc}) \\ \mathbf{V}^T(\beta_{oc})\boldsymbol{\Sigma}(\beta_c) & \mathbf{I}_c \end{bmatrix}$$

¹utilizando rodas suecas

²utilizando rodas convencionais

$$\mathbf{f}(\beta, \mathbf{u}) \stackrel{\Delta}{=} \begin{bmatrix} f_1(\beta_c, \beta_{oc}, \eta, \zeta) \\ f_2(\beta_c, \beta_{oc}, \eta, \zeta) \end{bmatrix}$$

$$\mathbf{F}(\beta) \stackrel{\Delta}{=} \begin{bmatrix} \mathbf{B}(\beta_c, \beta_{oc})\mathbf{P} & 0 \\ 0 & \mathbf{I} \end{bmatrix}$$

$$\tau_0 \stackrel{\Delta}{=} \begin{bmatrix} \tau_m \\ \tau_c \end{bmatrix}$$

II.4. Modelo Dinâmico de Postura

O modelo dinâmico de configuração de um robô móvel com rodas pode ser simplificado para

$$\begin{cases} \dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{u} \\ \dot{\mathbf{u}} = \mathbf{v} \end{cases} \quad (\text{II.62})$$

utilizando-se a seguinte realimentação de estados

$$\tau_0 = \mathbf{F}^\dagger(\beta)[\mathbf{H}(\beta)\mathbf{v} - \mathbf{f}(\beta, \mathbf{u})] \quad (\text{II.63})$$

onde $\mathbf{F}^\dagger(\beta)$ denota uma inversa à esquerda de $\mathbf{F}(\beta, \mathbf{u})$.

Por outro lado, para aplicações de controle em geral, o que interessa são basicamente nas coordenadas de postura do robô, isto é, ξ . Os valores das variáveis internas β_{oc} e φ não são de interesse e conseqüentemente, podem ser ignorados. Com isto obtém-se o seguinte modelo dinâmico de postura

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{B}(\mathbf{x})\mathbf{u} \\ \dot{\mathbf{u}} = \mathbf{v} \end{cases} \quad (\text{II.64})$$

mantendo-se as mesmas definições anteriores para \mathbf{x} e \mathbf{u} .

Este modelo descreve totalmente a dinâmica entre as coordenadas de postura ξ e as entradas de controle \mathbf{v} . Embora as coordenadas β_{oc} e φ tenham aparentemente desaparecido do modelo, elas permanecem implicitamente na realimentação definida pela expressão (II.63). As propriedades estruturais deste modelo são basicamente as mesmas do modelo cinemático de

postura. Portanto, este modelo também é genérico, no sentido de que é válido para qualquer classe possível de robô com rodas, e irreduzível [14].

II.5. Conclusão

Neste capítulo foram apresentados os modelos de robôs móveis com rodas. Foram considerados quatro tipos de rodas: rodas fixas, rodas orientáveis centradas, rodas orientadas não-centradas e rodas suecas. Sob hipóteses bastante realistas, como a de que as rodas não deslizam, mostrou-se que existem apenas cinco classes de robôs com rodas. Através da análise das características do modelo dinâmico foi possível deduzir o número mínimo de motores necessários para que robôs de uma determinada classe não sejam sub-atuados.

Capítulo III

Controle Linearizante

Neste capítulo, serão apresentadas técnicas para controle de robôs móveis baseadas na linearização do modelo cinemático (II.36) ou do modelo dinâmico (II.64). Serão considerados apenas os modelos de postura, pois o objetivo é controlar a posição e a orientação do robô. Comparando-se os modelos de configuração (II.42) e (II.36, II.39, II.40, II.57, II.58) com os modelos de postura, pode-se observar que as variáveis desprezadas não tem qualquer influência sobre a dinâmica da variável ξ e portanto, não fazem parte do estado do sistema dinâmico que gera a posição e a orientação do robô.

O interesse por uma estratégia linearizante deve-se ao fato de que se o sistema puder ser linearizado, uma lei de controle poderá ser projetada valendo-se da grande quantidade de resultados teóricos disponíveis para sistemas lineares. Além disto, com a linearização do sistema, espera-se também simplificar o problema de estimação da posição e orientação do robô. Por exemplo, utilizando-se o filtro de Kalman linear, ao invés do filtro de Kalman estendido [23].

Certamente, a técnica de linearização mais conhecida consiste em expandir-se o sistema em série de Taylor em torno de um ponto e desprezar os termos de ordem superior. No entanto, utilizando-se realimentação, pode-se obter uma linearização onde os termos de ordem superior não são desprezados, mas sim cancelados através da manipulação adequada das entradas do sistema. A vantagem deste segundo enfoque é que, a princípio, obtém-se uma descrição linear exata para o sistema não linear, ao invés de uma aproximação linear para o sistema não linear. Por outro lado, a principal desvantagem da linearização por realimentação reside no fato de que se os parâmetros e o estado do sistema não forem bem conhecidos, o

cancelamento dos termos de ordem superior não será exato e portanto, a descrição linear obtida não corresponderá ao sistema original. Conseqüentemente, a lei de controle linear projetada para o sistema linearizado, poderá perder eficiência e o sistema poderá ter o seu desempenho alterado ou até mesmo tornar-se instável.

Na maioria das vezes, no entanto, os parâmetros do sistema podem ser adequadamente obtidos através de calibração. Quando o estado do sistema não pode ser adequadamente medido através de sensores, pode-se utilizar técnicas de estimação, como discutido no capítulo VI. Além disso, para obtenção do modelo dinâmico de postura (II.64), foi assumido, implicitamente, que os parâmetros do robô são perfeitamente conhecidos e o estado do sistema é acessível. Caso contrário, a realimentação (II.63) não poderia ser computada. Portanto, a necessidade do conhecimento dos parâmetros e do estado do sistema para obter-se uma linearização por realimentação é uma exigência que já feita anteriormente. Com o objetivo de relaxar esta exigência, no capítulo V são desenvolvidos controladores adaptativos.

Técnicas de linearização por realimentação exata de estados [28] são empregadas em [77] para a determinação da lei de controle para um robô da classe (2,0). No entanto, é obtido o controle apenas de posição. A orientação não é controlada. Aqui, este resultado é generalizado para as demais classes de robôs móveis. Além disso, mostra-se que para as classe (3,0) é possível obter-se também controle de orientação. Para as classes (1,1) e (1,2), mostra-se que esta estratégia não tem aplicação prática.

III.1. Linearização Entrada-Saída por Realimentação Estática de Estados

Considere, com base em [61][62], a classe de sistemas não-lineares multivariáveis com o mesmo número de entradas e saídas, da forma

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}_1(\mathbf{x})u_1 + \dots + \mathbf{g}_p(\mathbf{x})u_p \quad (\text{III.1})$$

$$\begin{aligned} y_i &= h_i(\mathbf{x}) \\ \mathbf{M} \\ y_p &= h_p(\mathbf{x}) \end{aligned} \quad (III.2)$$

com $\mathbf{x} \in \mathfrak{R}^n$, $\mathbf{u} \in \mathfrak{R}^p$ e $\mathbf{y} \in \mathfrak{R}^p$, assumindo-se \mathbf{f} , \mathbf{g}_i , h_j suaves. Diferenciando com relação ao tempo a saída y_j , obtém-se

$$\dot{y}_j = L_f h_j(\mathbf{x}) + \sum_{i=1}^p (L_{g_i} h_j(\mathbf{x})) u_i \quad (III.3)$$

com $L_f h_j(\mathbf{x})$ representando a derivada de Lie da saída $h_j(\mathbf{x})$ com relação à $\mathbf{f}(\mathbf{x})$, dada por

$$L_f h_j(\mathbf{x}) = \sum_{i=1}^n \frac{\partial h_j(\mathbf{x})}{\partial x_i} \mathfrak{X}_{f_i} \quad (III.4)$$

e

$$L_{g_i} h_j(\mathbf{x}) = \sum_{i=1}^n \frac{\partial h_j(\mathbf{x})}{\partial x_i} \mathfrak{X}_{g_i} \quad (III.5)$$

onde \mathfrak{X}_{f_i} e \mathfrak{X}_{g_i} representam, respectivamente, as partes relativas à $\mathbf{f}(\mathbf{x})$ e $\mathbf{g}(\mathbf{x})$ da i -ésima equação de estado do sistema representado por (III.1) (III.2).

De acordo com (III.3), se cada um dos termos $L_{g_i} h_j(\mathbf{x}) = 0$, nenhuma das entradas surgirá na derivada temporal das saídas do processo. Define-se γ_j como sendo o menor inteiro tal que no mínimo uma das entradas apareçam em $y_j^{\gamma_j}$, isto é,

$$y_j^{\gamma_j} = L_f^{\gamma_j} h_j(\mathbf{x}) + \sum_{i=1}^p L_{g_i} (L_f^{\gamma_j-1} h_j(\mathbf{x})) u_i \quad (III.6)$$

sendo

$$L_f^{\gamma_j} h_j(\mathbf{x}) = \sum_{i=1}^n \frac{\partial L_f^{\gamma_j-1} h_j(\mathbf{x})}{\partial x_i} \mathfrak{X}_{f_i} \quad (III.7)$$

e

$$L_g L_f^{\gamma_j-1} h_j(\mathbf{x}) = \sum_{i=1}^n \frac{\partial L_g L_f^{\gamma_j-2} h_j(\mathbf{x})}{\partial x_i} \mathfrak{X}_{g_i} \quad (III.8)$$

com no mínimo um dos termos

$$L_{g_i}(L_f^{\gamma_i-1}h_j(\mathbf{x})) \neq 0 \quad (\text{III.9})$$

para todo \mathbf{x} pertencente a região onde a linearização é válida.

Define-se a matriz $E(\mathbf{x})$ como

$$E(\mathbf{x}) = \begin{bmatrix} L_{g_i} L_f^{\gamma_i-1} h_1(\mathbf{x}) & \Lambda & L_{g_p} L_f^{\gamma_i-1} h_1(\mathbf{x}) \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ L_{g_i} L_f^{\gamma_p-1} h_p(\mathbf{x}) & \Lambda & L_{g_p} L_f^{\gamma_p-1} h_p(\mathbf{x}) \end{bmatrix} \quad (\text{III.10})$$

uma matriz quadrada de ordem p . Pode-se, então, reescrever a equação (III.6) na forma

$$\begin{bmatrix} y_1^{\gamma_1} \\ \mathbf{M} \\ y_p^{\gamma_p} \end{bmatrix} = \begin{bmatrix} L_f^{\gamma_1} h_1(\mathbf{x}) \\ \mathbf{M} \\ L_f^{\gamma_p} h_p(\mathbf{x}) \end{bmatrix} + E(\mathbf{x}) \begin{bmatrix} u_1 \\ \mathbf{M} \\ u_p \end{bmatrix} \quad (\text{III.11})$$

Naturalmente, se a matriz inversa de $E(\mathbf{x})$ existir para todo o \mathbf{x} pertencente à região de interesse, a lei de controle por realimentação de estados

$$\mathbf{u}(\mathbf{x}) = -E^{-1}(\mathbf{x}) \begin{bmatrix} L_f^{\gamma_1} h_1(\mathbf{x}) \\ \mathbf{M} \\ L_f^{\gamma_p} h_p(\mathbf{x}) \end{bmatrix} + E^{-1}(\mathbf{x}) \mathbf{v} \quad (\text{III.12})$$

resulta, em malha fechada, no seguinte sistema,

$$\begin{bmatrix} y_1^{\gamma_1} \\ \mathbf{M} \\ y_p^{\gamma_p} \end{bmatrix} = \begin{bmatrix} v_1 \\ \mathbf{M} \\ v_p \end{bmatrix} \quad (\text{III.13})$$

linear sob o ponto de vista entrada-saída. Na Fig. III-1 é mostrado um diagrama de blocos do controle linearizante.

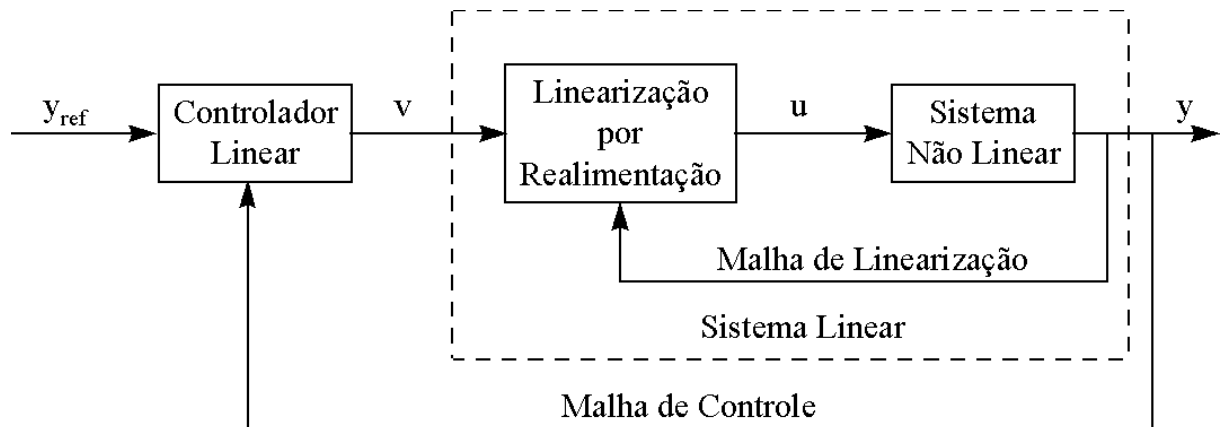


Fig. III-1 - Diagrama de blocos do controle linearizante.

III.2. Ponto de Referência

Semelhante a [77], elege-se um ponto, *a priori* arbitrário, da estrutura do robô como ponto de referência. Ou seja, a posição do robô é tomada como sendo a posição deste ponto. A este ponto de referência, associa-se um sistema de coordenadas $\{X_r, Y_r\}$, tal como mostrado na Fig. III-2. A orientação deste novo sistema de coordenadas é a mesma do sistema de coordenadas $\{X_c, Y_c\}$.

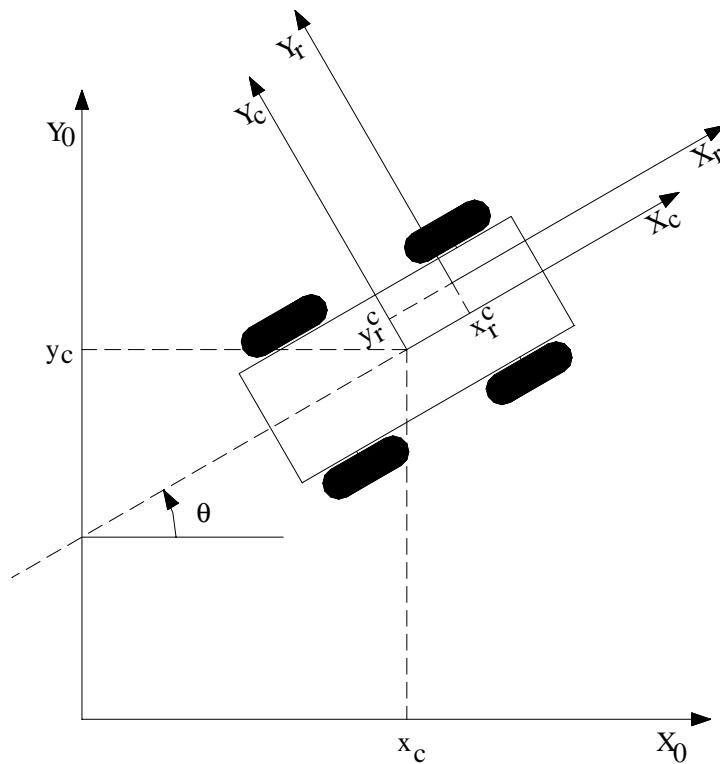


Fig. III-2 - Sistema de coordenadas do ponto de referência para controle do robô.

O ponto de referência pode ser descrito no sistema de coordenadas global $\{X_0, Y_0\}$ através da relação

$$\xi_r = \begin{bmatrix} x_r \\ y_r \\ \theta \end{bmatrix} = {}^0\mathbf{R}_c \begin{bmatrix} x_r^c \\ y_r^c \\ 0 \end{bmatrix} + \xi \quad (\text{III.14})$$

onde ξ_r representa as coordenadas do ponto de referência no sistema de coordenadas global e $[x_r^c, y_r^c]^T$ representa a posição da origem do sistema de coordenadas $\{X_r, Y_r\}$ representada no sistema de coordenadas $\{X_c, Y_c\}$.

Como será mostrado a seguir, a escolha do ponto de referência tem grande influência no desenvolvimento da lei de realimentação linearizante. Um dos fatores mais importantes, diz respeito às restrições impostas à posição deste ponto. Estas restrições são diferentes para cada classe de robô. Além disso, a classe do robô também está relacionada com as variáveis de estado que podem ser linearizadas. Portanto, a realimentação linearizante será desenvolvida especificamente para cada classe de robô.

Aspectos adicionais, não abordados aqui, sobre a escolha do ponto de referência podem ser encontrados em [48].

III.3. Realimentação Linearizante para a Classe (3,0)

Para esta classe de robô, o modelo dinâmico de postura (II.64) pode ser escrito mais explicitamente na forma

$$\begin{cases} \begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \\ \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \\ \dot{\eta}_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \end{cases} \quad (\text{III.15})$$

ou ainda

$$\begin{bmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta} \\ \dot{\eta}_1 \\ \dot{\eta}_2 \\ \dot{\eta}_3 \end{bmatrix} = \begin{bmatrix} \eta_1 \cos \theta - \eta_2 \sin \theta \\ \eta_1 \sin \theta + \eta_2 \cos \theta \\ \eta_3 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (\text{III.16})$$

que está na forma (III.1). Definindo-se a saída do sistema como sendo

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \xi_r \quad (\text{III.17})$$

pode-se aplicar o algoritmo de linearização descrito na seção III.1. Assim sendo, para que pelo menos uma das entradas v_i surja explicitamente na derivada temporal da saída, é necessário derivar y duas vezes. Com isto obtém-se

$$\dot{\mathbf{y}} = L_f^2 \mathbf{h}(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{v} \quad (\text{III.18})$$

com

$$\mathbf{E}(\mathbf{x}) = \begin{bmatrix} \cos \theta & -\sin \theta & -(x_r^c \sin \theta + y_r^c \cos \theta) \\ \sin \theta & \cos \theta & (x_r^c \cos \theta + y_r^c \sin \theta) \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{III.19})$$

e

$$L_f^2 \mathbf{h}(\mathbf{x}) = \begin{bmatrix} -\eta_1 \eta_3 \sin \theta - \eta_2 \eta_3 \cos \theta - \eta_3^2 (x_r^c \cos \theta - y_r^c \sin \theta) \\ \eta_1 \eta_2 \cos \theta - \eta_2 \eta_3 \sin \theta - \eta_3^2 (x_r^c \sin \theta + y_r^c \cos \theta) \\ 0 \end{bmatrix} \quad (\text{III.20})$$

A lei de realimentação linearizante será então dada por

$$\mathbf{v}(\mathbf{x}) = -\mathbf{E}^{-1}(\mathbf{x})L_f^2 \mathbf{h}(\mathbf{x}) + \mathbf{E}^{-1}(\mathbf{x})\bar{\mathbf{u}} \quad (\text{III.21})$$

Para esta classe de robôs em particular, a inversão da matriz $\mathbf{E}(\mathbf{x})$ não apresenta problemas. Logo, para que seja possível o cálculo de (III.21), resta apenas determinar o vetor de entradas equivalentes $\bar{\mathbf{u}}$. Além disso, com esta linearização, obtém-se um desacoplamento

completo entre as coordenadas de postura do robô, existindo uma entrada para controlar cada uma das variáveis de posição e orientação.

III.4. Realimentação Linearizante para a Classe (2,0)

Diferentemente do que ocorre para a classe anterior, agora se tem apenas duas entradas para o sistema, como pode ser observado escrevendo-se o modelo dinâmico de postura (II.64) na sua forma explícita

$$\begin{bmatrix} \ddot{x}_c \\ \ddot{y}_c \\ \ddot{\theta} \\ \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} \eta_1 \cos \theta \\ \eta_1 \sin \theta \\ \eta_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (\text{III.22})$$

Conseqüentemente, pode-se forçar a linearização de apenas duas das variáveis de estado. A saída do sistema é então definida como

$$y = h(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \xi_r \quad (\text{III.23})$$

ou seja, apenas a posição do robô é tomada como saída do sistema. Convém ressaltar aqui, que devido ao fato da orientação não ser linearizada, não será possível garantir a sua convergência para um valor desejado. No entanto, dependendo da aplicação, esta limitação pode ser contornada por outros métodos. Em alguns robôs, como o RAMSIS [26], o corpo do robô pode ter a sua orientação controlada independentemente da orientação da base utilizada para locomoção. Outro exemplo é quando se tem um manipulador montado sobre o veículo. Neste caso, pode ser que o manipulador consiga operar independentemente da orientação do veículo.

Utilizando-se o algoritmo para cálculo do controle linearizante, obtém-se

$$\mathbf{v}(\mathbf{x}) = -\mathbf{E}^{-1}(\mathbf{x})\mathbf{L}_f^2 h(\mathbf{x}) + \mathbf{E}^{-1}(\mathbf{x})\bar{\mathbf{u}} \quad (\text{III.24})$$

com

$$E(x) = \begin{bmatrix} \cos \theta & -(x_r^c \sin \theta + y_r^c \cos \theta) \\ \sin \theta & (x_r^c \cos \theta - y_r^c \sin \theta) \end{bmatrix} \quad (\text{III.25})$$

e

$$L_r^2 h(x) = \begin{bmatrix} -\eta_1 \eta_2 \sin \theta - \eta_2^2 (x_r^c \cos \theta - y_r^c \sin \theta) \\ \eta_1 \eta_2 \cos \theta - \eta_2^2 (x_r^c \sin \theta + y_r^c \cos \theta) \end{bmatrix} \quad (\text{III.26})$$

Pode ser facilmente verificado, que $E(x)$ é singular quando $x_r^c = 0$, portanto, o ponto de referência não poderá estar sobre o eixo comum à todas as rodas fixas. Isto ocorre devido ao fato de que pontos sobre este eixo podem mover-se, instantaneamente, em apenas uma dimensão, enquanto os demais pontos podem se mover em duas dimensões.

É interessante observar que a escolha de um ponto de referência deslocado em relação ao eixo das rodas permite obter uma linearização para o sistema. Se não fosse feito este deslocamento, apenas uma das entradas do sistema surgiria explicitamente na expressão de \dot{y} e a linearização não seria obtida. Isto ocorre porque nesta situação, uma das entradas afeta a saída após duas integrações e a outra entrada afeta a saída somente após três integrações. Ao deslocar-se o ponto de referência, força-se um acoplamento entre as variáveis de estado de tal forma que a saída é afetada pelas duas entradas após o mesmo número de integrações.

Convém notar que para o sistema (III.22), a expressão (III.18) também se aplica. Logo, este sistema possui grau relativo igual a quatro, vide [28] para detalhes. Deste modo, a dinâmica zero é de primeira ordem. Para tarefas de *parking*, tem-se que η_1 e η_2 convergem para zero, e portanto uma inspeção direta de (III.22) revela que todos os estados são limitados. Por outro lado, para tarefas gerais far-se-ia necessária a investigação da estabilidade da dinâmica zero. Isto requer a representação do sistema na forma normal, o que exige o cálculo de um mapeamento inverso não-linear de difícil determinação.

Uma outra forma de obter-se uma linearização por realimentação de estados, sem efetuar o deslocamento do ponto de referência (ou seja, com $x_r^c = 0$) seria derivar a saída do

sistema mais vezes, até que todas as entradas surgissem explicitamente. Obviamente, no caso do sistema descrito por (III.22) e (III.23), ambas as entradas surgiriam explicitamente na expressão de $\ddot{\theta}$. Na realidade, surgiriam também derivadas das entradas. A princípio, este fato é interessante, pois como as entradas e suas derivadas são independentes entre si, estariam sendo criadas variáveis de entrada adicionais. Estas entradas adicionais permitiriam que fosse definido um vetor de saída com mais componentes, por exemplo, linearizando-se também a orientação. Esta extensão do algoritmo de linearização por realimentação de estados é denominada linearização dinâmica por realimentação de estados [28][50]. No entanto, procedendo-se desta maneira, obter-se-ia a uma matriz $E(x)$, que seria singular quando $\eta_1=0$. Ou seja, este tipo de linearização apenas seria possível quando o robô estivesse em movimento. Como neste trabalho deseja-se abordar o problema de convergência para um ponto e não para uma trajetória, esta abordagem não é vantajosa. Para maiores detalhes, vide [70] onde esta abordagem é utilizada como parte de um controle híbrido, com o objetivo de rastrear uma trajetória e estabilizar o robô no ponto final da trajetória. A estabilização do robô quando não em movimento é obtida utilizando-se uma estratégia de controle variante no tempo.

III.5. Realimentação Linearizante para a Classe (2,1)

Similarmente ao que ocorre na classe (3,0), aqui também existem três variáveis de entrada. Conseqüentemente, é possível definir-se uma saída a ser linearizada com três componentes. A tendência natural portanto, é incluir na saída como sendo igual a ξ_r , ou seja, definir

$$y = h(x) = \xi_r \quad (III.27)$$

Desta forma, aplicando-se o método para obtenção da realimentação linearizante, obtém-se

$$\begin{aligned} \dot{\mathbf{y}} = & \begin{bmatrix} -\eta_1 \eta_2 \sin(\theta + \beta_c) - \eta_1 \zeta \sin(\theta + \beta_c) - \eta_2^2 (x_r^c \cos \theta - y_r^c \sin \theta) \\ \eta_1 \eta_2 \cos(\theta + \beta_c) + \eta_1 \zeta \cos(\theta + \beta_c) - \eta_2^2 (x_r^c \sin \theta + y_r^c \cos \theta) \\ 0 \end{bmatrix} + \\ & + \begin{bmatrix} \cos(\theta + \beta_c) & -(x_r^c \sin \theta + y_r^c \cos \theta) & 0 \\ \sin(\theta + \beta_c) & (x_r^c \cos \theta - y_r^c \sin \theta) & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{v} = \mathbf{L}_f^2 \mathbf{h}(\mathbf{x}) + \mathbf{E}(\mathbf{x}) \mathbf{v} \end{aligned} \quad (\text{III.28})$$

de onde pode-se verificar que $\mathbf{E}(\mathbf{x})$ não é inversível, e portanto não poderá ser obtida a linearização. No entanto, se for definindo

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_r \\ \beta_c \end{bmatrix} \quad (\text{III.29})$$

será obtida a linearização, pois resultará

$$\begin{aligned} \dot{\mathbf{y}} = & \begin{bmatrix} -\eta_1 \eta_2 \sin(\theta + \beta_c) - \eta_1 \zeta \sin(\theta + \beta_c) - \eta_2^2 (x_r^c \cos \theta - y_r^c \sin \theta) \\ \eta_1 \eta_2 \cos(\theta + \beta_c) + \eta_1 \zeta \cos(\theta + \beta_c) - \eta_2^2 (x_r^c \sin \theta + y_r^c \cos \theta) \\ 0 \end{bmatrix} + \\ & + \begin{bmatrix} \cos(\theta + \beta_c) & -(x_r^c \sin \theta + y_r^c \cos \theta) & 0 \\ \sin(\theta + \beta_c) & (x_r^c \cos \theta - y_r^c \sin \theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v} = \mathbf{L}_f^2 \mathbf{h}(\mathbf{x}) + \mathbf{E}(\mathbf{x}) \mathbf{v} \end{aligned} \quad (\text{III.30})$$

e portanto,

$$\mathbf{v}(\mathbf{x}) = -\mathbf{E}^{-1}(\mathbf{x}) \mathbf{L}_f^2 \mathbf{h}(\mathbf{x}) + \mathbf{E}^{-1}(\mathbf{x}) \bar{\mathbf{u}} \quad (\text{III.31})$$

É importante notar que a matriz $\mathbf{E}(\mathbf{x})$ será singular quando $\tan(\beta_c) = -\frac{x_r^c}{y_r^c}$. Fisicamente,

isto significa que o ponto de referência não poderá estar localizado sobre a normal que passa pelo centro da roda orientável. Esta situação é semelhante ao que ocorre com a classe (2,0) porém, aqui o problema é agravado pelo fato de que a restrição imposta à escolha do ponto de referência é dependente de uma das variáveis de estado. Conseqüentemente, a região de restrição dependerá da trajetória seguida pelo robô. Por outro lado, normalmente a variável β_c tem sua faixa de valores admissíveis limitada pela própria construção do robô e portanto,

deverá existir uma região onde o ponto de referência possa ser alocado de forma que $E(x)$ nunca seja singular.

Outro aspecto a considerar é que, com esta definição de vetor de saída, perde-se a capacidade de controlar a orientação do robô, tal como acontece para a classe (2,0).

III.6. Realimentação Linearizante para a Classe (1,1)

Embora nesta classe existam duas variáveis de entrada, uma das variáveis de saída deverá necessariamente ser β_c , tal como ocorre na classe (2,1). Consequentemente, tem-se a liberdade de escolher apenas uma das variáveis de saída. Portanto, não será possível sequer obter-se o controle da posição do robô.

III.7. Realimentação Linearizante para a Classe (1,2)

Da mesma forma que na classe anterior, tem-se a liberdade para escolher apenas uma das três variáveis componentes do vetor de saída. As outras duas variáveis devem necessariamente ser β_{c1} e β_{c2} . Pelos mesmos motivos, não será possível controlar a orientação do robô.

III.8. Resultados de Simulação

Para exemplificar a efetividade do método proposto a lei de controle (III.24) foi simulada para o robô IEEE-Rover (vide Capítulo VII), considerando-se como vetor de entradas equivalentes

$$\bar{u} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} = \begin{bmatrix} y_{m_1} + \alpha_2 (y_{m_1} - x_r) + \alpha_1 (y_{m_1} - x_r) \\ y_{m_2} + \alpha_4 (y_{m_2} - y_r) + \alpha_3 (y_{m_2} - y_r) \end{bmatrix} \quad (\text{III.32})$$

onde y_{m_1} e y_{m_2} são as saídas dos modelos de referência associados a cada componente de (III.23), descritos pelas seguintes funções de transferência

$$G_{m_1}(s) = \frac{\alpha_1}{s^2 + \alpha_2 s + \alpha_1} \quad (\text{III.33})$$

$$G_{m_2}(s) = \frac{\alpha_3}{s^2 + \alpha_4 s + \alpha_3} \quad (\text{III.34})$$

com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

A trajetória espacial obtida para estabilizar o robô no ponto $x_r=0$, $y_r=0$ e $\theta=0$, a partir da condição inicial $x_r(0)=0\text{m}$, $y_r(0)=4\text{m}$ e $\theta(0)=0$, é mostrada na Fig. III-3. As Figs. III-4 a III-6 mostram as variáveis de posição e orientação em função do tempo. Convém notar, que como o robô não pode se mover lateralmente, este movimento é considerado difícil.

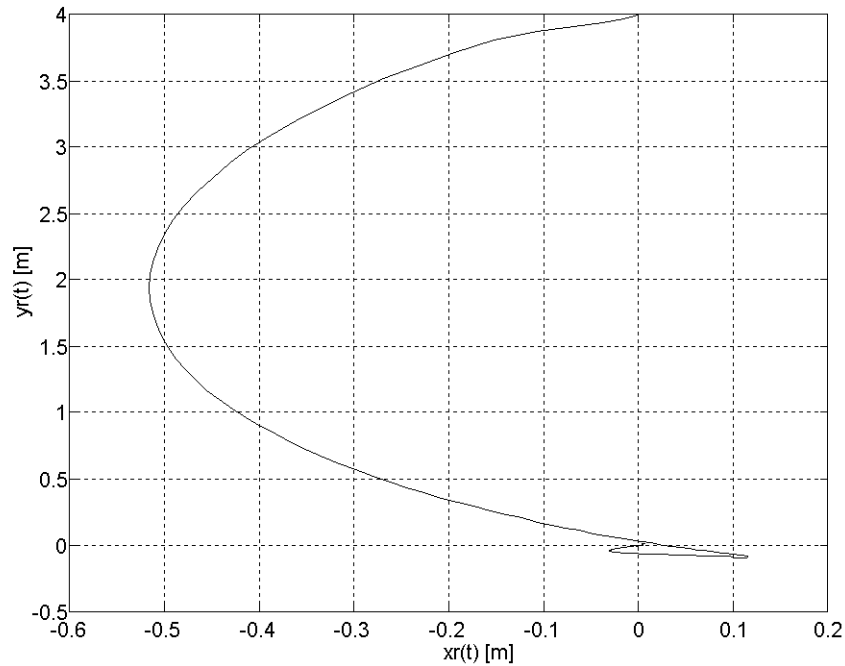


Fig. III-3 - Simulação da trajetória realizada pelo robô com acesso completo aos estados.

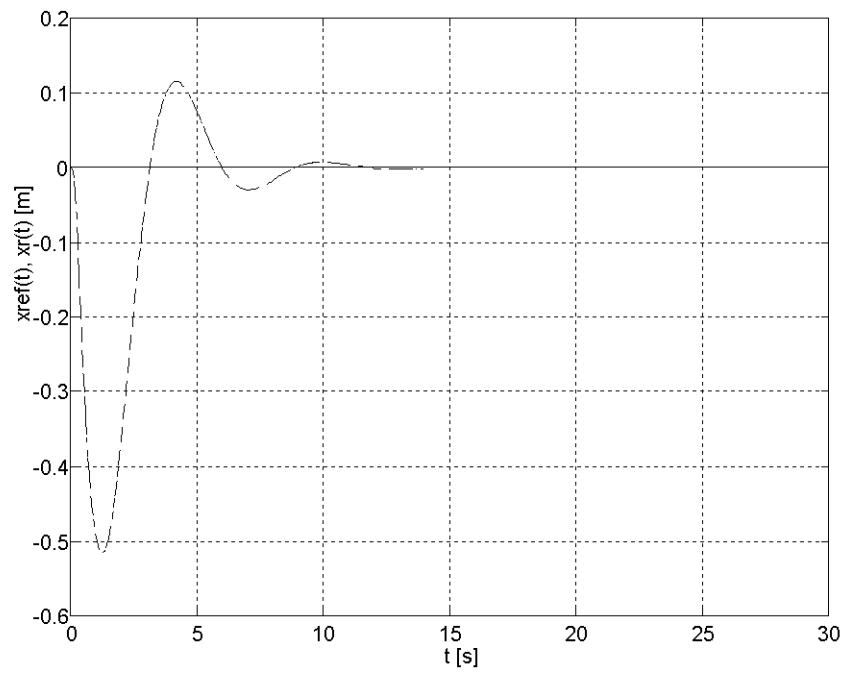


Fig. III-4 - Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover.

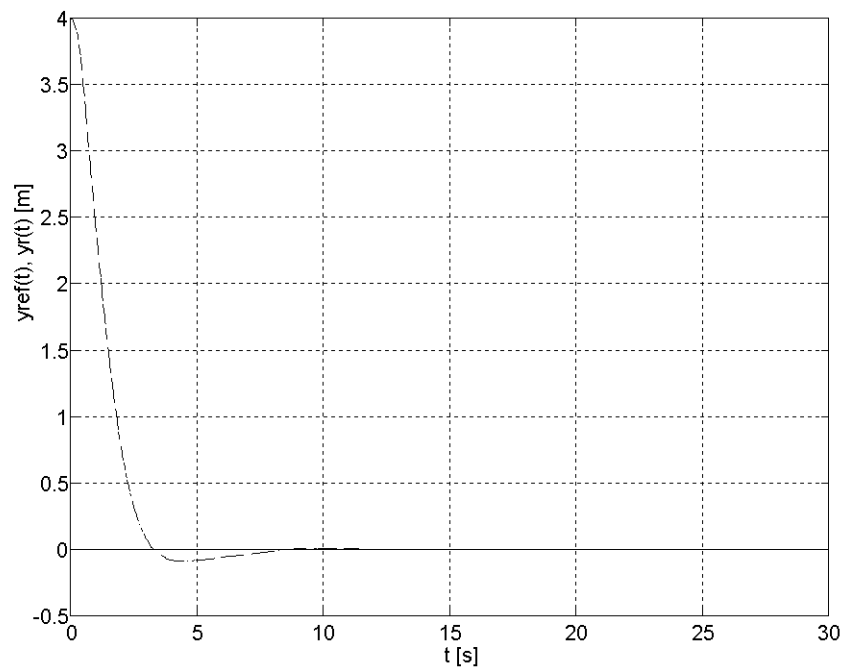


Fig. III-5 - Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover.

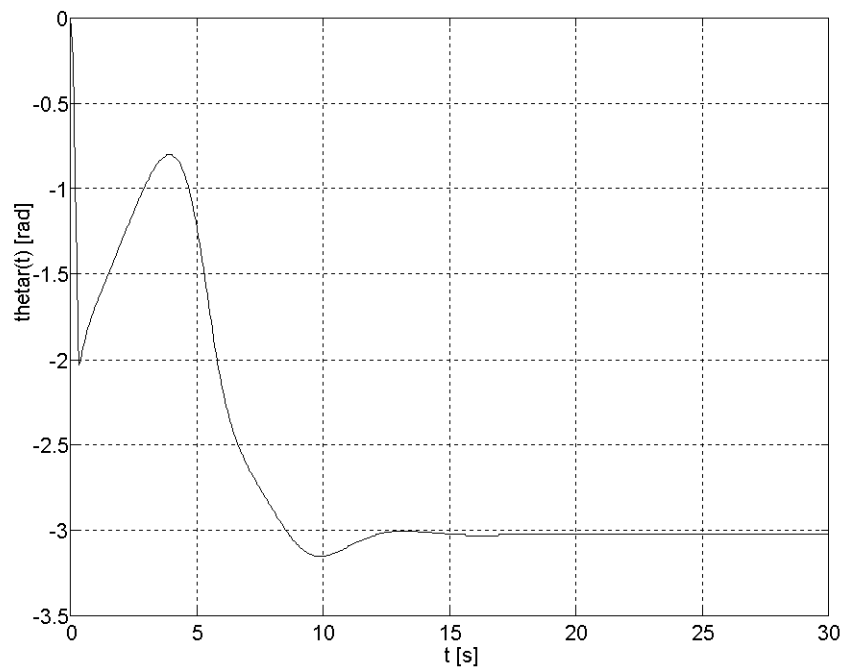


Fig. III-6 - Resposta de orientação para o robô IEEE-Rover.

Esta mesma lei de controle, também foi simulada para uma tarefa de rastreamento de uma trajetória, como mostrado nas Figs. III-7 a III-10. Propositalmente, como condição inicial foi escolhido um ponto não pertencente a trajetória de referência. Apesar disto, o robô converge para trajetória de referência e passa a rastrea-la. Na Fig. III-11 são mostrados os torques aplicados nas rodas.

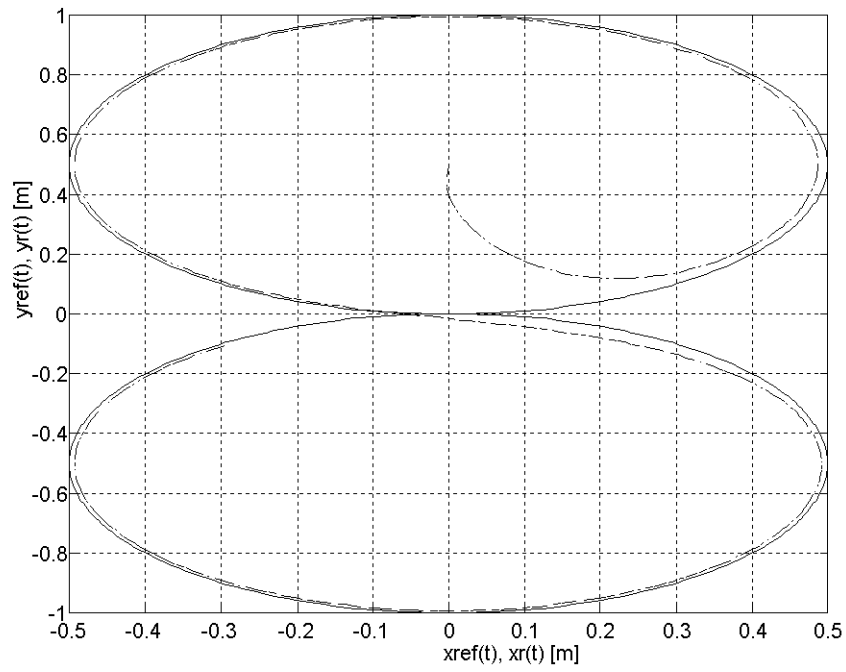


Fig. III-7 - Trajetória de referência (sólida) e trajetória realizada pelo robô (tracejada) com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

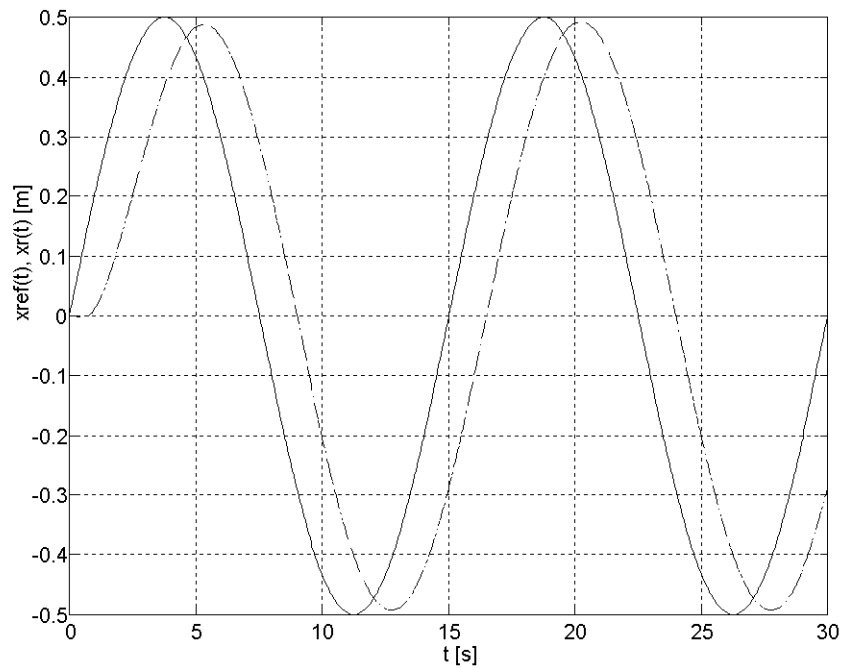


Fig. III-8 - Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

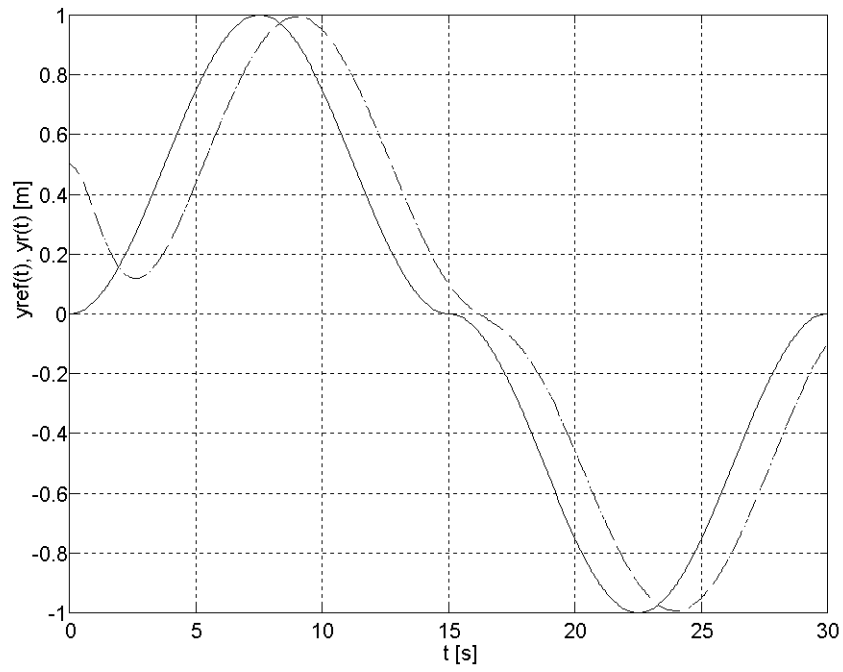


Fig. III-9 - Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

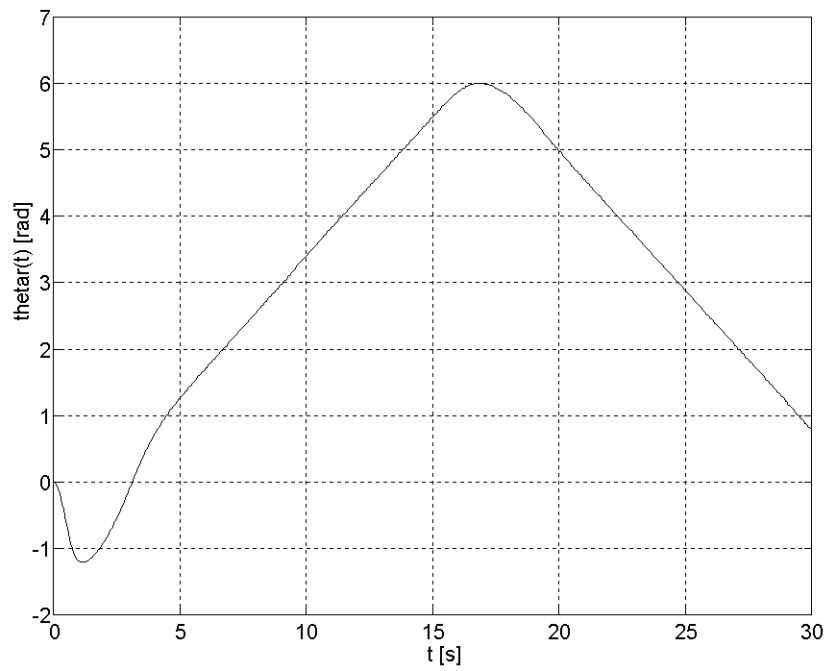


Fig. III-10 - Resposta de orientação para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

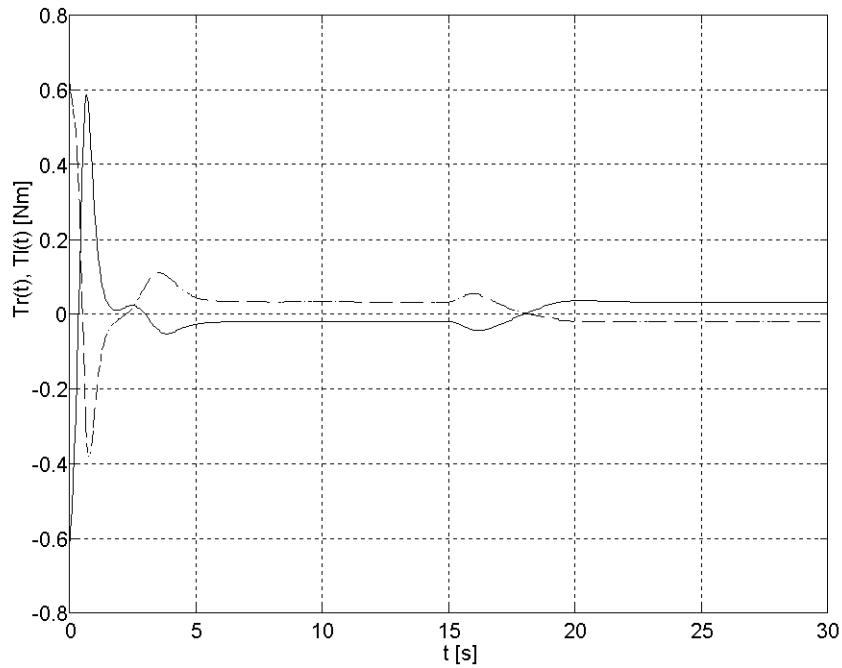


Fig. III-11 - Torque aplicado na roda direita (sólida) e torque aplicado na roda esquerda (tracejada) com $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

Através dos gráficos apresentados acima, pode-se perceber que existe um atraso significativo entre a trajetória de referência e a trajetória executada pelo robô. Na realidade, este atraso é determinado pelos parâmetros do modelo de referência. A princípio, pode-se reduzir este atraso utilizando-se um modelo de referência mais rápido. Porém, com isto, serão necessários torques maiores. A simulação anterior foi repetida utilizando-se modelos de referência com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$. Os resultados obtidos são mostrados nas Figs. III-12 a III-16. Pode-se observar, que uma redução do atraso por um fator de 2 ou 3, gerou um aumento do torque máximo necessário por um fator de aproximadamente 10. Percebe-se portanto, que a relação entre a velocidade de resposta do sistema e o torque necessário, não é linear. Em simulação, isto não representa problema, no entanto, na implementação em tempo real, os atuadores deverão ter capacidade de fornecer o torque requerido. Caso contrário, entrarão em saturação. Nestas condições, o cancelamento dos termos não lineares através da

realimentação, não será exato e portanto, o sistema terá o seu desempenho alterado, podendo até mesmo tornar-se instável.

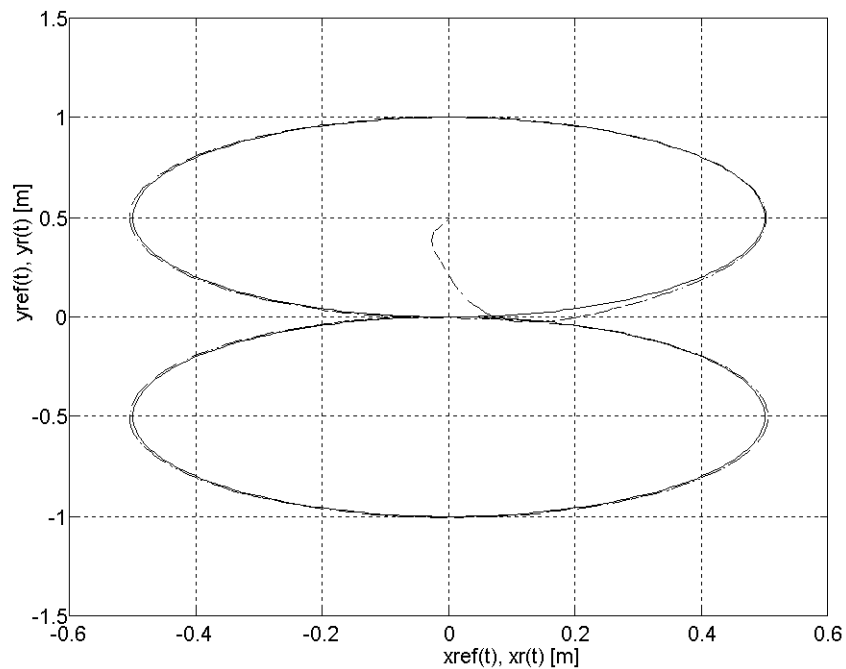


Fig. III-12 - Trajetória de referência (sólida) e trajetória realizada pelo robô (tracejada) com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$.

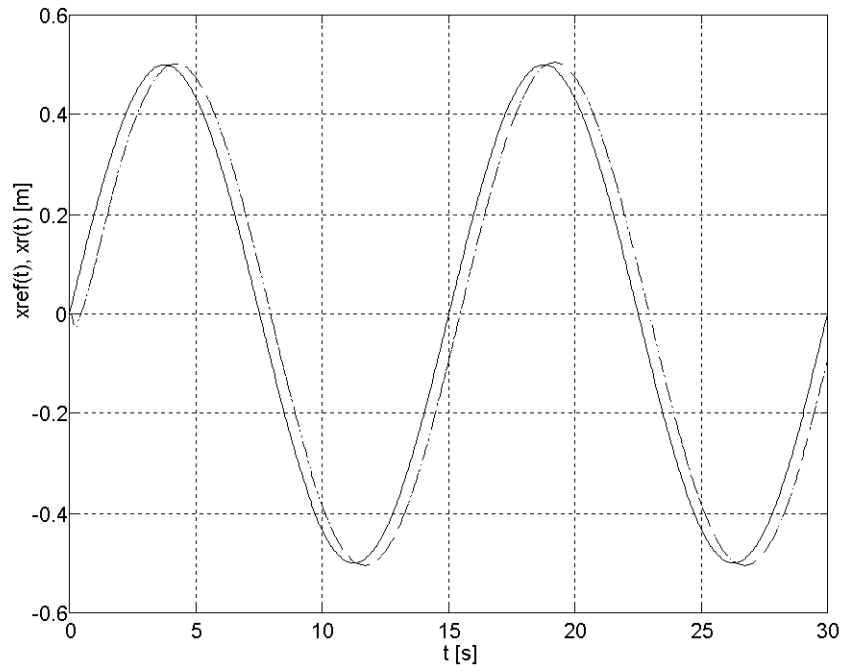


Fig. III-13 - Referência (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$.

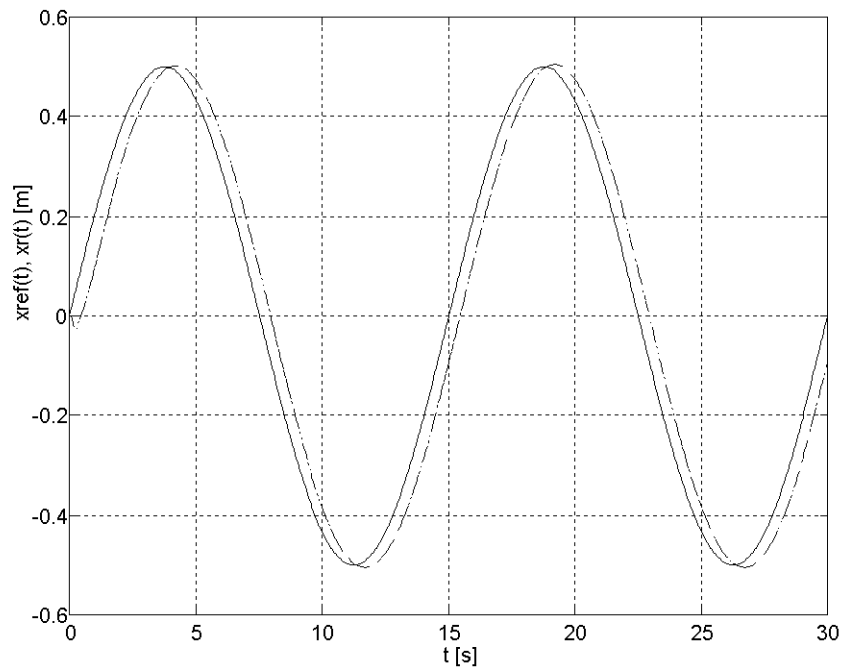


Fig. III-14 - Referência (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$.

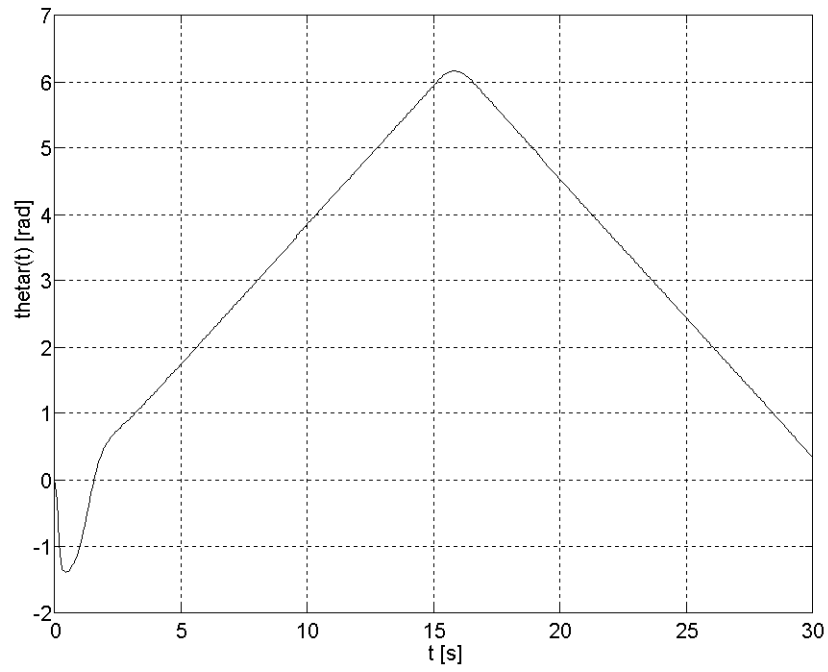


Fig. III-15 - Resposta de orientação para o robô IEEE-Rover com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$.

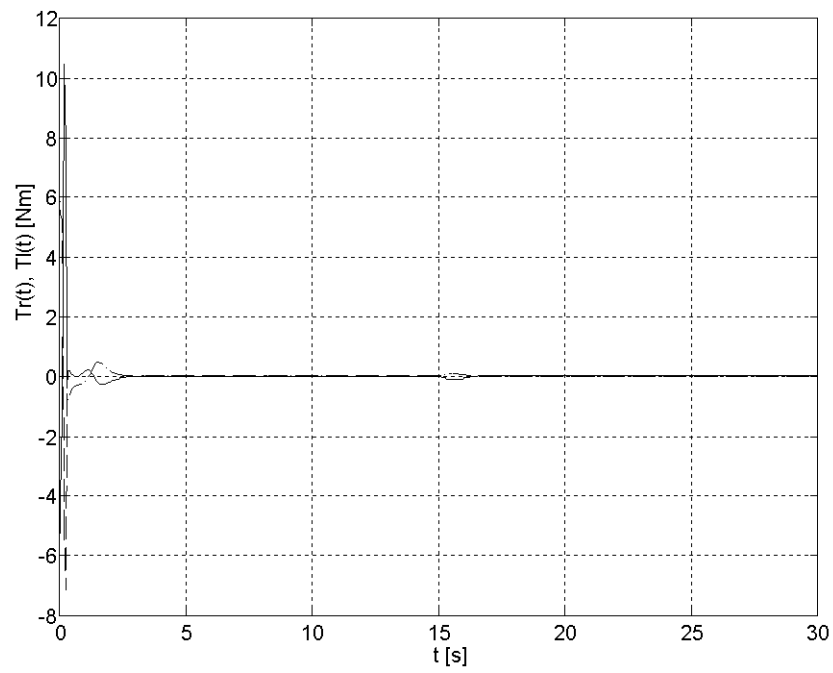


Fig. III-16 - Torque aplicado na roda direita (sólida) e torque aplicado na roda esquerda (tracejada) com $\alpha_1 = \alpha_2 = 10$ e $\alpha_3 = \alpha_4 = 4.43$.

III.9. Resultados de Tempo Real

A lei de controle (III.24), juntamente com os modelos de referência (III.33) e (III.34) foi implementada em tempo real no robô móvel Twil (vide capítulo VII). As trajetórias de referência utilizadas, são semelhantes às mesmas utilizadas nas simulações da seção anterior. Os parâmetros dos modelos de referência são $\alpha_1 = \alpha_2 = 1$ e $\alpha_3 = \alpha_4 = 1.4$.

A trajetória espacial obtida em tempo real, com dados dos *encoders* e em simulação para estabilizar o robô no ponto $x_r=0$, $y_r=0$ e $\theta=0$, a partir da condição inicial $x_r(0)=0$ m, $y_r(0)=3$ m e $\theta(0)=0$, é mostrada na Fig. III-17. As Figs. III-18 a III-20 mostram as variáveis de posição e orientação em função do tempo.

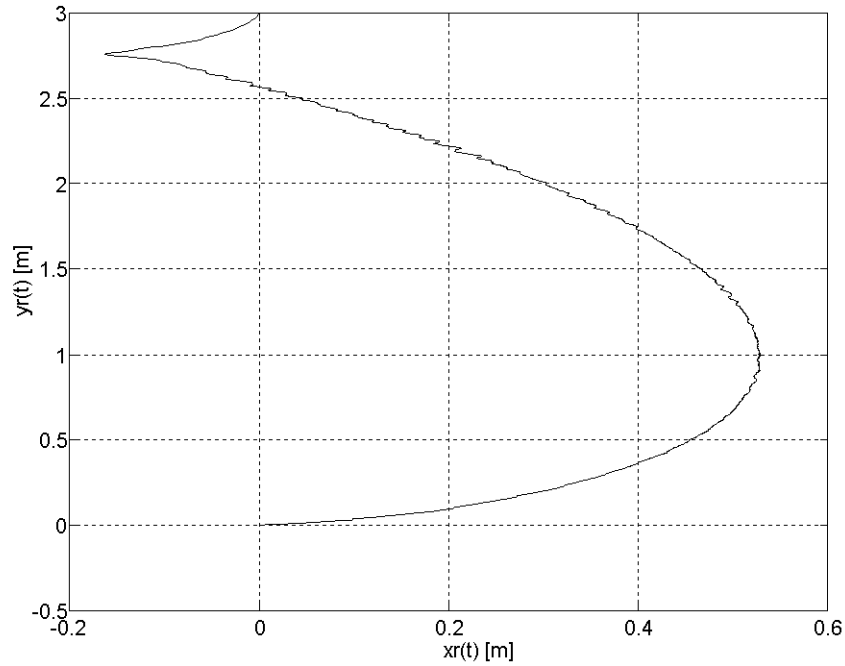


Fig. III-17 - Trajetória espacial em tempo real para o robô Twil em tarefa de parking.

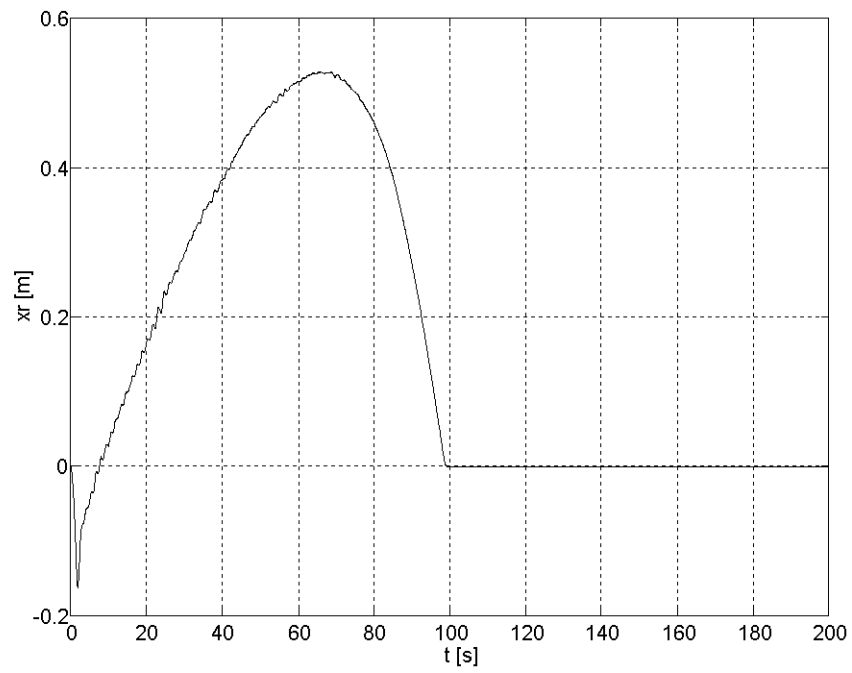


Fig. III-18 - Resposta em tempo real de $x_r(t)$ para o robô Twil em tarefa de parking.

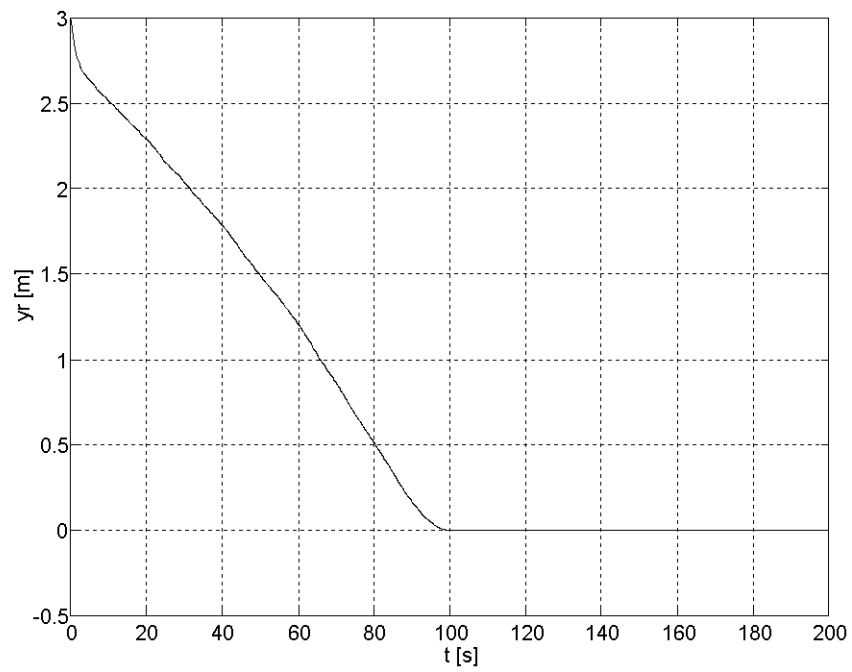


Fig. III-19 - Resposta em tempo real de $y_r(t)$ para o robô Twil em tarefa de parking.

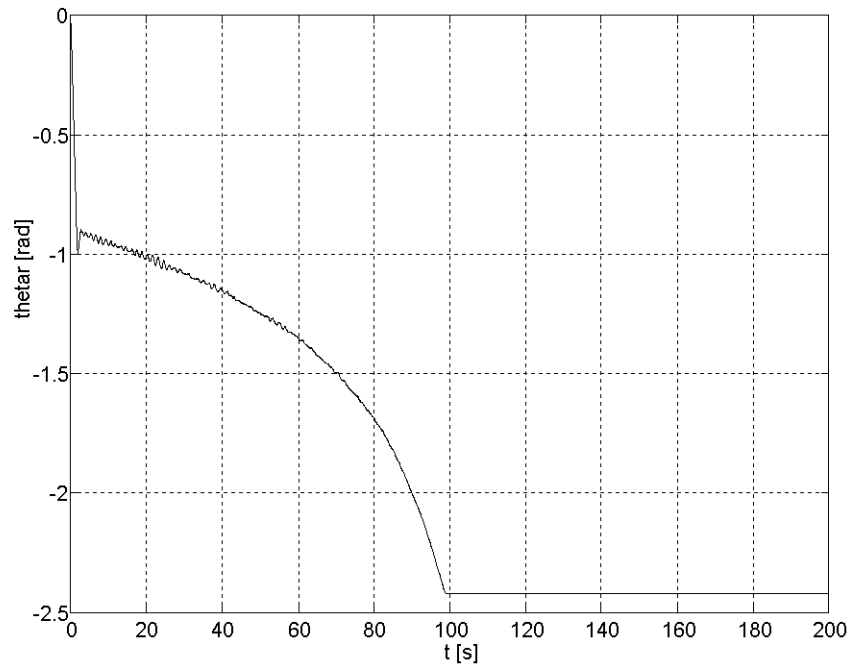


Fig. III-20 - Resposta em tempo real de orientação para o robô Twil em tarefa de parking.

De forma similar a seção anterior, uma tarefa de rastreamento de uma trajetória também foi executada em tempo real, como mostrado nas Figs. III-21 a III-24.

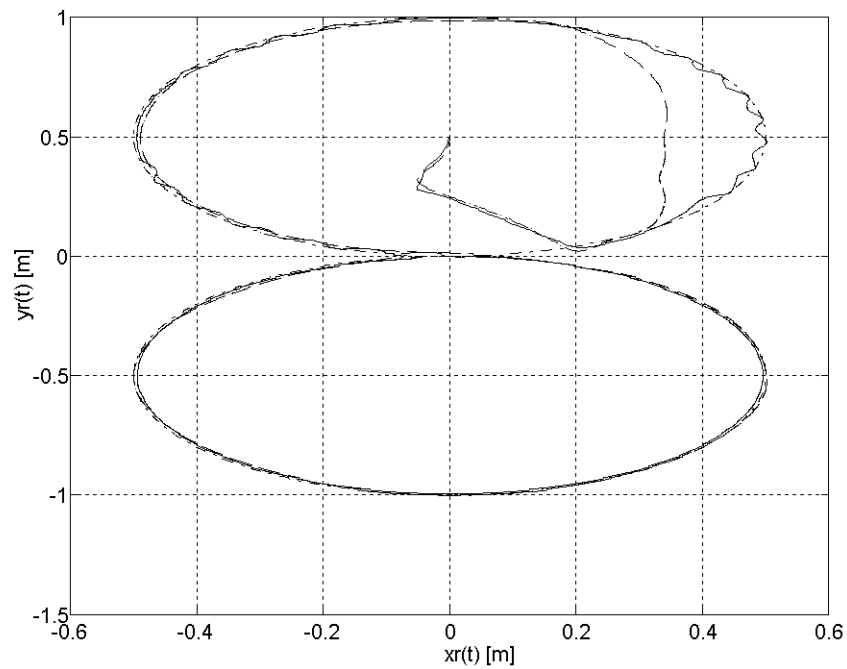


Fig. III-21 - Trajetória espacial de referência (traço ponto), trajetória de tempo real (sólida) e trajetória simulada (tracejada) em tarefa de rastreamento.

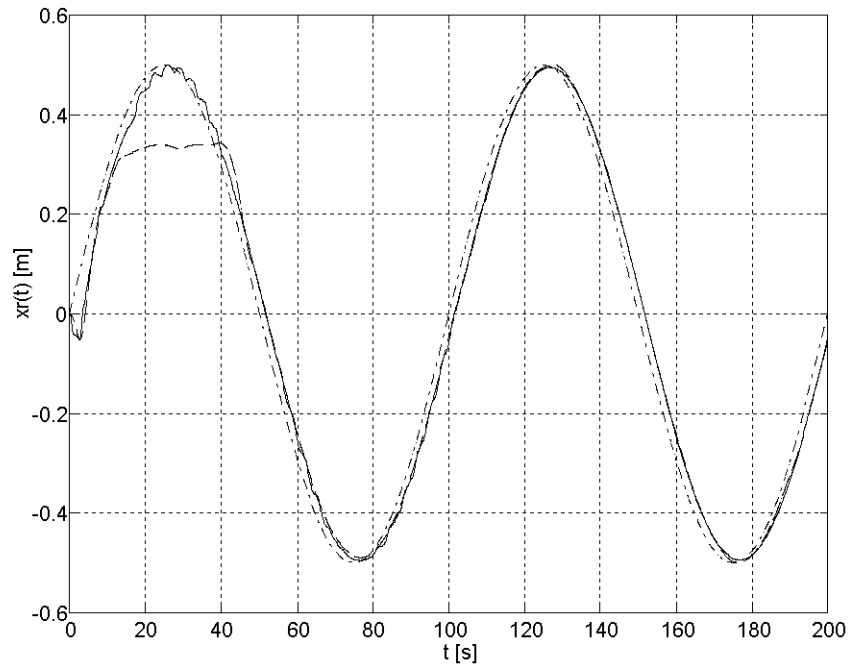


Fig. III-22 - Referência (traço ponto), resposta de tempo real (sólida) e resposta simulada (tracejada) de $x_r(t)$ para o robô Twil em tarefa de rastreamento.

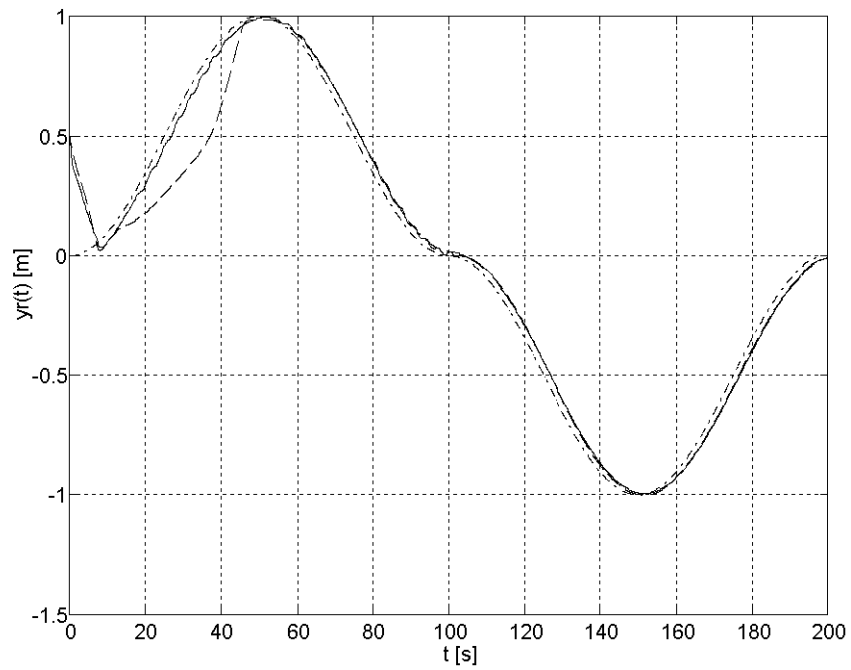


Fig. III-23 - Referência (traço ponto), resposta de tempo real (sólida) e resposta simulada (tracejada) de $y_r(t)$ para o robô Twil em tarefa de rastreamento.

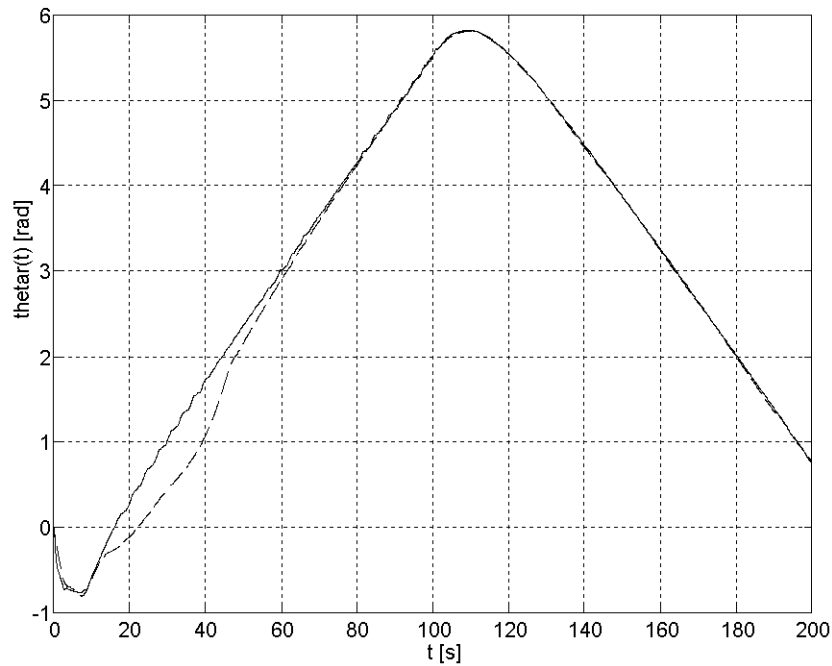


Fig. III-24 - Resposta de tempo real (sólida) e resposta simulada (tracejada) de orientação para o robô Twil em tarefa de rastreamento.

III.10. Conclusão

Utilizando-se o modelo dinâmico de postura, é possível se obter linearização das variáveis de posição e orientação apenas para robôs omnidirecionais, ou seja, para robôs pertencentes a classe (3,0). Para robôs pertencentes as classes (2,0) e (2,1), obtém-se a linearização apenas de duas variáveis de postura. Por conveniência, são escolhidas as variáveis de posição. Para as classes (1,1) e (1,2), é possível linearizar-se apenas uma das variáveis de postura, o que não tem grande utilidade para o projeto de um controlador para este tipo de robô, pois será possível controlar uma das variáveis de posição **ou** a orientação do robô.

Do exposto acima, torna-se óbvio que o número de variáveis de postura que podem ser linearizadas por realimentação estática de estados é igual ao grau de mobilidade δ_m do robô. Por outro lado, em [14] é afirmado que a dimensão do maior subsistema linearizável do

modelo dinâmico de postura é $2(\delta_m + \delta_s)$. Uma conclusão imediata, porém errônea, deste resultado é que para qualquer classe de robô seria possível obter-se a linearização de todas as variáveis de postura, pois em qualquer caso, $\dim(\xi_r) < 2(\delta_m + \delta_s)$. No entanto, como demonstrado nas seções anteriores, não se tem liberdade para escolher quais variáveis de estado serão linearizadas e portanto, nem sempre será possível fazer com que as variáveis de estado correspondentes às variáveis de postura sejam linearizadas.

É interessante notar também, que resultados semelhantes aos apresentados aqui podem ser obtidos se for utilizado apenas o modelo cinemático de postura. Neste caso porém, o sistema será de ordem relativa igual a 1. No entanto, as mesmas restrições quanto à escolha do ponto de referência e quanto as variáveis a serem linearizadas são exatamente as mesmas obtidas quando utilizando o modelo dinâmico de postura. Isto está relacionado ao fato de que as dificuldades no controle de robôs móveis advém da cinemática e não da dinâmica do mecanismo.

Adicionalmente, convém notar que no modelo cinemático de postura as variáveis de entrada são homogêneas a velocidades ou posição. Na realidade, fisicamente, as entradas do robô são os torques aplicados para rotação e orientação das rodas. Isto no entanto, não significa que o modelo cinemático de postura não pode ser utilizado para o desenvolvimento de estratégias de controle. Para a sua utilização, basta que existam malhas de controle internas que garantam que as velocidades ou posições comandadas sejam efetivamente aplicadas nas rodas. Isto pode ser feito, por exemplo, utilizando-se controladores PID convencionais. Esta estratégia de controle é interessante quando não se dispõe dos parâmetros do modelo dinâmico do robô, mas apenas dos parâmetros do modelo cinemático, que são mais facilmente obtidos.

Capítulo IV

Controle por Transformação Descontínua

Sabe-se que transformações de coordenadas adequadas podem ser bastante úteis para o projeto de controladores para sistemas não-holonômicos. A literatura mostra que a transformação para coordenadas polares permite o controle de robôs móveis com acionamento diferencial. No entanto, não existe um método para obtenção de tais transformações. Neste capítulo, propõe-se transformações de variáveis adicionais que permitem a obtenção de leis de controle para todas as classes de robôs móveis com rodas. As leis de controle obtidas garantem a convergência assintótica do robô para uma determinada posição e orientação. São apresentados resultados de simulação para um robô representativo de cada classe e resultados experimentais para um robô da classe (2,0).

IV.1. Transformação de Coordenadas

Nesta seção o modelo descrito pela expressão (II.35) será convertido para coordenadas polares. Conseqüentemente, a posição do robô não mais será representado em função das coordenadas cartesianas (x_c, y_c) , mas sim através das coordenadas polares (e, ϕ) . Observando-se a Fig. IV-1 é facilmente verificado que

$$e = \sqrt{x_c^2 + y_c^2} \quad (\text{VI.1})$$

$$\phi = \text{atan2}(y_c, x_c) \quad (\text{VI.2})$$

$$x_c = e \cos \phi \quad (\text{VI.3})$$

$$y_c = e \sin \phi \quad (\text{VI.4})$$

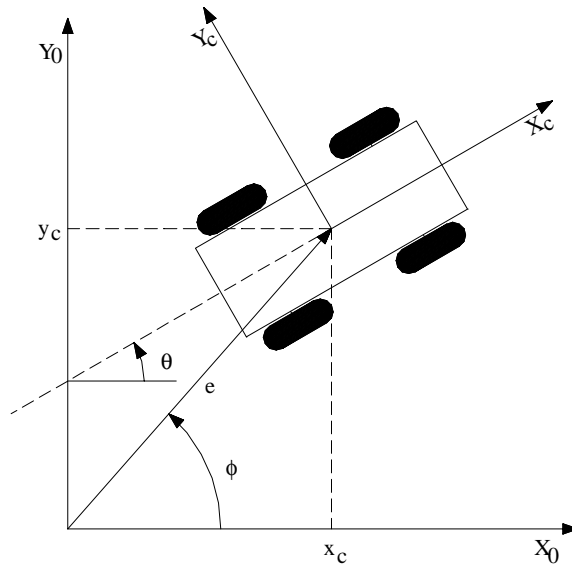


Fig. IV-1 - Definição das variáveis em coordenadas polares.

derivando-se (IV.1) e (VI.2) e substituindo-se (IV.3) e (IV.4) obtém-se

$$\dot{x} = \dot{x}_c \cos \phi + \dot{y}_c \sin \phi \quad (\text{IV.5})$$

$$\dot{\phi} = \frac{1}{e} (-\dot{x}_c \sin \phi + \dot{y}_c \cos \phi) \quad (\text{IV.6})$$

ou ainda, escrevendo-se na forma matricial

$$\begin{bmatrix} \dot{x} \\ \dot{\phi} \end{bmatrix} = \Gamma(e, \phi) \begin{bmatrix} \dot{x}_c \\ \dot{y}_c \end{bmatrix} \quad (\text{IV.7})$$

$$\text{onde } \Gamma(e, \phi) = \begin{bmatrix} \cos \phi & \sin \phi \\ -\frac{\sin \phi}{e} & \frac{\cos \phi}{e} \end{bmatrix}.$$

Definindo-se

$$z = \begin{bmatrix} e \\ \phi \\ \theta \\ \beta_c \end{bmatrix} \quad (\text{IV.8})$$

pode-se de (II.35) e (IV.7) escrever o modelo em coordenadas polares

$$\mathbf{z} = D(z)\mathbf{u} \quad (IV.9)$$

$$\text{com } D(z) = \begin{cases} \Gamma(e, \phi) \mathbf{0} \mathbf{R}_c \Sigma, N_c = 0 \\ \begin{bmatrix} \Gamma(e, \phi) \mathbf{0} \mathbf{R}_c \Sigma(\beta_c) & 0 \\ 0 & I \end{bmatrix}, N_c \geq 0 \end{cases}$$

Através da escolha conveniente da origem e da orientação do sistema de coordenadas $\{X_c, Y_c\}$ e definindo-se $\alpha = \theta - \phi$, a expressão (IV.9) pode ser escrita explicitamente para cada uma das cinco classes de robôs, conforme a Tab. IV-1.

Tab. IV - 1 - Modelos em coordenadas polares

Classe	Modelo
(3,0)	$\begin{bmatrix} \mathbf{z} \\ \phi \\ \alpha \end{bmatrix} = \begin{bmatrix} \frac{\cos \alpha}{e} & -\frac{\sin \alpha}{e} & 0 \\ \frac{\sin \alpha}{e} & \frac{\cos \alpha}{e} & 0 \\ -\frac{\sin \alpha}{e} & -\frac{\cos \alpha}{e} & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{bmatrix} \quad (IV.10)$
(2,0)	$\begin{bmatrix} \mathbf{z} \\ \phi \\ \alpha \end{bmatrix} = \begin{bmatrix} \frac{\cos \alpha}{e} & 0 \\ \frac{\sin \alpha}{e} & 0 \\ -\frac{\sin \alpha}{e} & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} \quad (IV.11)$
(2,1)	$\begin{bmatrix} \mathbf{z} \\ \phi \\ \alpha \\ \beta_c \end{bmatrix} = \begin{bmatrix} \frac{\cos(\alpha + \beta_c)}{e} & 0 & 0 \\ \frac{\sin(\alpha + \beta_c)}{e} & 0 & 0 \\ -\frac{\sin(\alpha + \beta_c)}{e} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta_1 \\ \eta_2 \\ \zeta \end{bmatrix} \quad (IV.12)$
(1,1)	$\begin{bmatrix} \mathbf{z} \\ \phi \\ \alpha \\ \beta_c \end{bmatrix} = \begin{bmatrix} \frac{d \cos \alpha \sin \beta_c}{e} & 0 \\ \frac{d \sin \alpha \sin \beta_c}{e} & 0 \\ \cos \beta_c - \frac{d \sin \alpha \sin \beta_c}{e} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (IV.13)$

(1,2)	$\begin{bmatrix} e \\ \phi \\ \alpha \\ \beta_{c1} \\ \beta_{c2} \end{bmatrix} = \begin{bmatrix} D_{11} & 0 & 0 \\ D_{21} & 0 & 0 \\ D_{31} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \eta \\ \zeta_1 \\ \zeta_2 \end{bmatrix} \quad (IV.14)$ <p>com</p> $D_{11} = L \left(\text{sen} \beta_{c1} \cos(\alpha + \beta_{c2}) + \text{sen} \beta_{c2} \cos(\alpha + \beta_{c1}) \right)$ $D_{21} = \frac{L \left(\text{sen} \beta_{c1} \text{sen}(\alpha + \beta_{c2}) + \text{sen} \beta_{c2} \text{sen}(\alpha + \beta_{c1}) \right)}{e}$ $D_{31} = \text{sen}(\beta_{c1} - \beta_{c2}) - \frac{L \left(\text{sen} \beta_{c1} \text{sen}(\alpha + \beta_{c2}) + \text{sen} \beta_{c2} \text{sen}(\alpha + \beta_{c1}) \right)}{e}$
-------	--

IV.2. Leis de Controle

As leis de controle propostas neste artigo baseiam-se em uma análise do tipo de Lyapunov e Barbalat [55]. Embora esta técnica possa ser aplicada para todas as classes de robôs móveis consideradas, são necessárias pequenas adaptações para cada uma das classes [34][35]. Sem perda de generalidade, é assumido que deseja-se a convergência de e , ϕ e α para zero.

IV.2.1. Classe (3,0)

Considere-se a seguinte candidata à função de Lyapunov:

$$V = \frac{1}{2} \lambda e^2 + \frac{1}{2} (\alpha^2 + h \phi^2) \quad (IV.15)$$

onde λ e h são constantes positivas.

Derivando-se (IV.15) em relação ao tempo e substituindo-se \dot{e} , $\dot{\phi}$, e $\dot{\alpha}$ de (IV.10) tem-se

$$\mathbb{V} = \lambda e \cos \alpha \eta_1 + h\phi \frac{\cos \alpha}{e} \eta_2 + \alpha \left(\eta_3 - \left(\lambda e \frac{\sin \alpha}{\alpha} + \frac{\cos \alpha}{e} \right) \eta_2 - \left(\frac{\sin \alpha}{e} - h\phi \frac{\sin \alpha}{\alpha} \right) \eta_1 \right) \quad (\text{IV.16})$$

que pode ser escrita na forma

$$\mathbb{V} = \mathbb{V}_1 + \mathbb{V}_2 \quad (\text{IV.17})$$

sendo

$$\mathbb{V}_1 = \lambda e \cos \alpha \eta_1 + h\phi \frac{\cos \alpha}{e} \eta_2 \quad (\text{IV.18})$$

e

$$\mathbb{V}_2 = \alpha \left(\eta_3 - \left(\lambda e \frac{\sin \alpha}{\alpha} + \frac{\cos \alpha}{e} \right) \eta_2 - \frac{\sin \alpha}{e\alpha} (\alpha - h\phi) \eta_1 \right) \quad (\text{IV.19})$$

Este particionamento é feito de forma a deixar em \mathbb{V}_1 os termos que não são identicamente nulos quando α é nulo, ou seja, os termos que não são função explícita de α ou $\sin \alpha$ e em \mathbb{V}_2 os demais termos.

Pode-se verificar que \mathbb{V}_1 pode ser tornado não positivo fazendo-se

$$\eta_1 = -\gamma_1 e \cos \alpha \quad (\text{IV.20})$$

e

$$\eta_2 = -\gamma_2 \phi e \cos \alpha \quad (\text{IV.21})$$

onde γ_1 e γ_2 são constantes positivas. Com isto tem-se que

$$\mathbb{V}_1 = -\lambda \gamma_1 e^2 \cos^2 \alpha - h\phi^2 \cos^2 \alpha \leq 0 \quad (\text{IV.22})$$

e

$$\mathbb{V}_2 = \alpha \left(\eta_3 + \gamma_2 \phi \cos \alpha \left(\lambda e^2 \frac{\sin \alpha}{\alpha} + \cos \alpha \right) + \gamma_1 \cos \alpha \frac{\sin \alpha}{\alpha} (\alpha - h\phi) \right) \quad (\text{IV.23})$$

logo, fazendo-se

$$\eta_3 = -\gamma_3 \alpha - \gamma_2 \phi \cos \alpha \left(\lambda e^2 \frac{\sin \alpha}{\alpha} + \cos \alpha \right) - \gamma_1 \cos \alpha \frac{\sin \alpha}{\alpha} (\alpha - h\phi) \quad (\text{IV.24})$$

sendo γ_3 uma constante positiva, tem-se

$$\dot{V}_2 = -\gamma_3 \alpha^2 \leq 0 \quad (\text{IV.25})$$

e portanto,

$$\dot{V} = -\lambda \gamma_1 e^2 \cos^2 \alpha - h \gamma_2 \phi^2 \cos^2 \alpha - \gamma_3 \alpha^2 \leq 0 \quad (\text{IV.26})$$

que juntamente com o fato de que V é contínua, garante a estabilidade do sistema em malha fechada. Por outro lado, considerando-se que: a) V é limitada inferiormente; b) V é não crescente, de (IV.26); c) \dot{V} é uniformemente contínua, pois \dot{V} é limitada, tem-se pelo lema de Barbalat que \dot{V} tende a zero e portanto, pode-se verificar pela expressão (IV.26) que e , ϕ e α também tendem a zero.

IV.2.2. Classe (2,0)

Agindo-se de forma semelhante ao caso anterior, a candidata a função de Lyapunov é particionada em

$$\dot{V}_1 = \lambda e \cos \alpha \eta_1 \quad (\text{IV.27})$$

e

$$\dot{V}_2 = \alpha \left(\eta_2 - \frac{\sin \alpha}{e \alpha} (\alpha - h \phi) \eta_1 \right) \quad (\text{IV.28})$$

de onde obtém-se

$$\eta_1 = -\gamma_1 e \cos \alpha \quad (\text{IV.29})$$

e

$$\eta_2 = -\gamma_2 \alpha - \gamma_1 \cos \alpha \frac{\sin \alpha}{\alpha} (\alpha - h \phi) \quad (\text{IV.30})$$

resultando

$$\dot{V} = -\lambda \gamma_1 e^2 \cos^2 \alpha - \gamma_2 \alpha^2 \leq 0 \quad (\text{IV.31})$$

Portanto, através dos mesmos argumentos utilizados no caso anterior, pode-se concluir que e e α tendem a zero. A convergência de ϕ para zero pode ser provada observando-se as equações do sistema em malha fechada:

$$\begin{cases} \dot{e} = -\gamma_1 e \cos^2 \alpha \\ \dot{\phi} = -\gamma_1 \sin \alpha \cos \alpha \\ \dot{\alpha} = -\gamma_2 \alpha + \gamma_1 h \phi \cos \alpha \frac{\sin \alpha}{\alpha} \end{cases} \quad (IV.32)$$

Considerando-se a convergência para zero de e e α , conclui-se de (IV.32) que \dot{e} e $\dot{\phi}$ também tendem a zero. A convergência de $\dot{\phi}$ para zero, por sua vez, faz com que α convirja para algum valor constante dado por $\gamma_1 h \phi^*$. Por outro lado, a continuidade uniforme de $\dot{\alpha}$ juntamente com a convergência para zero de α , garante, pelo lema de Barbalat, que α tende a zero. Portanto, ϕ^* tem, necessariamente, que ser zero.

As leis de controle para as demais classes, são obtidas a partir das leis de controle das classes (3,0) e (2,0) através da utilização do seguinte teorema:

Teorema IV.1. Equivalência de Modelos Cinemáticos

Considere as cinco classes de modelos cinemáticos de robôs móveis obtidas a partir de (II.34).

Seja um modelo pertencente à uma classe com grau de manobrabilidade igual a δ_M^* .

É possível obter-se um modelo equivalente pertencente a qualquer outra classe com o mesmo grau de manobrabilidade δ_M^* .

■

Prova do Teorema IV.1.

Modelos pertencentes às classes (3,0) e (2,1) podem ser convertidos entre si utilizando-se as transformações de variáveis (IV-34)-(IV-35).

Modelos pertencentes às classes (3,0) e (1,2) podem ser convertidos entre si utilizando-se as transformações de variáveis (IV-53)-(IV-35). Portanto, em duas etapas pode-se fazer a transformação entre as classes (2,1) e (1,2).

Modelos pertencentes às classes (2,0) e (1,1) podem ser convertidos entre si utilizando-se as transformações de variáveis (IV-45)-(IV-46).

■

IV.2.3. Classe (2,1)

Por conveniência, para esta classe de robôs, será considerado o modelo na forma (II.34), ou seja, a ordem do modelo não é aumentada definindo-se $\zeta = \beta_c$. Considera-se, portanto, η_1 , η_2 e β_c como sendo as entradas de controle. Com isto, a expressão (IV.12) torna-se

$$\begin{cases} \dot{\alpha} = \cos(\alpha + \beta_c) \eta_1 \\ \dot{\phi} = \frac{\sin(\alpha + \beta_c)}{e} \eta_1 \\ \dot{\alpha} = -\frac{\sin(\alpha + \beta_c)}{e} \eta_1 + \eta_2 \end{cases} \quad (IV.33)$$

Definindo-se

$$u_1 = \eta_1 \cos \beta_c \quad (IV.34)$$

$$u_2 = \eta_1 \sin \beta_c \quad (IV.35)$$

$$u_3 = \eta_2 \quad (IV.36)$$

O modelo (IV.33) assume a forma

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\phi} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \frac{\cos \alpha}{e} & -\frac{\sin \alpha}{e} & 0 \\ \frac{\sin \alpha}{e} & \frac{\cos \alpha}{e} & 0 \\ -\frac{\sin \alpha}{e} & -\frac{\cos \alpha}{e} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (IV.37)$$

que é exatamente a mesma forma obtida para o modelo de um robô pertencente a classe (3,0).

Logo, através do mesmo procedimento utilizado na seção IV.2.1, obtém-se

$$u_1 = -\gamma_1 e \cos \alpha \quad (\text{IV.38})$$

$$u_2 = -\gamma_2 \phi e \cos \alpha \quad (\text{IV.39})$$

$$u_3 = -\gamma_3 \alpha - \gamma_2 \phi \cos \alpha \left(\lambda e^2 \frac{\sin \alpha}{\alpha} + \cos \alpha \right) - \gamma_1 \cos \alpha \frac{\sin \alpha}{\alpha} (\alpha - h \phi) \quad (\text{IV.40})$$

Obviamente, as entradas de controle η_1 , η_2 e β_c podem ser computadas através das seguintes expressões:

$$\eta_1 = \sqrt{u_1^2 + u_2^2} \quad (\text{IV.41})$$

$$\beta_c = \text{atan2}(u_2, u_1) \quad (\text{IV.42})$$

$$\eta_2 = u_3 \quad (\text{IV.43})$$

IV.2.4. Classe (1,1)

Para esta classe de robôs móveis também é conveniente considerar o modelo escrito na forma da expressão (II.34). Após a transformação para coordenadas polares, tem-se explicitamente

$$\begin{cases} \dot{e} = d \cos \alpha \sin \beta_c \eta \\ \dot{\phi} = \frac{d \sin \alpha}{e} \sin \beta_c \eta \\ \dot{\alpha} = \left(\cos \beta_c - \frac{d \sin \alpha}{e} \sin \beta_c \right) \eta \end{cases} \quad (\text{IV.44})$$

sendo η e β_c as entradas do sistema.

Definindo-se

$$u_1 = \eta d \sin \beta_c \quad (\text{IV.45})$$

$$u_2 = \eta \cos \beta_c \quad (\text{IV.46})$$

o modelo (IV.44) assume a forma

$$\begin{bmatrix} \& \\ \phi \\ \alpha \end{bmatrix} = \begin{bmatrix} \cos \alpha & 0 \\ \frac{\text{sen} \alpha}{e} & 0 \\ -\frac{\text{sen} \alpha}{e} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (\text{IV.47})$$

que é a mesma forma do modelo para a classe (2,0). Logo, obtém-se

$$u_1 = -\gamma_1 e \cos \alpha \quad (\text{IV.48})$$

$$u_2 = -\gamma_2 \alpha - \gamma_1 \cos \alpha \frac{\text{sen} \alpha}{\alpha} (\alpha - h\phi) \quad (\text{IV.49})$$

Com as entradas η e β_c obtidas a partir de u_1 e u_2 através das expressões

$$\eta = \sqrt{\left(\frac{u_1}{d}\right)^2 + u_2^2} \quad (\text{IV.50})$$

$$\beta_c = \text{atan2}\left(\frac{u_1}{d}, u_2\right) \quad (\text{IV.51})$$

IV.2.5. Classe (1,2)

A semelhança do procedimento adotado para as classes (2,1) e (1,1), aqui também será utilizado o modelo na forma (II.34). Após a transformação para coordenadas polares resulta

$$\begin{cases} \& = L \left(\cos \alpha \text{sen}(\beta_{c1} + \beta_{c2}) - 2 \text{sen} \alpha \text{sen} \beta_{c1} \text{sen} \beta_{c2} \right) \eta \\ \phi = L \frac{\text{sen} \alpha \text{sen}(\beta_{c1} + \beta_{c2}) + 2 \cos \alpha \text{sen} \beta_{c1} \text{sen} \beta_{c2}}{e} \eta \\ \alpha = \left(\text{sen}(\beta_{c1} - \beta_{c2}) - L \frac{\text{sen} \alpha \text{sen}(\beta_{c1} + \beta_{c2}) + 2 \cos \alpha \text{sen} \beta_{c1} \text{sen} \beta_{c2}}{e} \right) \eta \end{cases} \quad (\text{IV.52})$$

Neste caso, as entradas do sistema são η e β_{c1} e β_{c2} . Definindo-se novas variáveis de entrada

$$u_1 = \eta L \text{sen}(\beta_{c1} + \beta_{c2}) \quad (\text{IV.53})$$

$$u_2 = 2\eta L \sin\beta_{c1} \sin\beta_{c2} \quad (\text{IV.54})$$

$$u_3 = \eta \sin(\beta_{c1} - \beta_{c2}) \quad (\text{IV.55})$$

tem-se que o modelo do sistema pode ser escrito na forma

$$\begin{bmatrix} \alpha \\ \phi \\ \alpha \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \frac{\sin\alpha}{e} & \frac{\cos\alpha}{e} & 0 \\ -\frac{\sin\alpha}{e} & -\frac{\cos\alpha}{e} & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (\text{IV.56})$$

que é igual ao modelo para a classe (3,0). Portanto, através de uma análise semelhante chega-se a

$$u_1 = -\gamma_1 e \cos\alpha \quad (\text{IV.57})$$

$$u_2 = -\gamma_2 e \phi \cos\alpha \quad (\text{IV.58})$$

$$u_3 = -\gamma_3 \alpha - \gamma_2 \phi \cos\alpha \left(\lambda e^2 \frac{\sin\alpha}{\alpha} + \cos\alpha \right) - \gamma_1 \cos\alpha \frac{\sin\alpha}{\alpha} (\alpha - h\phi) \quad (\text{IV.59})$$

Para obter-se η e β_{c1} e β_{c2} a partir de u_1 , u_2 e u_3 procede-se da seguinte maneira: (IV.53)

e (IV.54) podem ser reescritas na forma

$$\frac{u_1}{L} = \eta \sin(\beta_{c1} + \beta_{c2}) \quad (\text{IV.60})$$

$$\frac{u_2}{L} = 2\eta \sin\beta_{c1} \sin\beta_{c2} \quad (\text{IV.61})$$

dividindo-se (IV.61) pela diferença e pela soma de (IV.60) e (IV.55) obtém-se

$$\beta_{c1} = \text{atan2} \left(\frac{u_2}{L}, \frac{u_1}{L} - u_3 \right) \quad (\text{IV.62})$$

$$\beta_{c2} = \text{atan2} \left(\frac{u_2}{L}, \frac{u_1}{L} + u_3 \right) \quad (\text{IV.63})$$

Elevando ao quadrado (IV.60), (IV.61) e (IV.55) e somando as três expressões resulta

$$\eta = \pm \sqrt{\frac{\frac{u_1^2}{L^2} + \frac{u_2^2}{L^2} + u_3^2}{2(\sin^2 \beta_{c1} + \sin^2 \beta_{c2})}} \quad (\text{IV.64})$$

que pode ser calculado após β_{c1} e β_{c2} serem obtidos através de (IV.62)-(IV.63). Convém ressaltar que esta expressão não é válida quando $\beta_{c1}=n_1\pi$ e simultaneamente $\beta_{c2}=n_2\pi$, $n_1, n_2=0, 1, 2, \dots$. No entanto, para estas condições, o próprio modelo (IV.14) não é válido, pois $\delta_m = \dim \mathcal{N}(\mathbf{C}_1^*(\beta_c))$ será igual a 2 e não mais igual a 1 como é necessário para que o robô pertença a classe (1,2). Ou seja, nesta situação, o robô degenera para um robô pertencente à classe (2,0). Fisicamente, esta situação equivalente às duas rodas orientáveis estarem em paralelo. Nesta situação o robô só poderá ser controlado se as velocidades das rodas puderem ser comandadas independentemente, tal como ocorre na classe (2,0). O sinal correto de η pode ser determinado de (IV.53) e (IV.55) observando-se que $\sin(\beta_{c1}+\beta_{c2})$ e $\sin(\beta_{c1}-\beta_{c2})$ serão simultaneamente nulos apenas se $\beta_{c1}=n_1\pi$ e $\beta_{c2}=n_2\pi$. Além disso, tem-se que $\text{sgn}(a/b)=\text{sgn}(ab)$. Portanto, pode-se escrever

$$\text{sgn}(\eta) = \begin{cases} \text{sgn}(u_1 L \sin(\beta_{c1} + \beta_{c2})), & |u_1 L \sin(\beta_{c1} + \beta_{c2})| > |u_3 \sin(\beta_{c1} - \beta_{c2})| \\ \text{sgn}(u_3 \sin(\beta_{c1} - \beta_{c2})), & \text{caso contrário} \end{cases} \quad (\text{IV.65})$$

IV.2.6 Resultados de Simulação

As leis de controle obtidas na seção anterior, foram simuladas para um robô típico pertencente a cada classe discutida. As condições iniciais para todas as simulações foram $x_c(0)=4\text{m}$, $y_c(0)=4\text{m}$ e $\theta(0)=\pi/2$ e deseja-se estabilizar o robô na origem. As trajetórias espaciais obtidas podem ser vistas na Fig IV-2. Como seria de se esperar, observa-se que as trajetórias convergem de modo relativamente suave para a origem. As curvas de orientação são mostradas na Fig. IV-3. Para um robô da classe (1,2), a convergência da orientação pode

ser verificada nas Figs. IV-4-IV-6, onde são mostradas as curvas de $x_c(t)$, $y_c(t)$ e $\theta(t)$. Para as demais classes são obtidas curvas semelhantes.

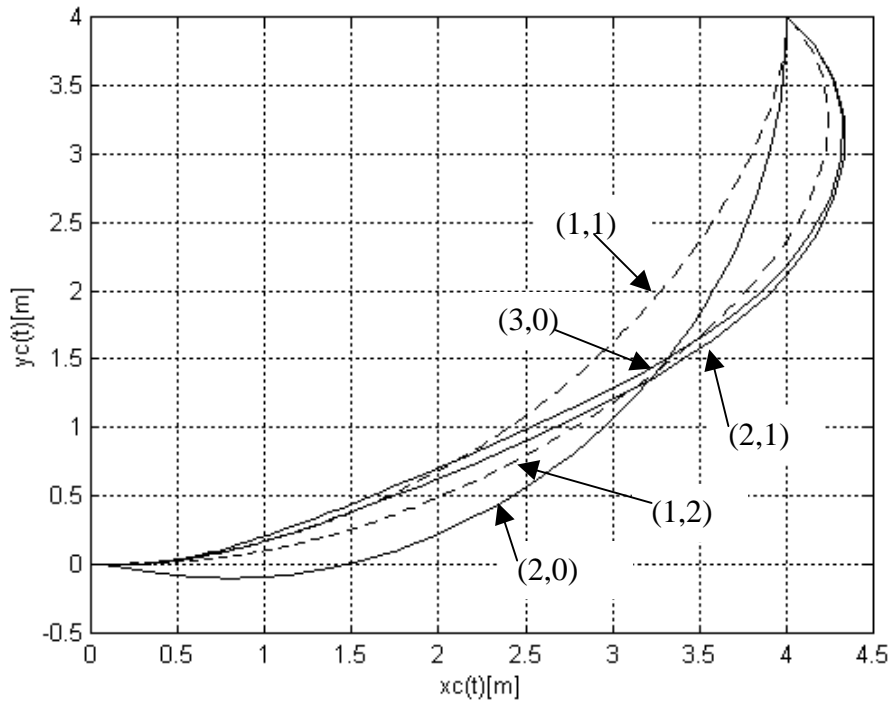


Fig. IV-2 - Trajetória espacial para um robô de cada classe.

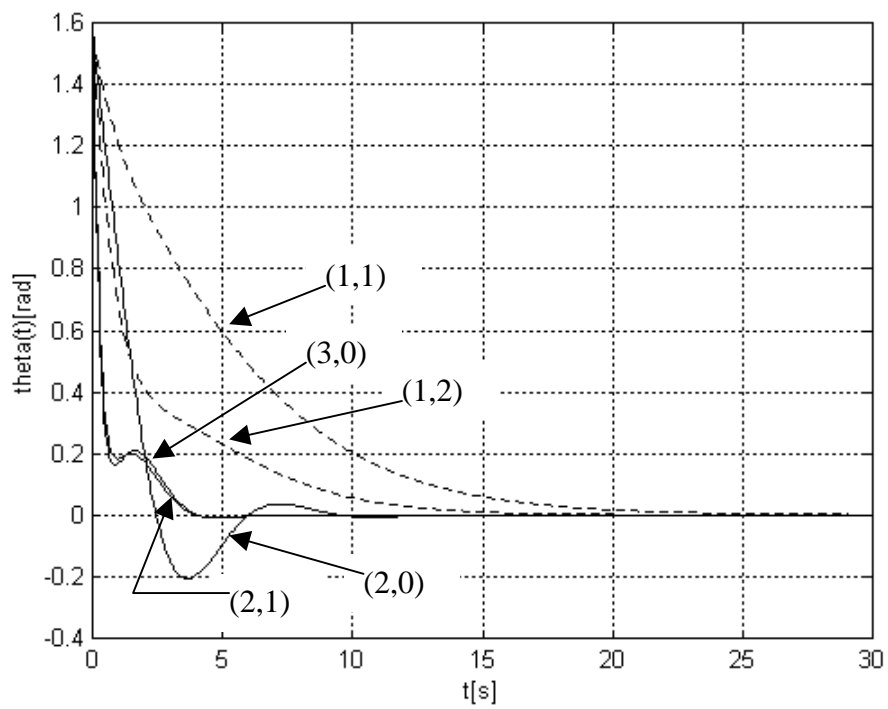


Fig. IV-3 - Resposta de $\theta(t)$ para um robô de cada classe.

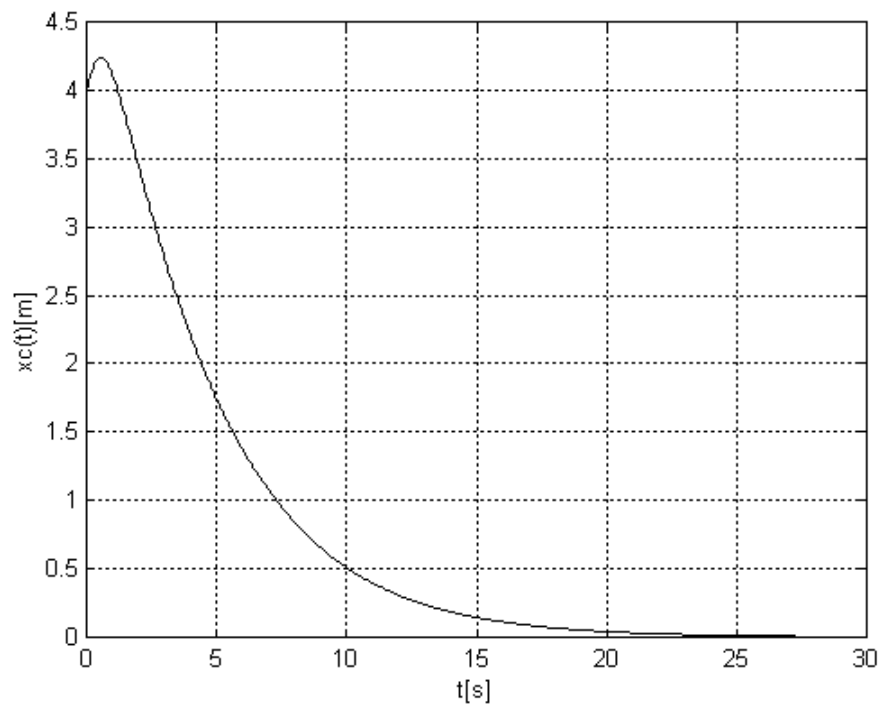


Fig. IV-4 - Resposta de $x_c(t)$ para robô classe (1,2).

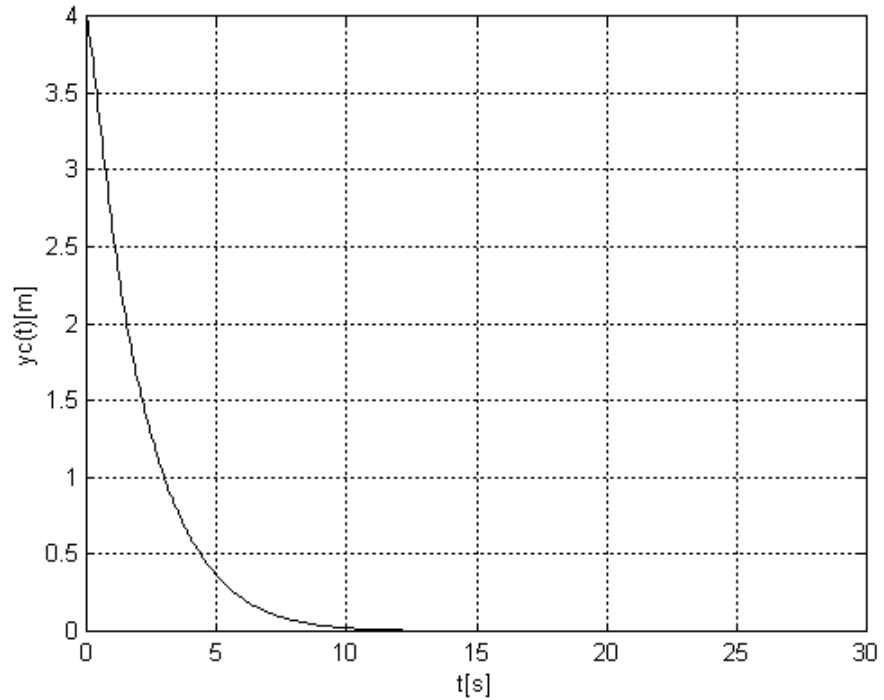


Fig. IV-5 - Resposta de $y_c(t)$ para robô classe (1,2).

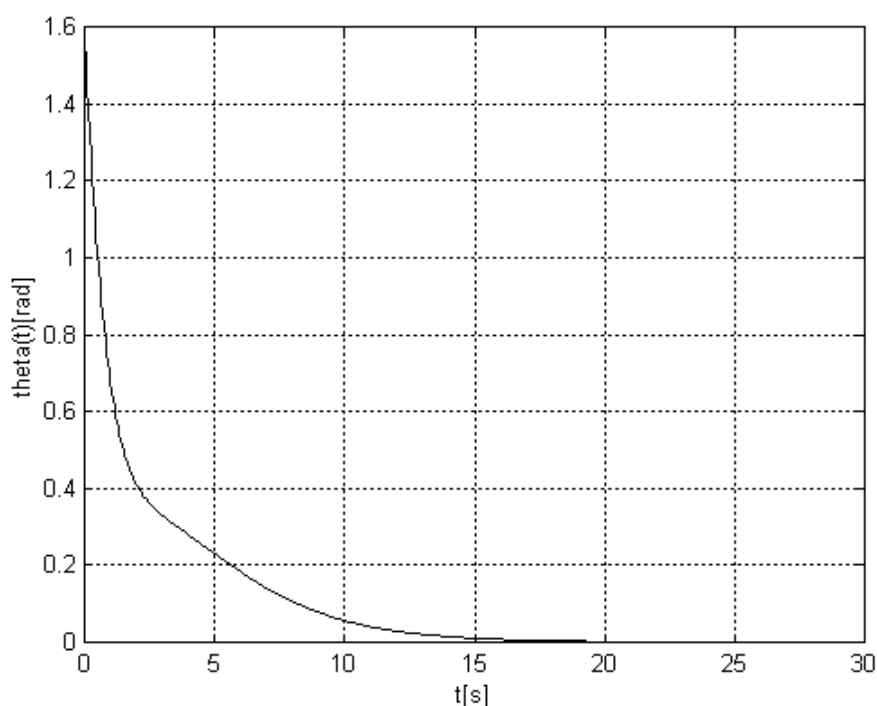


Fig. IV-6 - Resposta de $\theta(t)$ para robô classe (1,2).

IV.3. Backstepping Utilizando PID

IV.3.1. Motivação

Neste capítulo, até aqui, foi considerado apenas o modelo cinemático dos robôs, e que as variáveis de entrada são homogêneas a velocidades ou posição. Na realidade, fisicamente, as entradas do robô são os torques aplicados para rotação e orientação das rodas. Estes torques poderiam ser obtidos derivando-se em relação ao tempo as leis de controle obtidas aqui, e utilizando-se as expressões (II.62-II.64). Porém, esta abordagem leva a problemas de robustez.

Isto no entanto, não significa que as leis de controle obtidas aqui não podem ser utilizadas na prática. Em geral, para robôs pequenos, basta que existam malhas de controle internas que garantam que as velocidades ou posições comandadas sejam efetivamente aplicadas nas rodas. Isto pode ser feito utilizando-se controladores PID convencionais. Esta

técnica é denominada *backstepping* [19]. Na Fig. IV-7 é mostrado um diagrama de blocos do controlador proposto

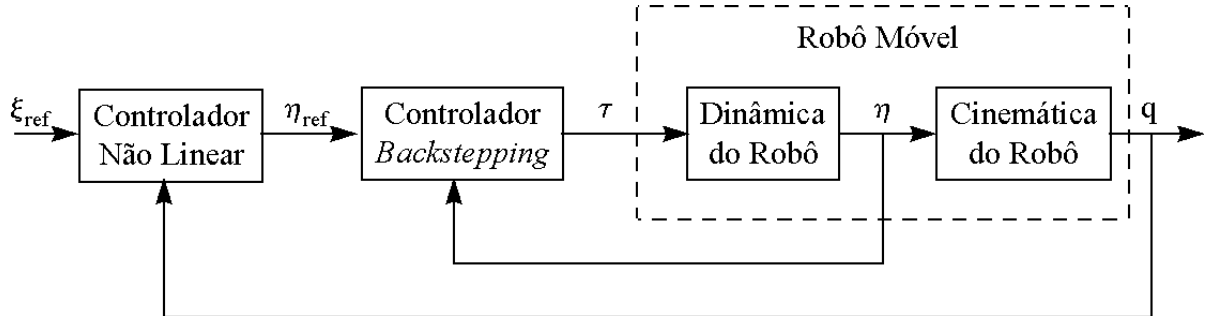


Fig. IV-7 - Controlador com backstepping.

IV.3.2. Resultados de Tempo Real

A lei de controle (IV-29) e (IV-30) foi implementada em tempo real no robô móvel Twil (vide capítulo VII). Este robô é atuado diferencialmente, portanto pertencendo a classe (2,0) e possui rodas com 15cm de diâmetro e uma distância entre rodas (*wheelbase*) de 28.5cm.

As velocidades desejadas nas rodas do robô pode ser computadas de η_1 e η_2 através de (II.40), obtendo-se

$$\omega_r = \frac{\eta_1 + \eta_2 b / 2}{r} \quad (IV.66)$$

$$\omega_l = \frac{\eta_1 - \eta_2 b / 2}{r} \quad (IV.67)$$

onde ω_r e ω_l são as velocidades angulares das rodas direita e esquerda, respectivamente, b é a distância entre as rodas e r é o raio das rodas.

Experimentalmente, determinou-se que a função de transferência entre a velocidade de cada roda e a tensão aplicada no motor correspondente é dada por

$$G(s) = \frac{1.7779(s+5.845)}{s^2 + 10.3078s + 42.4597} \quad (IV.66)$$

Para controlar este sistema, foi utilizado um controlador PID com $K_p=4.154$, $K_i=1.1943$ e $K_d=2.260$, garantido que as velocidades nas rodas são as desejadas.

A trajetória espacial obtida em tempo real e em simulação para estabilizar o robô no ponto $x_c=0$, $y_c=0$ e $\theta=0$, a partir da condição inicial $x_c(0)=0\text{m}$, $y_c(0)=4\text{m}$ e $\theta(0)=0$, é mostrada na Fig. IV-8. A Fig. IV-9 mostra a orientação em função do tempo. Convém notar, que como o robô não pode mover-se lateralmente, este movimento é considerado difícil. É impossível perceber nos gráficos, mas o ponto final corresponde a $x_c(200)=0.0011\text{m}$, $y_c(200)=-1.595 \times 10^{-06}\text{m}$ e $\theta(200)=-0.0006\text{ rad}$ (dados de tempo real). Portanto, o erro final é de aproximadamente 1.13 mm e 0.035° .

Os parâmetros γ_1 , γ_2 e h foram selecionados por simulação. Observa-se que γ_1 , e γ_2 estão relacionados principalmente com a taxa de convergência, enquanto h possui mais influência na forma da trajetória espacial. Para o robô em questão, os valores $\gamma_1=0.05$, $\gamma_2=0.1$ and $h=1.35$ permitem uma boa taxa de convergência sem que haja saturação dos atuadores. As Figs. IV-10 e IV-11 mostram as entradas de controle. Esta estratégia de controle mostrou-se relativamente robusta em relação à saturação dos atuadores, embora obviamente, nesta situação haja uma grande discrepância entre a trajetória real e a trajetória simulada. Ainda assim, a trajetória real converge para o ponto $x_c=0$, $y_c=0$ e $\theta=0$.

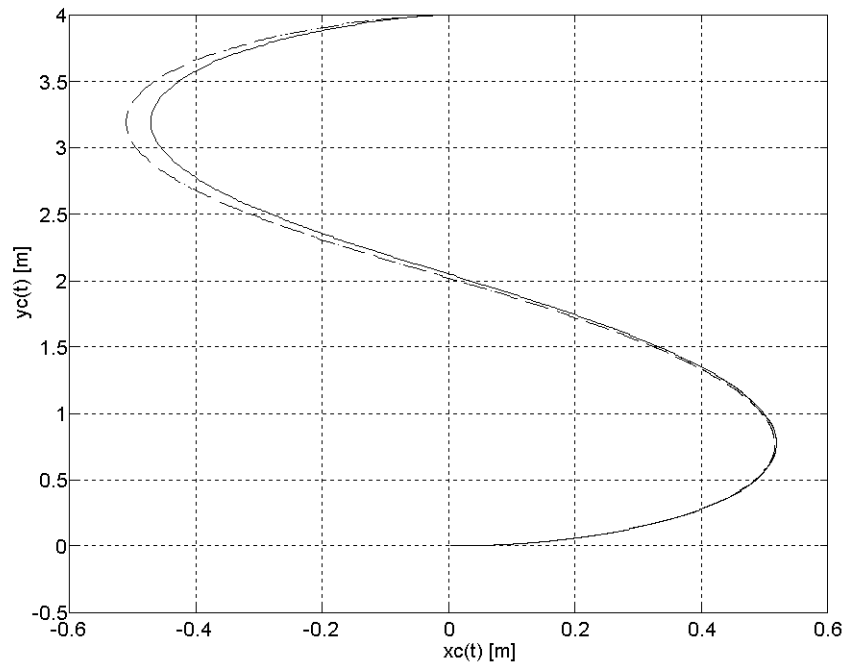


Fig. IV-8 - Trajetória espacial em tempo real (contínua) e simulada (tracejada) para um robô classe (2,0).

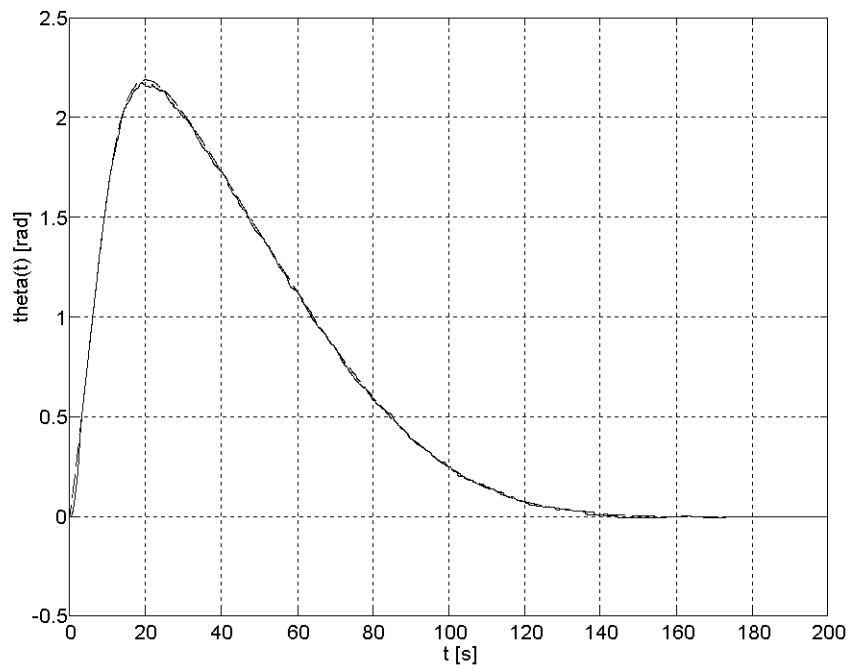


Fig. IV-9 - Resposta de $\theta(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0).

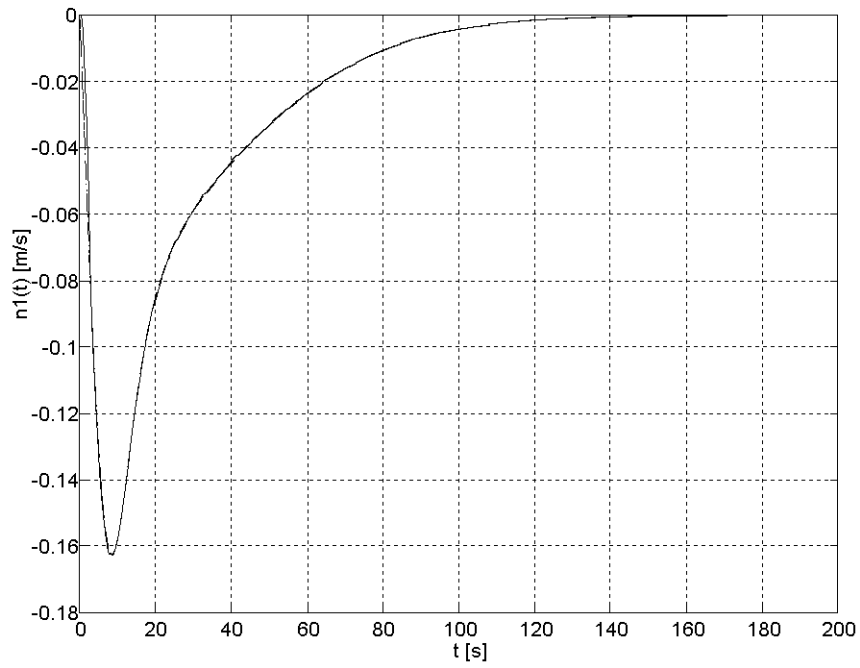


Fig. IV-10 - Resposta de $\eta_1(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0).

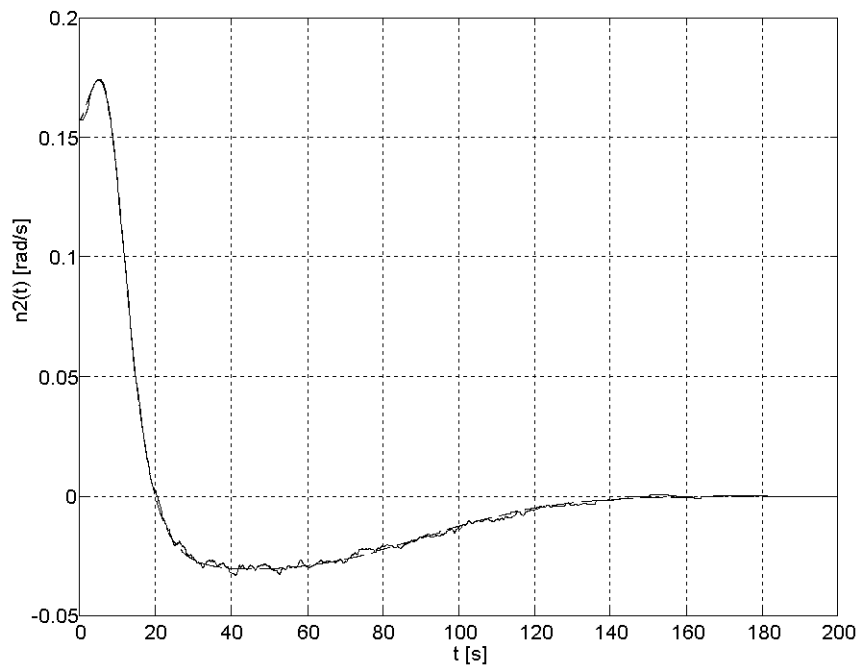


Fig. IV-11 - Resposta de $\eta_2(t)$ em tempo real (contínua) e simulada (tracejada) para robô classe (2,0).

IV.4. Backstepping Utilizando Controle Linearizante

Outra solução para implementar o *backstepping*, é a utilização de um controlador linearizante, semelhante ao proposto no capítulo III, mas aplica-lo aqui apenas para controlar a dinâmica do robô, ou mais especificamente, a dinâmica entre as velocidades e os torques aplicados nos atuadores, conforme o diagrama de blocos da Fig. 3. A utilização desta estratégia de *backstepping* permite que o modelo dinâmico do robô seja considerado explicitamente, ao contrário da estratégia anterior. Obviamente, aqui surgem os mesmos problemas, já comentados no capítulo III, com relação ao conhecimento dos parâmetros do modelo dinâmico. A lei de controle linearizante, neste caso, é a realimentação descrita por (II.63).

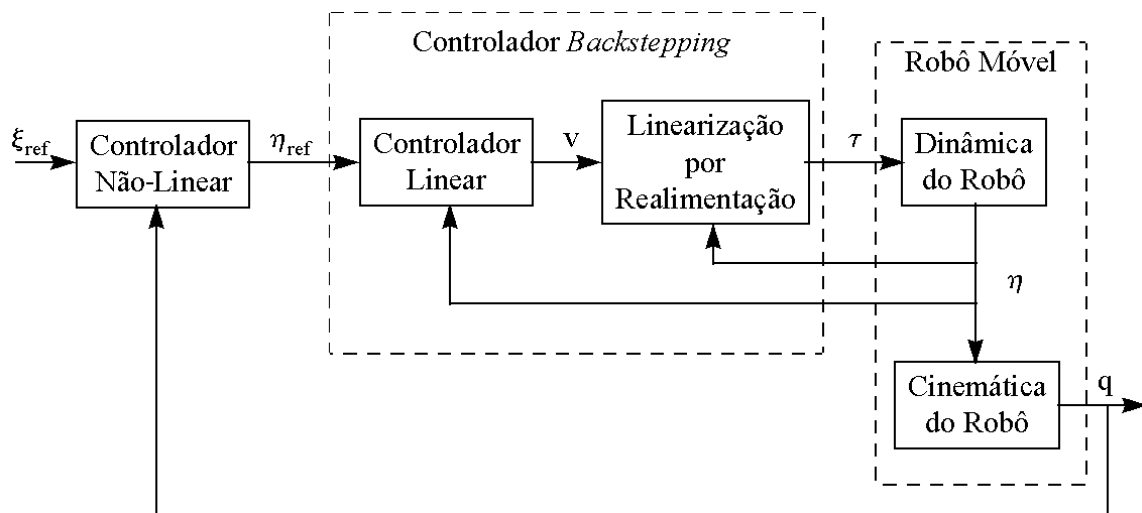


Fig. VI-12 - Backstepping com controlador linearizante.

Da mesma forma que no capítulo III, a utilização de um controlador linear utilizando modelo de referência facilita o desenvolvimento de versões adaptativas do controlador.

IV.5. Conclusão

A técnicas apresentadas neste capítulo permitem que se obtenha leis de controle que garantam a convergência para um ponto e não apresentem oscilações excessivas, como

geralmente ocorre com leis de controle variantes no tempo. Além disso, as leis de controle obtidas são contínuas e suaves.

Mostrou-se que os modelos de robôs pertencentes a uma determinada classe podem ser transformados para modelos equivalentes pertencentes a qualquer outra classe com o mesmo grau de manobrabilidade.

É importante perceber, que as trajetórias apresentadas nas Figs. IV-2 e IV-3 não se prestam para que se tire conclusões acerca de qual classe de robô é mais manobrável ou apresenta melhor desempenho. Para um mesmo robô, com a mesma condição inicial, a trajetória obtida pode ser bastante diferente da apresentada aqui, dependendo dos parâmetros de projeto utilizados no cálculo da lei de controle.

A estratégia de *backstepping* utilizando controle linearizante será utilizada no próximo capítulo para o desenvolvimento de uma versão de controlador adaptativo.

Capítulo V

Controle Adaptativo

Embora controle adaptativo seja usado já há algum tempo em manipuladores robóticos, a sua utilização em robôs móveis não é muito difundida. Obviamente, existem vários trabalhos utilizando variantes menos tradicionais de controle adaptativo, tais como redes neurais [19] [54]; controle inteligente [21], lógica nebulosa [59] e outras técnicas de computação suave. No entanto, estas técnicas parecem ser mais úteis para os níveis mais altos de controle, como controle de missão ou negociação de ambiente. Para os níveis mais baixos de controle, como rastreamento de trajetória ou estabilização em um ponto (*parking*), métodos de controle mais tradicionais devem ser utilizados. A justificativa para isto é que estes níveis tratam com a potência dos atuadores, e não apenas com informação. Portanto, uma falha poderia gerar uma potência mecânica descontrolada, comprometendo a segurança física do robô e do ambiente. Logo, provas formais de estabilidade tornam-se necessárias.

A falta de trabalhos tratando do controle adaptativo de robôs móveis deve-se, em parte, à utilização do modelo cinemático apenas. Como detalhado no capítulo II, este modelo descreve a posição e orientação do robô dadas entradas que são homogêneas à velocidade ou posição. Os parâmetros necessários para este tipo de modelo estão relacionados somente com a geometria do robô e podem ser determinados com boa precisão através de métodos adequados de projeto e construção, medições em protótipo ou procedimentos de calibração [9][10]. De qualquer modo, estes parâmetros são constantes ao longo da vida útil do robô. Conseqüentemente, não se justifica a utilização de controle adaptativo.

No entanto, fisicamente, as entradas de um robô móvel são os torques aplicados para girar ou direcionar as suas rodas [14]. Para robôs pequenos, é possível utilizar controladores

convencionais (por exemplo, PID) para se construir servos de velocidade ou posição para cada roda. Se estes laços de controle internos forem rápidos o suficiente, o sistema apresentará o comportamento descrito pelo modelo cinemático [35]. Por outro lado, para robôs maiores, as forças dinâmicas são significativas e portanto esta abordagem, utilizando servos de velocidade para cada roda, exigiria atuadores muito potentes. Além disso, agindo-se deste modo despreza-se as perturbações mútuas produzidas pelos atuadores.

Considerando-se o modelo dinâmico do robô, as forças dinâmicas podem ser levadas em conta no projeto do controle de posição e orientação do robô. O modelo dinâmico descreve a posição e orientação do robô em função dos torques aplicados para girar ou direcionar as suas rodas. Os parâmetros para este modelo são os mesmos do modelo cinemático complementados por um conjunto de parâmetros relacionados com a dinâmica do robô, como massa e momentos de inércia. Ao contrário dos parâmetros geométricos, os parâmetros dinâmicos podem sofrer variações apreciáveis até mesmo durante uma única missão do robô. Estes parâmetros podem variar, por exemplo, devido à alterações na massa da carga útil, consumo de combustível, variações na distribuição espacial da carga útil, etc. Em geral, estas variações são difíceis de serem determinadas a priori, logo os valores para os parâmetros dinâmicos não podem ser obtidos por procedimentos de calibração. Por outro lado, a estrutura do modelo dinâmico de um robô móvel é bem conhecida, fazendo com que este tipo de sistema seja um bom candidato para a utilização de técnicas de controle adaptativo.

Embora não exista uma teoria geral para o controle adaptativo de sistemas não lineares, sob algumas condições não muito restritivas, é possível desenvolver controladores adaptativos para uma classe bastante representativa de sistemas não lineares [52]. Mais especificamente, estas condições são:

- 1) a dinâmica não linear do sistema pode ser parametrizada linearmente;

2) o estado do sistema é mensurável;

3) se os parâmetros dos sistema forem conhecidos, as não-linearidades podem ser canceladas com dinâmica interna estável, através das entradas de controle.

Nas seções seguintes, será mostrado que um robô móvel preenche as condições acima e leis de controle linearizante adaptativas serão projetadas. Em particular, será utilizado controle (adaptativo) por modelo de referência. As estratégias de adaptação propostas são baseadas em uma formulação de mínimos quadrados, onde o custo a minimizar inclui o erro paramétrico e o erro de rastreamento [33].

Um dos principais problemas da estratégia de controle linearizante apresentada no capítulo III é que se os parâmetros e o estado do sistema não forem perfeitamente conhecidos, os termos não lineares não serão exatamente cancelados. Conseqüentemente, o sistema resultante será diferente do esperado e a lei de controle projetada para o sistema linearizado perderá suas propriedades, provavelmente comprometendo o desempenho do controlador. O problema de incerteza nos parâmetros é tratado aqui desenvolvendo-se versões adaptativas dos controladores propostos nos capítulos anteriores. O problema de determinação do estado é tratado no capítulo VI através de métodos de fusão de dados [10][26]

V.1. Versão Adaptativa do Controle Linearizante

Considere a lei de realimentação linearizante dada por (III.12), que na ausência de incerteza paramétrica pode ser completamente calculada através de (III.7) e (III.10). Se os valores dos parâmetros forem incertos, (III.12) é modificada, com base no princípio da equivalência a certeza, para uma forma adaptativa

$$\mathbf{u}(\mathbf{x}) = -\bar{\mathbf{E}}^{-1}(\mathbf{x}) \begin{bmatrix} \bar{\mathbf{E}}_f^1 \mathbf{h}_1(\mathbf{x}) \\ \mathbf{M} \\ \bar{\mathbf{E}}_f^p \mathbf{h}_p(\mathbf{x}) \end{bmatrix} + \bar{\mathbf{E}}^{-1}(\mathbf{x}) \mathbf{v} \quad (\text{V.1})$$

onde $\bar{\mathbf{E}}^{-1}(\mathbf{x})$ e $\bar{\mathbf{E}}_r^{\gamma_p} h_p(\mathbf{x})$ são estimativas de $\mathbf{E}^{-1}(\mathbf{x})$ e $\mathbf{L}_r^{\gamma_p} h_p(\mathbf{x})$, respectivamente.

Por conveniência, a expressão (V.1) é escrita de forma mais compacta

$$\mathbf{u}(\mathbf{x}) = -\bar{\mathbf{E}}^{-1}(\mathbf{x})\bar{\mathbf{E}}_r h(\mathbf{x}) + \bar{\mathbf{E}}^{-1}(\mathbf{x})\mathbf{v} \quad (\text{V.2})$$

Similarmente, a expressão (III.11) torna-se

$$\mathbf{y}^\gamma = \mathbf{L}_r h(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{u} \quad (\text{V.3})$$

De (V.2) é fácil verificar que

$$\mathbf{v} = \bar{\mathbf{E}}_r h(\mathbf{x}) + \bar{\mathbf{E}}(\mathbf{x})\mathbf{u} \quad (\text{V.4})$$

Subtraindo-se (V.4) de (V.3) e rearranjando os termos, obtém-se

$$\mathbf{y}^\gamma = \mathbf{v} + \mathbf{L}_r h(\mathbf{x}) - \bar{\mathbf{E}}_r h(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{u} - \bar{\mathbf{E}}(\mathbf{x})\mathbf{u} \quad (\text{V.5})$$

Se $\mathbf{L}_r h(\mathbf{x})$ e $\mathbf{E}(\mathbf{x})$ puderem ser parametrizadas linearmente em função do vetor de parâmetros desconhecidos Θ , então a expressão (V.5) pode ser escrita na forma

$$\mathbf{y}^\gamma = \mathbf{v} + \mathbf{W}\Phi \quad (\text{V.6})$$

onde $\Phi = \Theta - \bar{\Theta}$ é o vetor de erros paramétricos e \mathbf{W} é a matriz de regressão.

O algoritmo utilizado para estimar os parâmetros desconhecidos é baseado na seguinte função custo, proposta em [52]

$$J(\Phi, \mathbf{e}) = \int_0^t \left(\mathbf{L}^{-1}(s)(\mathbf{W}\Phi) \right)^T \left(\mathbf{L}^{-1}(s)(\mathbf{W}\Phi) \right) + \mathbf{e}^T \mathbf{e} d\tau \quad (\text{V.7})$$

onde $\mathbf{e} = \mathbf{y} - \mathbf{y}_r$ representa o vetor de erro de rastreamento entre a saída do sistema e a saída do modelo de referência, $\mathbf{W}\Phi$ representa o erro de estimação paramétrica e $\mathbf{L}(s)$ é o polinômio definido por

$$\mathbf{L}(s) = s^\gamma + \alpha_\gamma s^{\gamma-1} + \Lambda + \alpha_1 \quad (\text{V.8})$$

sendo α_i os coeficientes do polinômio característico do modelo de referência.

A notação híbrida $L^{-1}(s)(W\Phi)$ denota $W\Phi$ filtrado através de um filtro com função de transferência $L^{-1}(s)$. Ou seja, a convolução entre a transformada inversa de Laplace de $L^{-1}(s)$ e $W\Phi$.

A minimização da função custo (V.7) com relação a $\bar{\Theta}$ resulta na lei de adaptação de parâmetros

$$\dot{\bar{\Theta}} = -\dot{\bar{\Theta}} = -PW_f^T [I + W_f P W_f^T + W_f W_f^T]^{-1} W_f \Phi - \alpha_a P X^T [I + X P X^T + X X^T]^{-1} X \Phi \quad (V.9)$$

$$\dot{P} = -P W_f^T [I + W_f P W_f^T + W_f W_f^T]^{-1} W_f P \quad (V.10)$$

onde

W_f é a matriz de regressão filtrada por $F(s)$

α_a é o ganho de adaptação

X é a matriz de regressão filtrada pelo modelo de referência $G(s)$

O filtro $F(s)$ tem a forma

$$F(s) = \frac{1}{(\varepsilon s + 1)^\gamma} \quad (V.11)$$

sendo ε um parâmetro de projeto escolhido de forma que o filtro possua uma dinâmica bem mais rápida do que o modelo de referência.

Note-se que embora a variável Φ não possa ser computada em tempo real, os produtos $W_f \Phi$ e $X \Phi$ podem ser computados através das expressões

$$W_f \Phi = e_f^\gamma + \alpha_\gamma e_f^{\gamma-1} + \Lambda + \alpha_1 e_f + F(s)(W\bar{\Theta}) - (F(s)W)\bar{\Theta} \quad (V.12)$$

$$X \Phi = e + L^{-1}(s)(W\bar{\Theta}) - (L^{-1}(s)W)\bar{\Theta} \quad (V.13)$$

onde $e_f^\gamma, e_f^{\gamma-1}, \dots, e_f$ são obtidos filtrando-se o erro de rastreamento através de $s^\gamma F(s), s^{\gamma-1} F(s), \dots, F(s)$. É importante perceber que estes filtros são próprios, por construção.

V.1.1. Parametrização do Modelo de Robôs Móveis

Nesta seção será mostrado que o modelo de um robô móvel pode ser parametrizado na forma da expressão (V.6), possibilitando o uso da técnica de controle linearizante adaptativo descrita na seção anterior. Para simplificar os desenvolvimentos, assume-se um robô da classe (2,0), mas resultados semelhantes podem ser obtidos com robôs de qualquer classe. Além disso, pelos motivos expostos nos comentários iniciais deste capítulo, considera-se que os parâmetros de massa e momentos de inércia são desconhecidos e os parâmetros geométricos são conhecidos.

Considere o modelo dinâmico de postura de um robô da classe (2,0), como na expressão (III.22). Pode ser observado que este modelo não possui nenhum parâmetro de massa ou momento de inércia. Isto ocorre porque os termos dependentes destes parâmetros foram cancelados através da realimentação (II.63), utilizada para obtenção deste modelo. Portanto, para que estes parâmetros apareçam explicitamente no modelo, é necessário que seja utilizado o modelo dinâmico de configuração, expresso por (II.61). No caso de um robô da classe (2,0), este modelo assume a seguinte forma [36]

$$\ddot{\mathbf{x}} = \begin{bmatrix} S\mathbf{v} \\ \mathbf{f}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (S^T M S)^{-1} \end{bmatrix} \boldsymbol{\tau} \quad (\text{V.14})$$

com

$$\mathbf{f}_2 = (S^T M S)^{-1} (-S^T M \ddot{\mathbf{v}} - S^T \mathbf{V}) \quad (\text{V.15})$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{q} \\ \mathbf{v} \end{bmatrix} = [x_c \quad y_c \quad \theta \quad \varphi_r \quad \varphi_l \quad \dot{\varphi}_r \quad \dot{\varphi}_l]^T \quad (\text{V.16})$$

$$M = \begin{bmatrix} m & 0 & m_c d \sin \theta & 0 & 0 \\ 0 & m & -m_c d \cos \theta & 0 & 0 \\ m_c d \sin \theta & -m_c d \cos \theta & I & 0 & 0 \\ 0 & 0 & 0 & I_w & 0 \\ 0 & 0 & 0 & 0 & I_w \end{bmatrix} \quad (\text{V.17})$$

$$V = [m_c d \ddot{\theta} \cos \theta \quad m_c d \ddot{\theta} \sin \theta \quad 0 \quad 0 \quad 0]^T \quad (V.18)$$

$$S = \begin{bmatrix} c(d \sin \theta + b \cos \theta) & c(b \cos \theta - d \sin \theta) \\ c(b \sin \theta - d \cos \theta) & c(b \sin \theta + d \cos \theta) \\ c & -c \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (V.19)$$

$$c = \frac{r}{2b} \quad (V.20)$$

onde

P_o : intersecção entre os eixos de simetria e o eixo das rodas;

P_c : centro de massa da plataforma;

d : distância entre P_o e P_c ;

b : distância entre as rodas e o eixo de simetria;

r : raio da roda;

m_c : massa da plataforma sem considerar as rodas e os motores;

m_w : massa de cada conjunto roda/motor;

I : momento de inércia gerado por m_c e m_w com relação ao eixo vertical, a partir de P_o ;

I_w : momento de inércia gerado pelo conjunto roda/motor, com relação ao eixo das rodas;

τ : torque nas rodas.

Definindo-se o vetor de saídas do sistema de forma idêntica ao capítulo III, como definido na expressão (III.23), tem-se que a entrada de torque que lineariza o sistema (V.14) é dada, conforme (III.12) por

$$\tau = -E^{-1} L_f h + E^{-1} v \quad (V.21)$$

com

$$E^{-1} = S^T M S \beta^{-1} \quad (V.22)$$

$$L_f h = \alpha + \beta (S^T M S)^{-1} (-S^T V - S^T M S v) = \alpha + \beta f_2 \quad (V.23)$$

sendo

$$\alpha = c^2 \begin{bmatrix} (\phi_r^c - \phi_l^c)^2 [(d - x_r^c) \cos \theta + y_r^c \sin \theta] - (\phi_r^2 - \phi_l^2) b \sin \theta \\ (\phi_r^c - \phi_l^c)^2 [(d - x_r^c) \sin \theta - y_r^c \cos \theta] + (\phi_r^2 - \phi_l^2) b \cos \theta \end{bmatrix} \quad (V.24)$$

$$\beta = c \begin{bmatrix} (d - x_r^c) \sin \theta + (b - y_r^c) \cos \theta & -(d - x_r^c) \sin \theta + (b + y_r^c) \cos \theta \\ (b - y_r^c) \sin \theta - (d - x_r^c) \cos \theta & (b + y_r^c) \sin \theta + (d - x_r^c) \cos \theta \end{bmatrix} \quad (V.25)$$

Retornando ao problema da parametrização, de (V.5) e (V.6) pode ser verificado que

$$W\Phi = L_r h(\mathbf{x}) - \bar{E}_r h(\mathbf{x}) + E(\mathbf{x})\mathbf{u} - \bar{E}(\mathbf{x})\mathbf{u} \quad (V.26)$$

substituindo-se $L_r h(\mathbf{x})$ e $E(\mathbf{x})\mathbf{u}$ de (V.22.) e (V.23) resulta

$$W\Phi = \beta \begin{bmatrix} 64 \ 7 \ 48 \\ f_2 - \bar{f}_2 + (S^T MS)^{-1} \tau - (S^T MS)^{-1} \tau \end{bmatrix} \quad (V.27)$$

uma vez que α e β não dependem dos parâmetros desconhecidos.

Computando-se literalmente f_2 e $(S^T MS)^{-1}$ obtém-se as seguintes expressões

$$f_2 = c^3 db \begin{bmatrix} (\phi_r^2 - \phi_l^2 \phi_r^c) & (\phi_r^2 - \phi_l^2 \phi_r^c) \\ (\phi_r^2 - \phi_l^2 \phi_r^c) & (\phi_r^2 - \phi_l^2 \phi_r^c) \end{bmatrix} \begin{bmatrix} K_{sc} \\ K_{pc} \end{bmatrix} \quad (V.28)$$

$$(S^T MS)^{-1} = \begin{bmatrix} K_p & K_s \\ K_s & K_p \end{bmatrix} \quad (V.29)$$

onde K_{sc} , K_{pc} , K_p e K_s são constantes dependentes apenas dos parâmetros desconhecidos (massa e momentos de inércia) e dos parâmetros geométricos do robô.

Portanto, lembrando que Φ é o vetor de erros paramétricos e que $\tau = [\tau_r \ \tau_l]^T$, $W\Phi$ pode ser escrito na forma

$$W\Phi = \beta \begin{bmatrix} \tau_r & \tau_l & c^3 db (\phi_r^2 - \phi_l^2 \phi_r^c) & c^3 db (\phi_r^2 - \phi_l^2 \phi_r^c) \\ \tau_l & \tau_r & c^3 db (\phi_r^2 - \phi_l^2 \phi_r^c) & c^3 db (\phi_r^2 - \phi_l^2 \phi_r^c) \end{bmatrix} \begin{bmatrix} K_p - \bar{K}_p \\ K_s - \bar{K}_s \\ K_{sc} - \bar{K}_{sc} \\ K_{pc} - \bar{K}_{pc} \end{bmatrix} \quad (V.30)$$

logo,

$$W = \beta \begin{bmatrix} \tau_r & \tau_l & c^3 db(\phi_r^2 - \phi_r \phi_l) & c^3 db(\phi_l^2 - \phi_r \phi_l) \\ \tau_l & \tau_r & c^3 db(\phi_l^2 - \phi_r \phi_l) & c^3 db(\phi_r^2 - \phi_r \phi_l) \end{bmatrix} \quad (V.31)$$

e

$$\Phi = \begin{bmatrix} K_p - \bar{K}_p \\ K_s - \bar{K}_s \\ K_{sc} - \bar{K}_{sc} \\ K_{pc} - \bar{K}_{pc} \end{bmatrix} \quad (V.32)$$

conseqüentemente, o vetor de parâmetros estimados será $\bar{\Theta} = [\bar{K}_p \quad \bar{K}_s \quad \bar{K}_{sc} \quad \bar{K}_{pc}]^T$.

Como se supõe que os parâmetros desconhecidos Θ são constantes, é óbvio que $\dot{\Theta} = -\dot{\bar{\Theta}}$. Portanto, dada uma estimativa inicial $\bar{\Theta}(0)$, pode-se atualizá-la recursivamente utilizando (V.9) e (V.10) e a partir de (V.28) e (V.29) computar

$$\bar{F}_2 = c^3 db \begin{bmatrix} (\phi_r^2 - \phi_r \phi_l) & (\phi_l^2 - \phi_r \phi_l) \\ (\phi_l^2 - \phi_r \phi_l) & (\phi_r^2 - \phi_r \phi_l) \end{bmatrix} \begin{bmatrix} \bar{K}_{sc} \\ \bar{K}_{pc} \end{bmatrix} \quad (V.33)$$

$$(S^T MS)^{-1} = \begin{bmatrix} \bar{K}_p & \bar{K}_s \\ \bar{K}_s & \bar{K}_p \end{bmatrix} \quad (V.34)$$

A seguir, utilizando-se (V.22) e (V.23) pode-se determinar a entrada de torque linearizante

$$\tau = (S^T MS)^{-1} (v - \alpha - \beta \bar{F}_2) \quad (V.35)$$

A inversibilidade de β é garantida sob as mesmas condições discutidas no capítulo III. No caso de robôs da classe (2,0), isto significa que o ponto de referência no robô deve ser

escolhido fora do centro das rodas. A matriz $(S^T MS)^{-1}$ também tem sua inversibilidade assegurada para os valores reais de \bar{K}_p e \bar{K}_s . No entanto, como estes valores são estimados,

pode ocorrer que assumam valores tais que $\bar{K}_p^2 = \bar{K}_s^2$. Neste caso, a matriz $(S^T MS)^{-1}$ tornar-

se-á singular e o torque linearizante não poderá ser computado. A solução para este problema se dá através do chaveamento da entrada do sistema para um sinal de teste quando a lei (V.35) não puder ser computada. Este sinal de teste força o sistema para fora da região de singularidade. Pode ser provado que um número finito de chaveamentos entre o sinal de controle e a lei (V.35) serão realizados. Vide [22][52] para detalhes. Além disso, o custo utilizado para desenvolvimento do algoritmo de estimação paramétrica procura fazer com que os parâmetros converjam para os seus valores reais, para os quais não existe o problema de singularidade.

Um diagrama de blocos da estratégia de controle linearizante adaptativo pode ser visto na Fig. V.1.

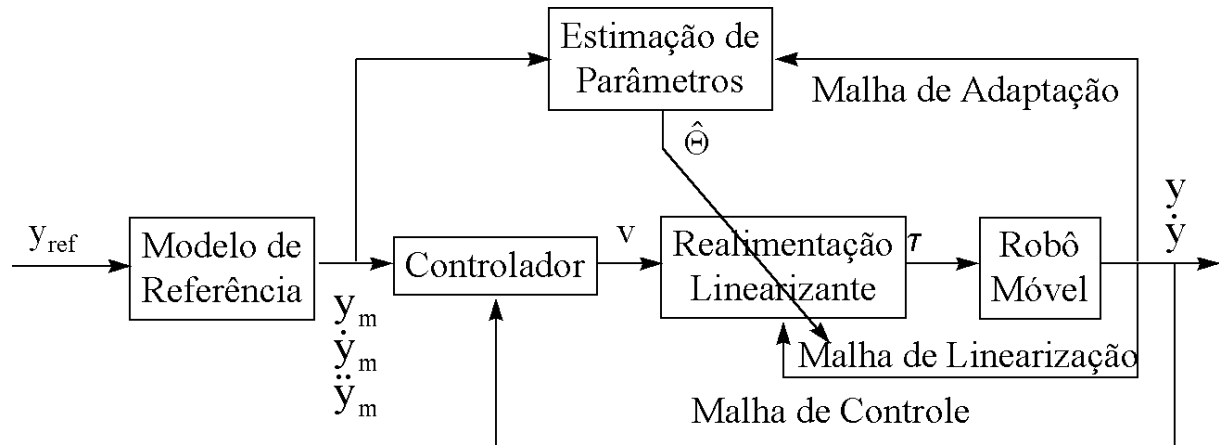


Fig. V-1 - Diagrama de blocos do controle linearizante adaptativo

V.1.2. Resultados de Simulação

Resultados de simulação para o robô móvel Twil são mostrados nas Figs. V-2 a V-5. Convém notar que neste robô o centro de massa é coincidente com o centro das rodas, portanto, tem-se que $f_2 = \bar{f}_2 = 0$. Logo, existem apenas dois parâmetros a estimar. O modelo de referência utilizado para ambas as direções foi

$$G_{m1}(s) = G_{m2}(s) = \frac{10}{s^2 + 1.4\sqrt{10}s + 10} \quad (V.36)$$

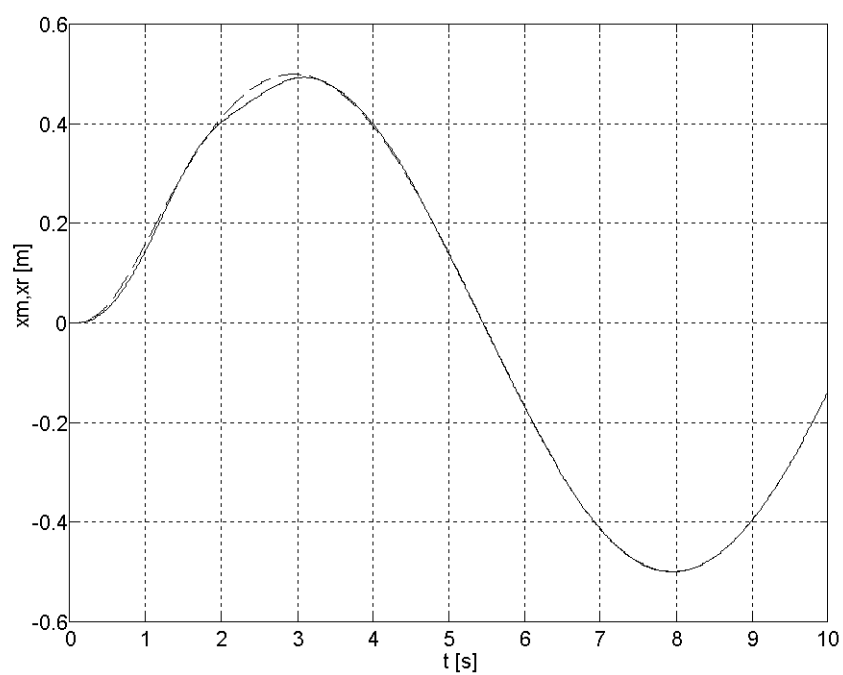


Fig. V-2 - Saída do modelo de referência (tracejada) e posição simulada do robô na direção X_0 (sólida).

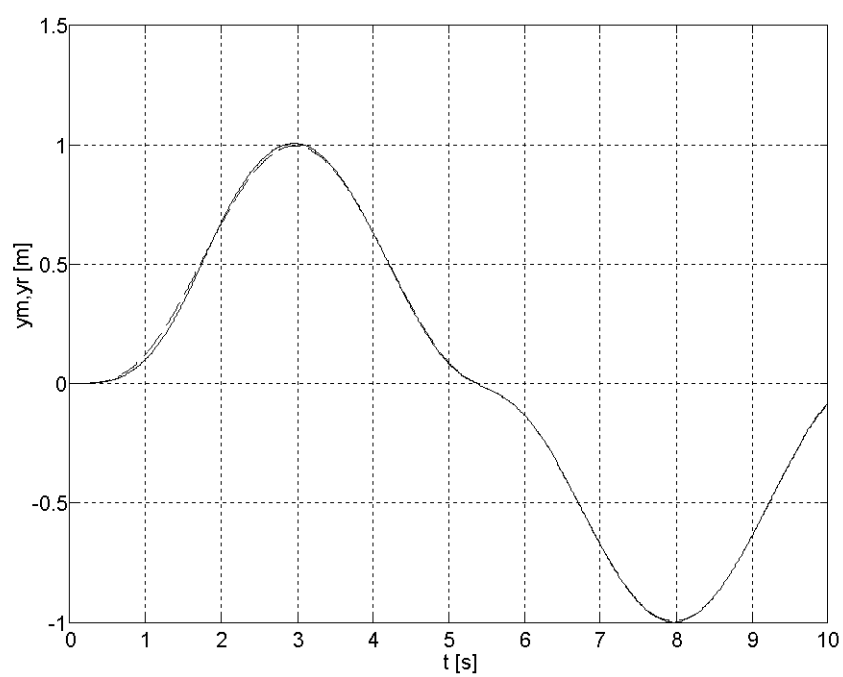


Fig. V-3 - Saída do modelo de referência (tracejada) e posição simulada do robô na direção Y_0 (sólida).

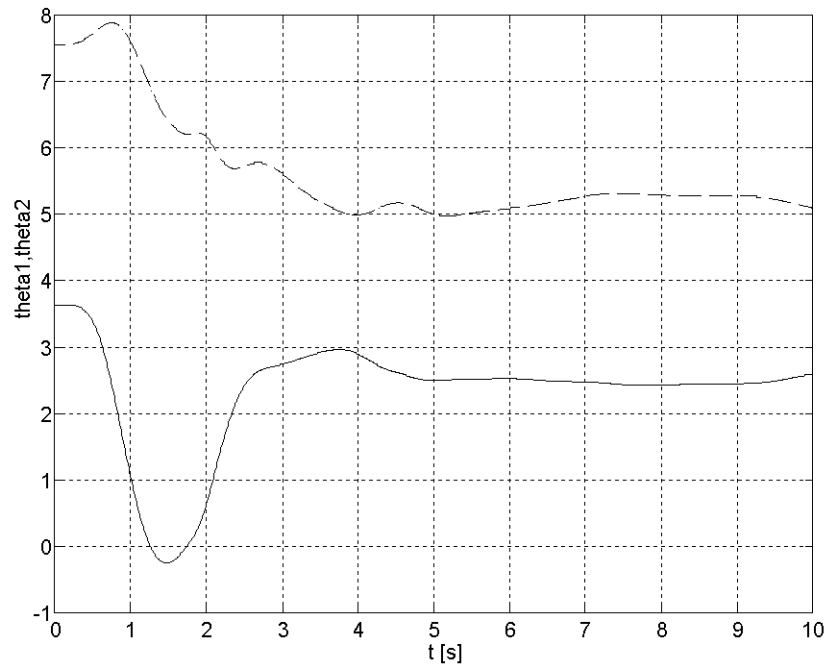


Fig. V-4 - Evolução simulada de $\bar{\Theta}_1(t)$ (tracejada) e $\bar{\Theta}_2(t)$ (sólida).

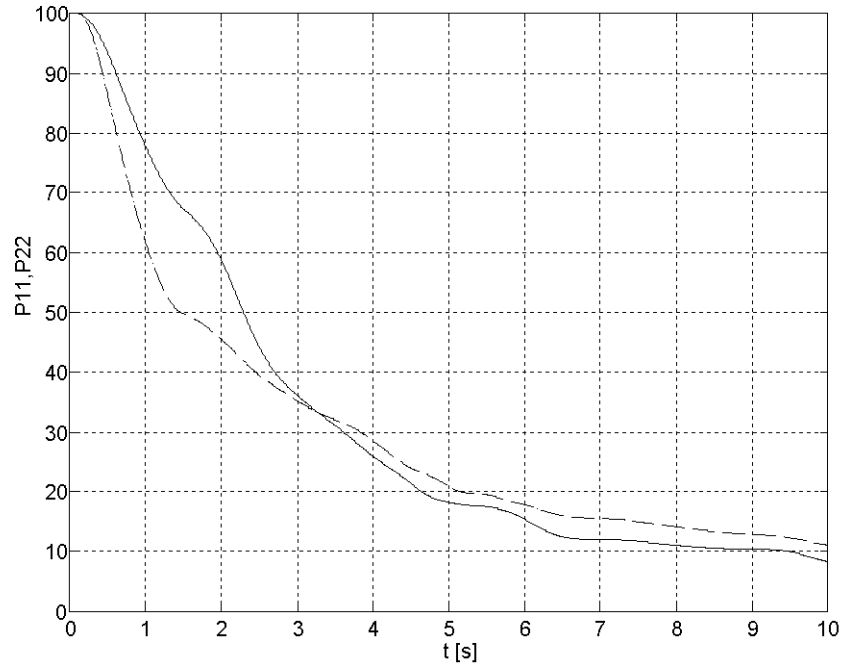


Fig. V-5 - Evolução simulada da diagonal da matriz de covariância.

Os valores iniciais dos parâmetros desconhecidos foram escolhidos como 50% superiores aos seus valores reais. A matriz de covariância foi inicializada com $P=100I$ e

condições iniciais nulas foram utilizadas em todos os filtros. O período de amostragem foi de 10ms.

Como esperado, o desempenho do controlador melhora com o tempo devido à sintonia dos parâmetros desconhecidos, que convergem para os seus valores reais. Após 10s, os valores estimados são $\hat{\Theta}_1 = 5.09242$ e $\hat{\Theta}_2 = 2.58819$, que estão bastante próximos dos valores reais: $\hat{\Theta}_1 = 5.03886$ e $\hat{\Theta}_2 = 2.42035$.

V.1.3. Resultados Experimentais

Resultados experimentais para as versões adaptativas e não adaptativas do controlador linearizante são mostradas nas Figs. V-6 a V-9. Os parâmetros desconhecidos foram inicializados com seus valores nominais. Como pode ser observado, inicialmente ambos os controladores apresentam um desempenho semelhante, mas com o passar do tempo, o desempenho do controlador adaptativo melhora devido a sintonia dos parâmetros. Neste caso, os parâmetros estimados provavelmente não convergiram para os seus valores reais. Apesar disto, o erro de rastreamento diminui com a sintonia dos parâmetros.

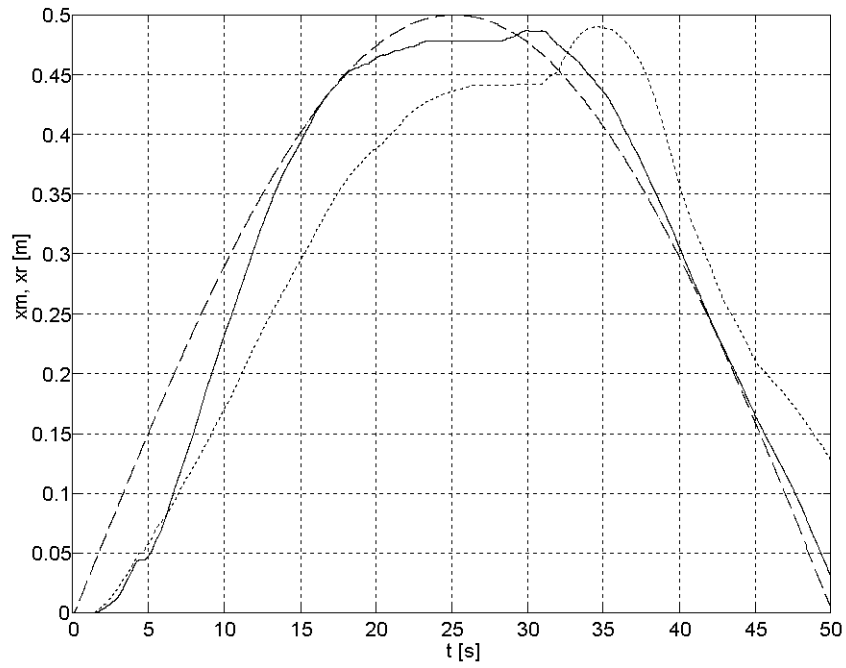


Fig. V-6 - Saída do modelo de referência (tracejada), posição do robô na direção X_0 com controlador adaptativo (sólida) e posição do robô na direção X_0 com controlador não-adaptativo (pontilhada).

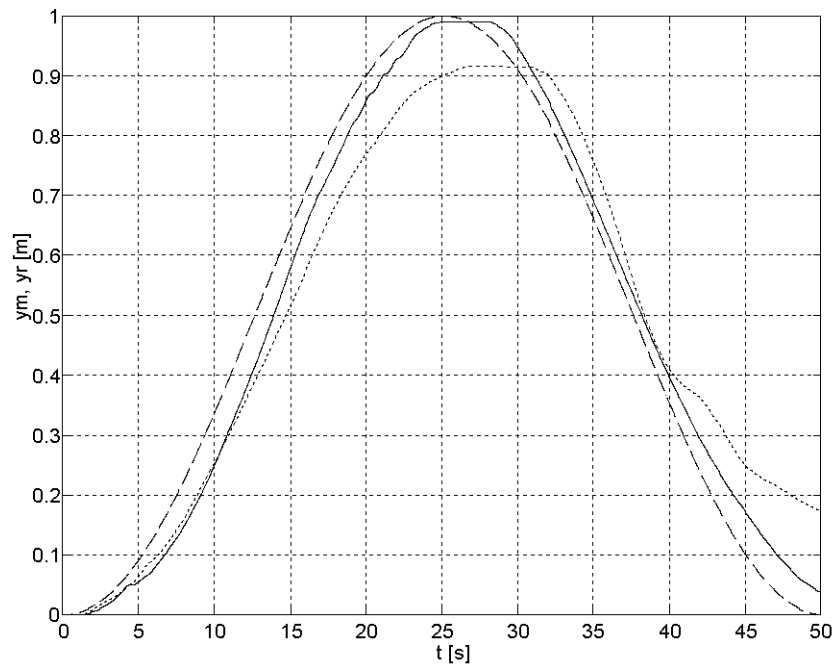


Fig. V-7 - Saída do modelo de referência (tracejada), posição do robô na direção Y_0 com controlador adaptativo (sólida) e posição do robô na direção Y_0 com controlador não-adaptativo (pontilhada).

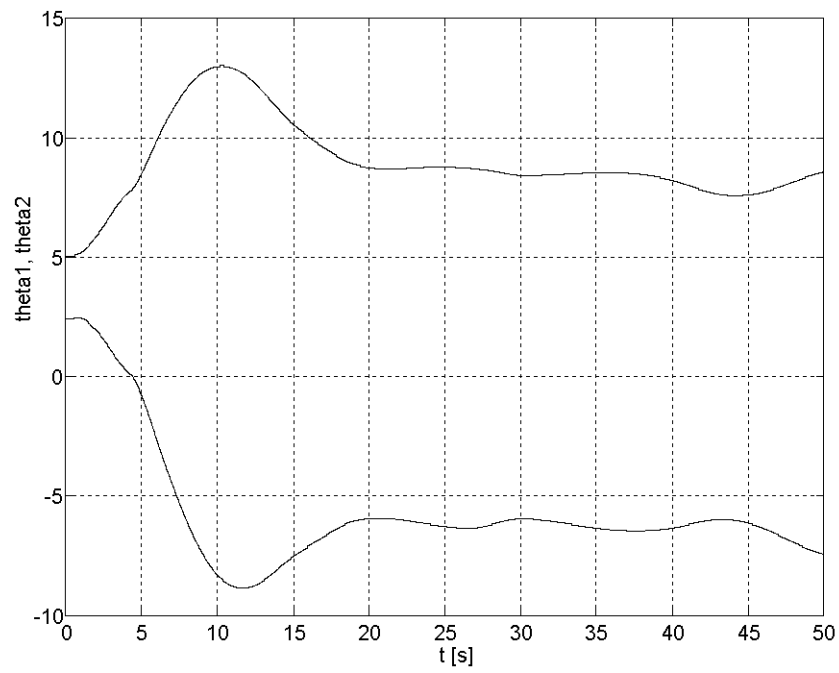


Fig. V-8 - Evolução de $\bar{\Theta}_1(t)$ e $\bar{\Theta}_2(t)$ em tempo real.

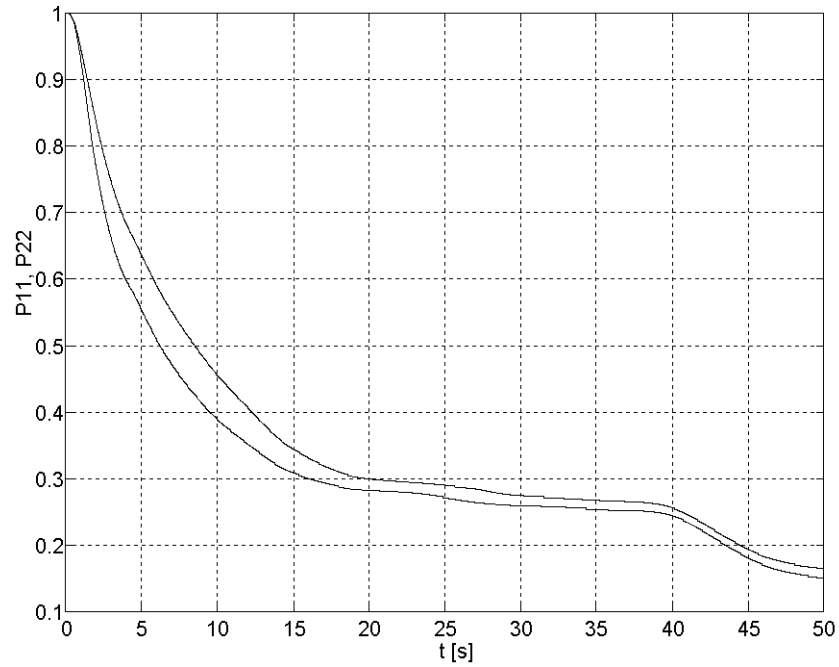


Fig. V-9 - Evolução da diagonal da matriz de covariância em tempo real.

V.2. Versão Adaptativa do Controle por Transformação Descontínua

Por ser baseado no controlador linearizante apresentado no capítulo III, o controle adaptativo apresentado na seção V.1 possui também as mesmas limitações quanto às variáveis de postura que podem ser controladas. Ou seja, não é possível o controle da orientação de robôs das classes (2,0) e (2,1) e para robôs das classes (1,1) e (1,2) esta estratégia de controle é praticamente inútil.

Estas limitações podem ser contornadas utilizando-se um controlador de *backstepping* [19], de forma semelhante ao que foi apresentado no capítulo IV. Na Fig. V-10 é apresentado um diagrama de blocos da versão adaptativa do controle por transformação descontínua adaptativo.

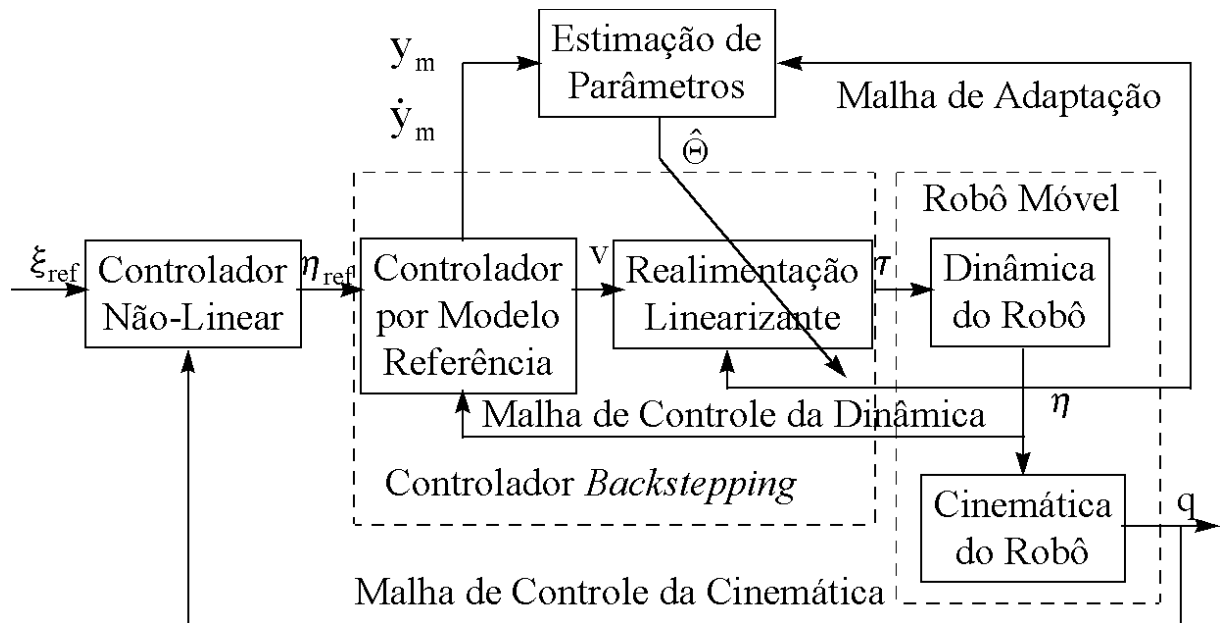


Fig. V-10 - Diagrama de blocos do controle por transformação descontínua adaptativo

O desenvolvimento deste controlador é baseado na observação de que o modelo de um robô móvel é na realidade formado por dois sistemas dinâmico em cascata, como pode ser observado na Fig. V-10 e pelas expressões (II.61) e (V.14). A entrada do sistema são os torques aplicados nas rodas do robô. Estes torques aplicados à dinâmica do robô geram as

velocidades nas rodas, que aplicadas à cinemática do robô geram a sua posição e orientação.

De (V.14), pode-se escrever as seguintes expressões

$$\dot{\mathbf{q}} = \mathbf{S}\mathbf{v} \quad (\text{V.37})$$

que representa a cinemática do robô, enquanto a dinâmica é representada por

$$\dot{\mathbf{v}} = \mathbf{f}_2 + (\mathbf{S}^T \mathbf{M} \mathbf{S})^{-1} \boldsymbol{\tau} \quad (\text{V.38})$$

Convém notar que embora a expressão (V.38) aparentemente dependa de variáveis geradas por (V.37), isto efetivamente não ocorre, pois estas variáveis são canceladas nos cálculos ou podem ser computadas em função de \mathbf{v} . Fisicamente, isto pode ser explicado pelo fato de que os torques que precisam ser aplicados para gerar uma certa aceleração nas rodas não dependem da posição ou orientação do robô no espaço, mas tão somente das velocidades das rodas e dos parâmetros de massa e inércia do robô.

Com base nestes fatos, propõe-se um controlador não linear para controlar a cinemática do robô, descrita por (V.37) e uma malha interna de controle de velocidade, tal como desenvolvido no capítulo IV. Obviamente, a implementação deste controle linearizante está sujeita aos mesmos problemas com relação aos valores dos parâmetros dinâmicos já comentados na seção V.1. Adota-se portanto, uma solução semelhante, utilizando-se uma versão adaptativa do controlador linearizante. Porém, aqui a situação é mais favorável, pois o sistema (V.38) apresenta grau relativo 1, permitido que a prova de estabilidade do sistema em malha fechada, isto é processo, controle e adaptação seja realizada em conjunto [52].

V.2.1. Parametrização do Modelo

Como o sistema descrito por (V.38) possui grau relativo 1, pode-se escrever (V.38) na forma

$$\dot{\mathbf{y}} = \mathbf{L}_f \mathbf{h}(\mathbf{x}) + \mathbf{E}(\mathbf{x})\mathbf{u} \quad (\text{V.39})$$

com $L_f h(x) = f_2$ e $E(x) = (S^T MS)^{-1}$, que analogamente à seção V.1 pode ser parametrizado linearmente na forma

$$\hat{y} = v + W\Phi \quad (V.40)$$

onde

$$W = \begin{bmatrix} \tau_r & \tau_l & c^3 db(\phi_r^2 - \phi_r \phi_l) & c^3 db(\phi_l^2 - \phi_r \phi_l) \\ \tau_l & \tau_r & c^3 db(\phi_l^2 - \phi_r \phi_l) & c^3 db(\phi_r^2 - \phi_r \phi_l) \end{bmatrix} \quad (V.41)$$

e

$$\Phi = \begin{bmatrix} K_p - \bar{K}_p \\ K_s - \bar{K}_s \\ K_{sc} - \bar{K}_{sc} \\ K_{pc} - \bar{K}_{pc} \end{bmatrix} \quad (V.42)$$

É conveniente notar que o vetor de erro paramétrico é exatamente o mesmo obtido na seção V.1, pois os parâmetros desconhecidos também são os mesmos. A matriz de regressão também é bastante semelhante. A lei de controle que lineariza a dinâmica do robô é dada por

$$u = E^{-1}(x)(v - L_f h(x)) = \tau = S^T MS(v - f_2) \quad (V.43)$$

ou, na forma adaptativa

$$u = \bar{E}^{-1}(x)(v - \bar{E}_f h(x)) = \tau = S^T MS(v - \bar{f}_2) \quad (V.44)$$

sendo v o vetor de entradas equivalentes, computado tal como no capítulo III, utilizando-se os modelos de referência

$$G_{m1}(s) = \frac{\alpha_1}{s + \alpha_1} \quad (V.45)$$

$$G_{m2}(s) = \frac{\alpha_2}{s + \alpha_2} \quad (V.46)$$

ou seja,

$$v = \begin{bmatrix} \hat{y}_{m1} + \alpha_1(y_{m1} - x_r) \\ \hat{y}_{m2} + \alpha_2(y_{m2} - y_r) \end{bmatrix} = \hat{y}_m + \alpha(y_m - y) \quad (V.47)$$

Definindo-se o erro de rastreamento $e = y - y_m$, tem-se, através da substituição de (V.47) em (V.40)

$$\hat{y}_m + \alpha e = W\Phi \quad (V.48)$$

que pode ser escrita, utilizando-se a notação híbrida empregada na seção V.1, na forma

$$e = M(s)(W\Phi) \quad (V.49)$$

$$\text{com } M(s) = \frac{1}{s + \alpha}.$$

A expressão (V.48) pode ainda ser escrita na forma

$$\hat{y}_m - \hat{y}_m + \alpha e = W\Theta - W\bar{\Theta} \quad (V.50)$$

de onde, lembrando-se que $\hat{y}_m = W\Theta$, obtém-se

$$e = \frac{1}{\alpha}(\hat{y}_m - W\hat{\Theta}) \quad (V.51)$$

V.2.2. Lei de Adaptação

O algoritmo de estimação de parâmetros desenvolvido para estimação paramétrica aplica-se a sistemas tipo MIMO de fase mínima e grau relativo 1, descritos pela expressão (V.39) com $\det(E(x)) \neq 0$ e é baseado na seguinte função custo

$$J(\Phi, e) = \int_0^{\infty} \frac{1}{\alpha^2} \Phi^T W^T W \Phi + e^T e d\tau \quad (V.52)$$

onde e representa o erro de rastreamento e $W\Phi$ representa o erro paramétrico.

Utilizando-se (V.51), a função custo (V.52) pode ser reescrita como

$$J(\Theta, \bar{\Theta}) = \int_0^{\infty} \frac{1}{\alpha^2} (\Theta^T - \bar{\Theta}^T) W^T W (\Theta - \bar{\Theta}) + \frac{1}{\alpha^2} (\hat{y}_m^T - \Theta^T W^T) (\hat{y}_m - W\Theta) d\tau \quad (V.52)$$

Desconsiderando-se a influência implícita dos parâmetros estimados na matriz de regressão, é possível obter-se $\bar{\Theta}$ da condição

$$\frac{\partial}{\partial \bar{\Theta}} J(\Theta, \bar{\Theta}) = 0 \quad (\text{V.53})$$

de onde obtém-se

$$\int_0^T W^T W \bar{\Theta} d\tau = \int_0^T W^T W \Phi d\tau + \int_0^T W^T \mathcal{Y}_m d\tau \quad (\text{V.54})$$

definindo-se

$$P = \left(\int_0^T W^T W d\tau \right)^{-1} \quad (\text{V.55})$$

chega-se a expressão do vetor de parâmetros estimados

$$\bar{\Theta} = \Phi + P \int_0^T W^T \mathcal{Y}_m d\tau \quad (\text{V.56})$$

Derivando-se (V.56) e (V.55) obtém-se

$$\dot{\bar{\Theta}} = \dot{\Phi} + P \int_0^T W^T \dot{\mathcal{Y}}_m d\tau + P W^T \dot{\mathcal{Y}}_m \quad (\text{V.57})$$

e

$$\dot{P} = - \left(\int_0^T W^T W d\tau \right)^{-1} W^T W \left(\int_0^T W^T W d\tau \right)^{-1} = -P W^T W P \quad (\text{V.58})$$

que substituído em (V.57) resulta

$$\dot{\bar{\Theta}} = \dot{\Phi} - P W^T W P \int_0^T W^T \mathcal{Y}_m d\tau + P W^T \dot{\mathcal{Y}}_m \quad (\text{V.59})$$

De (V.56) tem-se

$$P \int_0^T W^T \mathcal{Y}_m d\tau = \bar{\Theta} - \Phi \quad (\text{V.60})$$

que pode ser substituído em (V.59) resultando

$$\dot{\bar{\Theta}} = \dot{\Phi} + P W^T W \Phi + P W^T (\mathcal{Y}_m - W \bar{\Theta}) \quad (\text{V.61})$$

mas, de (V.51) $\mathcal{Y}_m - W \bar{\Theta} = \alpha e$, logo

$$\dot{\Theta} = \dot{\Phi} + PW^T W \Phi + PW^T \alpha e \quad (V.62)$$

Uma vez que Θ é suposto constante, tem-se que $\dot{\Phi} = -\dot{\Theta}$, conseqüentemente

$$\dot{\Theta} = \frac{1}{2} PW^T W \Phi + \frac{1}{2} PW^T \alpha e \quad (V.63)$$

Embora a expressão (V.63) minimize a função custo, a implementação em tempo real desta lei de adaptação é comprometida, pois como pode ser visto substituindo-se (V.48) em (V.63) seria necessário o conhecimento da derivada do erro de rastreamento, pois resultaria

$$\dot{\Theta} = \frac{1}{2} PW^T (\dot{\Phi} + \alpha e) + \frac{1}{2} PW^T \alpha e \quad (V.64)$$

A solução para este problema surge aproximando-se (V.64) por uma lei de adaptação paramétrica semelhante a (V.63) (V.58), porém realizável. Para tanto, considere-se a matriz W_f , formada por cada um dos componentes de W filtrado por

$$F(s) = \frac{1}{\varepsilon s + 1} \quad (V.65)$$

sendo ε um parâmetro de projeto escolhido de forma que o filtro possua uma dinâmica bem mais rápida do que o modelo de referência.

A derivada da saída do sistema filtrada por $F(s)$ será então dada por

$$\dot{\mathcal{Y}}_f = sF(s)y = F(s)sy = F(s)(L_f h(x) + E(x)u) \quad (V.66)$$

que conforme a expressão (V.5) pode ser escrita na forma

$$\dot{\mathcal{Y}}_f = F(s)(L_f h(x) + E(x)u) - F(s)(\bar{E}_f h(x) + \bar{E}(x)u) + F(s)v \quad (V.67)$$

ou ainda

$$\dot{\mathcal{Y}}_f - F(s)v = F(s)(W\Phi) \quad (V.68)$$

como de (V.47) tem-se que $v = \dot{\mathcal{Y}}_m - \alpha e$, resulta

$$\dot{\mathcal{Y}}_f + e_f = F(s)(W\Phi) \quad (V.69)$$

somando-se $F(s)(W\bar{\Theta}) - (F(s)W)\bar{\Theta}$ em ambos os lados de (V.69) e lembrando que $\Phi = \Theta - \bar{\Theta}$ e que $F(s)$ é um operador linear, chega-se a

$$F(s)(W\Theta) - (F(s)W)\bar{\Theta} = \hat{\mathcal{E}}_f + \alpha e_f + F(s)(W\bar{\Theta}) - (F(s)W)\bar{\Theta} \quad (V.70)$$

e como Θ é suposto constante pode-se escrever

$$(F(s)W)\Theta - (F(s)W)\bar{\Theta} = \hat{\mathcal{E}}_f + \alpha e_f + F(s)(W\bar{\Theta}) - (F(s)W)\bar{\Theta} \quad (V.71)$$

que é equivalente a

$$W_f \Phi = \hat{\mathcal{E}}_f + \alpha e_f + F(s)(W\bar{\Theta}) - W_f \bar{\Theta} \quad (V.72)$$

Definindo-se a variável auxiliar

$$e_1 = M(s)(W_f \Phi) \quad (V.73)$$

resultante da equação diferencial

$$\hat{\mathcal{E}}_f + \alpha e_1 = W_f \Phi \quad (V.74)$$

semelhante à equação (V.49), correspondente ao erro de rastreamento, pode-se obter a seguinte lei de adaptação paramétrica

$$\dot{\hat{\Theta}} = \frac{1}{2} P W_f^T W_f \Phi + \frac{1}{2} P W_f^T \alpha e_1 \quad (V.75)$$

$$\dot{P} = -P W_f^T W_f P \quad (V.76)$$

que é completamente computável em tempo real, sendo as variáveis e_f e $\hat{\mathcal{E}}_f$ obtidas através da filtragem de e através de $F(s)$ e $sF(s)$, respectivamente. Note-se que estes filtros são próprios, por construção.

A prova da estabilidade em malha fechada do controle adaptativo utilizando (V.75) e (V.76) será dada no seguinte teorema

Teorema V.1

Considere um sistema não-linear do tipo MIMO, exponencialmente de fase mínima, descrito por (III.1) e (III.2), com $f(x)$, $g(x)$, $h(x)$, $L_f h(x)$ e $E(x)$ e Lipschitz contínuas. A lei de controle adaptativo

$$u = \bar{E}^{-1}(x)(v - \bar{E}_f h(x)) \quad (V.77)$$

com $\det(\bar{E}^{-1}(x)) > \delta$, com $\delta > 0$, para todo x pertencente à região onde a linearização entrada-saída é válida, com $y_m(t)$ limitado, empregada juntamente com a lei de estimação paramétrica (V.75) e (V.76), resulta na variável de saída do processo, $y(t)$, convergindo assintoticamente para a variável de saída do modelo de referência, $y_m(t)$.

■

Prova do Teorema V.1

Considere a seguinte função não negativa

$$V = \frac{1}{2} e_1^T e_1 + \frac{1}{\alpha} \Phi^T P^{-1} \Phi \geq 0 \quad \forall \alpha \in \mathfrak{R}^+ \quad (V.78)$$

sendo P^{-1} uma matriz simétrica e positiva definida. A derivada temporal de (V.78) é dada por

$$\dot{V} = e_1^T \dot{e}_1 + \frac{2}{\alpha} \Phi^T P^{-1} \dot{\Phi} + \frac{1}{\alpha} \Phi^T \dot{P}^{-1} \Phi \quad (V.79)$$

que utilizando (V.74) pode ser escrita na forma

$$\dot{V} = e_1^T (W\Phi - \alpha e_1) + \frac{2}{\alpha} \Phi^T P^{-1} \dot{\Phi} + \frac{1}{\alpha} \Phi^T \dot{P}^{-1} \Phi \quad (V.80)$$

que implica

$$\dot{V} = -\alpha e_1^T e_1 \leq 0 \quad \forall \alpha \in \mathfrak{R}^+ \quad (V.81)$$

se

$$\dot{\Phi} = -\frac{1}{2} P W_f^T W_f \Phi - \frac{1}{2} P W_f^T \alpha e_1 \quad (V.82)$$

$$\dot{P} = -P W_f^T W_f P \quad (V.83)$$

que coincide com (V.75) e (V.76). Note que de (V.55) tem-se $\dot{\mathbf{e}}_1 = \mathbf{W}_f^T \mathbf{W}_f$.

Constata-se portanto, que $V(t)$ é uma função não crescente, limitada superiormente por $V(0)$ e inferiormente por $V(\infty)$ portanto

$$V \geq \frac{1}{\alpha} \Phi^T P^{-1} \Phi \quad (\text{V.84})$$

Como P^{-1} é uma matriz simétrica e positiva definida, de acordo com Chen, citado por [52], tem-se a seguinte relação

$$\lambda_{\min}(P^{-1}) \|\Phi\|^2 \leq \Phi^T P^{-1} \Phi \leq \lambda_{\max}(P^{-1}) \|\Phi\|^2 \quad (\text{V.85})$$

Logo, de (V.84) e (V.85) obtém-se

$$V \geq \lambda_{\min}(P^{-1}) \|\Phi\|^2 \quad (\text{V.86})$$

e considerando-se que $\lambda_{\min}(P^{-1}) > 0$, conclui-se que $\Phi \in L_\infty$. De (V.78) e (V.81) deduz-se que $e_1(t)$ é uma variável limitada. Uma vez que $e_1(t)$ é a saída de um filtro estável de primeira ordem, de acordo com Narendra, citado por [52], pode-se estabelecer a seguinte relação

$$\|W(t)\| = O \left[\sup_{t \geq \tau} \|W_f(\tau)\| \right] \quad (\text{V.87})$$

de onde conclui-se que

$$W_f \Phi = O[W\Phi] \quad (\text{V.88})$$

e lembrando que $e_1(t)$ é uma variável limitada tem-se, de (V.88) e (V.49), que o erro de rastreamento $e(t)$ também é limitado. O fato da saída do modelo de referência, $y_m(t)$, ser limitada implica a limitação da variável de saída do processo $y(t)$. Como o sistema é suposto exponencialmente de fase mínima, segue-se que $x(t)$ e conseqüentemente $W(t)$ são variáveis limitadas. Portanto, pelas expressões (V.75) e (V.76) pode-se concluir que $\dot{\mathbf{e}}_1$ e $\mathbf{e}_1 \in L_\infty$.

Derivando-se novamente a expressão (V.81) tem-se que

$$\dot{\mathbf{e}}_1 = -2\alpha e_1^T (W\Phi - e_1) \in L_\infty \quad (\text{V.89})$$

e portanto, Ψ é uniformemente contínua. Logo, utilizando-se o lema de Barbalat [55] pode-se concluir que $e_1 \rightarrow 0$ com $t \rightarrow \infty$.

Novamente conforme Narendra, citado por [52], tem-se que

$$W_f(t)\Phi(t) = O\left[\sup_{t \geq \tau} |e_1(\tau)|\right] \quad (V.90)$$

de onde concluí-se que $W_f\Phi \rightarrow 0$ com $t \rightarrow \infty$ e portanto $\Theta \rightarrow 0$ com $t \rightarrow \infty$.

Da expressão (V.72) estabelece-se a seguinte relação

$$e_f(t) = O\left[\sup_{t \geq \tau} |W_f\Phi - F(s)(W\bar{\Theta}) + W_f\bar{\Theta}|\right] \quad (V.91)$$

que implica $e_f(t) \rightarrow 0$ com $t \rightarrow \infty$. Como $e_f(t)$ representa o $e(t)$ filtrado por $F(s)$, tem-se que

$$e(t) = O\left[\sup_{t \geq \tau} |e_f(\tau)|\right] \quad (V.92)$$

logo $e(t) \rightarrow 0$ com $t \rightarrow \infty$.

■

V.2.3. Resultados de Simulação

Resultados de simulação para o robô móvel IEEE-Rover são mostrados nas Figs. V-11 a V-14. O modelo de referência utilizado para ambas as direções foi

$$G_{m1}(s) = G_{m2}(s) = \frac{10}{s^2 + 1.4\sqrt{10}s + 10} \quad (V.77)$$

A trajetória espacial obtida em tempo real e em simulação para estabilizar o robô no ponto $x_c=0$, $y_c=0$ e $\theta=0$, a partir da condição inicial $x_c(0)=0m$, $y_c(0)=4m$ e $\theta(0)=0$, é mostrada na Fig. V-11. As Figs. V-12 a V-14 mostram a posição e orientação em função do tempo. Convém notar, que como o robô não pode se mover lateralmente, este movimento é considerado difícil. Observa-se também que estas curvas são bastante semelhantes às obtidas com a versão não adaptativa deste controlador, apresentadas no capítulo IV.

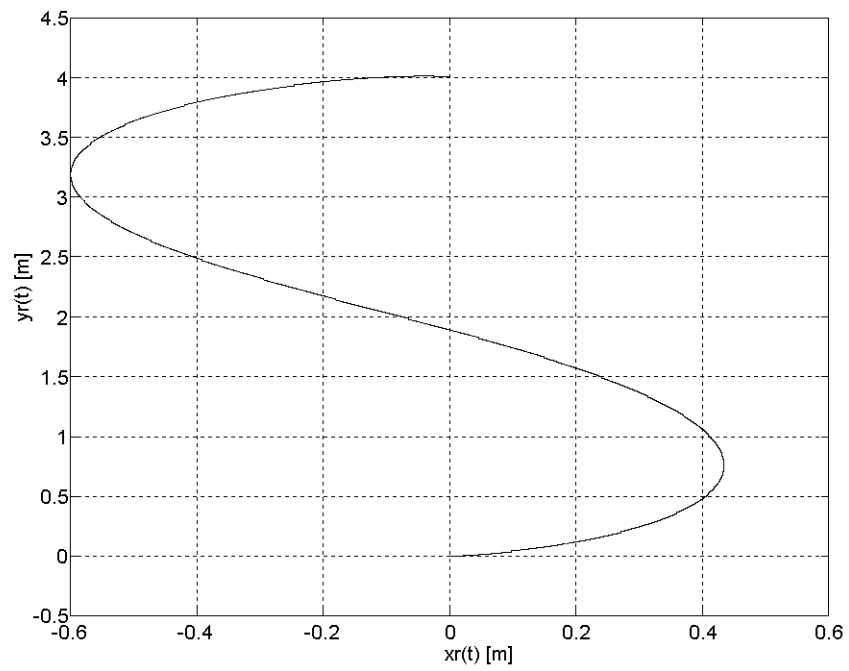


Fig. V-11 - Simulação da trajetória espacial do robô utilizando controlador por transformação descontínua adaptativo.

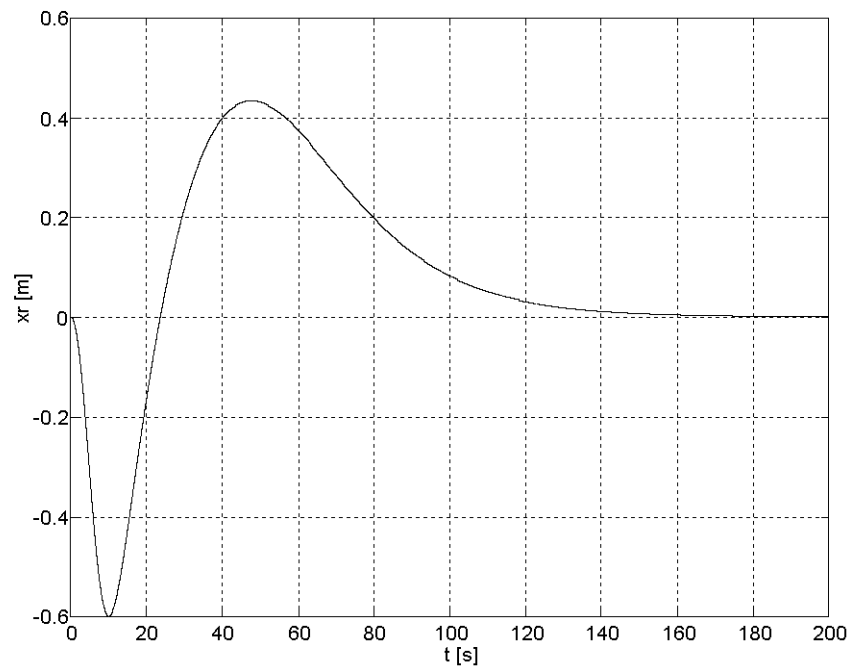


Fig. V-12 - Posição simulada do robô na direção X_0 em função do tempo.

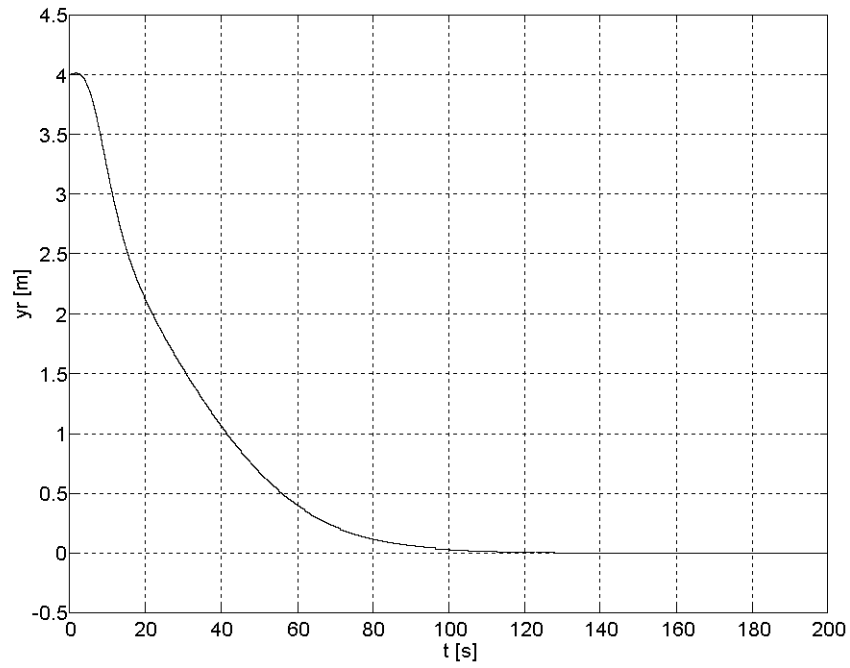


Fig. V-13 - Posição simulada do robô na direção Y_0 em função do tempo.

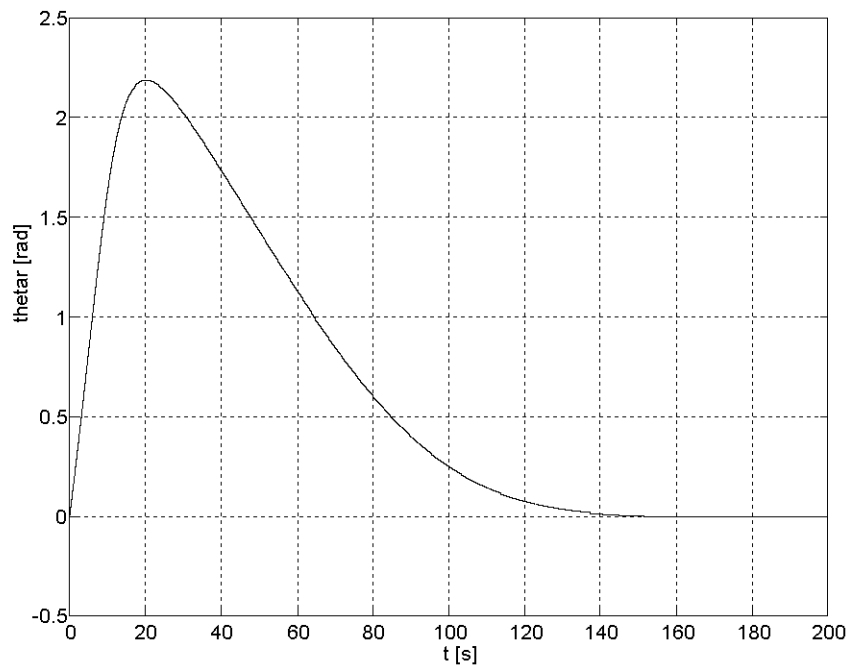


Fig. V-14 - Orientação simulada do robô na direção Y_0 em função do tempo.

Os valores iniciais dos parâmetros desconhecidos foram escolhidos como 50% superiores aos seus valores reais. A matriz de covariância foi inicializada com

$P=\text{diag}(10^6, 10^6, 10^9, 10^9)$ e condições iniciais nulas foram utilizadas em todos os filtros. O período de amostragem foi de 10ms.

Os valores finais dos parâmetros estimados são $\bar{\Theta}_1 = 69.9470$, $\bar{\Theta}_2 = 47.2626$, $\bar{\Theta}_3 = -969.3900$ e $\bar{\Theta}_4 = -1.4520e+003$, que estão bastante próximos dos valores reais: $\Theta_1 = 66.1579$, $\Theta_2 = 44.0349$, $\Theta_3 = -933.5400$ e $\Theta_4 = -1.4025e+003$. Os gráficos mostrando as evoluções destes parâmetros e das respectivas covariâncias estão nas Figs. V-15 a V-18.

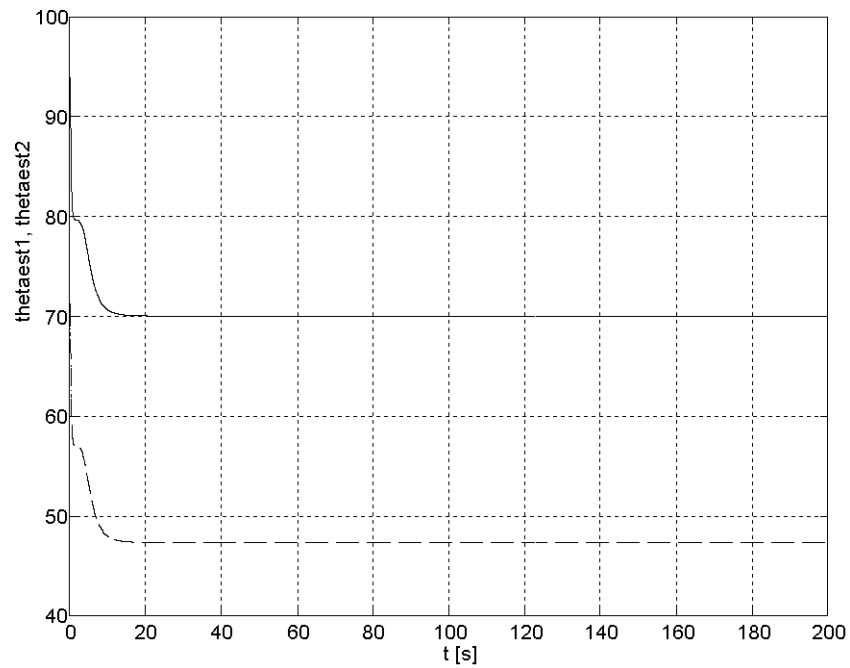


Fig. V-15 - Evolução simulada de $\bar{\Theta}_1(t)$ (sólida) e $\bar{\Theta}_2(t)$ (tracejada).

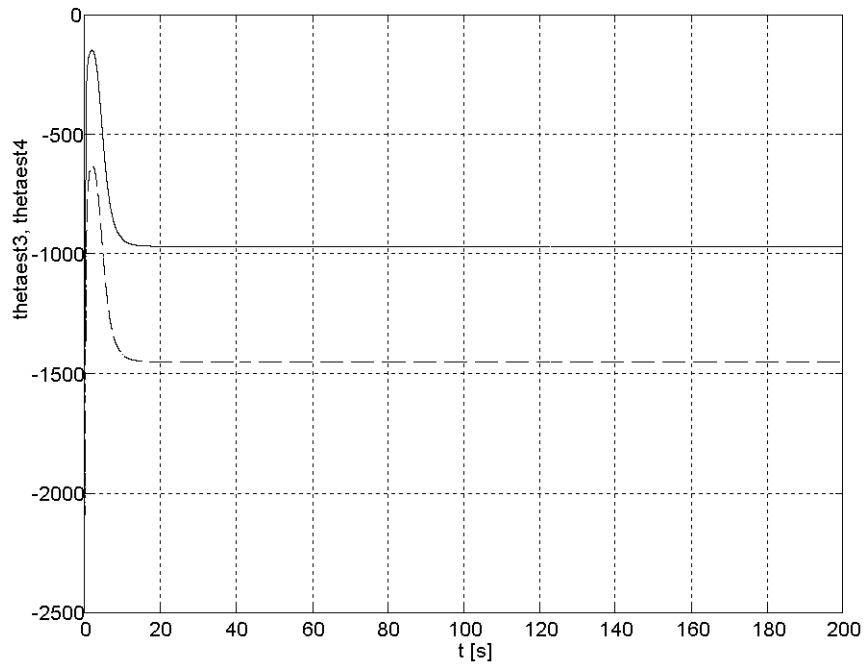


Fig. V-16 - Evolução simulada de $\bar{\Theta}_3(t)$ (sólida) e $\bar{\Theta}_4(t)$ (tracejada).

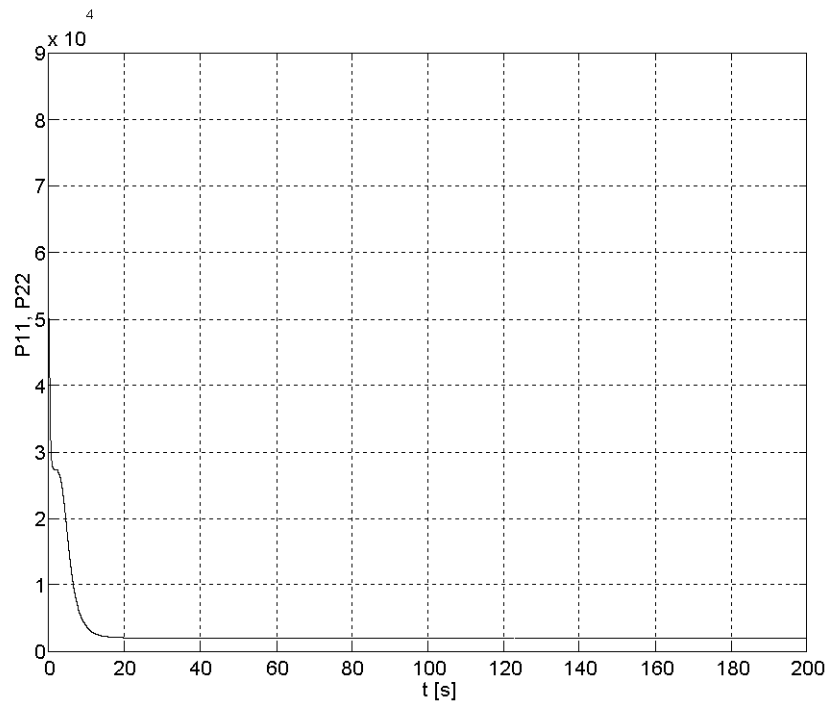


Fig. V-17 - Evolução simulada de $P_{11}(t)$ (sólida) e $P_{22}(t)$ (tracejada).

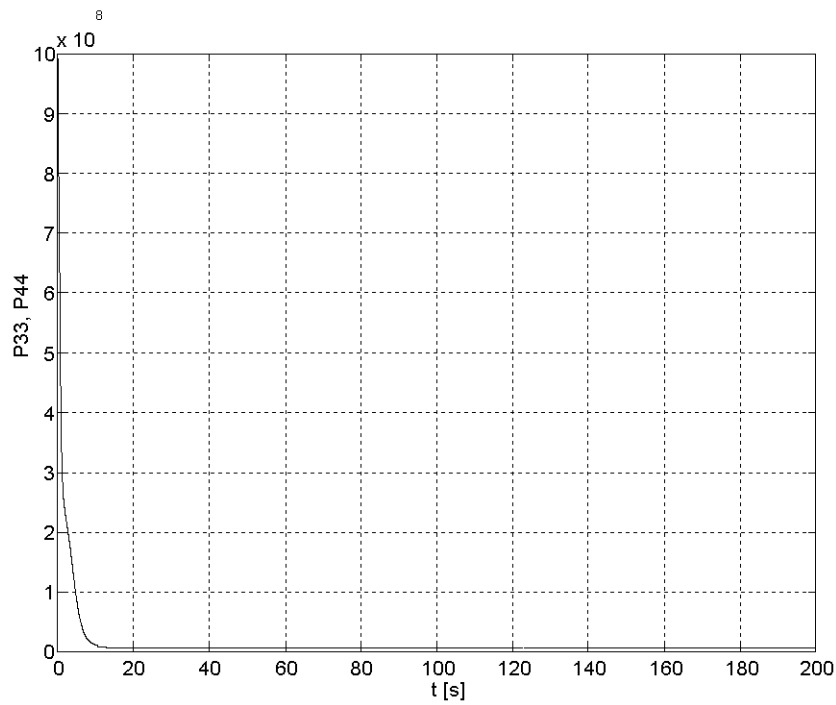


Fig. V-18 - Evolução simulada de $P_{33}(t)$ (sólida) e $P_{44}(t)$ (tracejada).

V.3. Conclusão

Neste capítulo foram desenvolvidas versões adaptativas dos controladores propostos nos capítulos III e IV. A motivação principal para utilização de controle adaptativo é a incerteza associada os parâmetros do modelo dinâmico do robô. Como esperado, o desempenho do controlador adaptativo melhora com o passar do tempo, pois os parâmetros são sintonizados. Os controladores propostos neste capítulo foram validados por simulação e implementação em tempo real.

Capítulo VI

Estimação de Posição e Orientação

Considerando-se que o objetivo principal deste trabalho é desenvolver técnicas para controle de robôs móveis, a estimação de posição e orientação é abordada aqui como forma de permitir a implementação em tempo real e simulação realística das estratégias de controle propostas. Várias técnicas estão disponíveis para a obtenção da posição e da orientação de robôs móveis, utilizando-se *encoders*, visão computacional [6][74][76], sonar [11][32][49] e scanner a laser [71]. Uma descrição bastante detalhada dos métodos para estimação de posição e orientação de robôs móveis pode ser encontrada em [10].

O método de *dead-reckoning* consiste em calcular-se o deslocamento linear e angular do robô a partir dos deslocamentos das rodas obtidos por *encoders* incrementais. É conhecido [25] que para trajetórias longas, o método de *dead-reckoning* [10] não é adequado para estimação da posição de robôs móveis, pois pela própria natureza do método, os erros vão sendo acumulados ao integrar-se os deslocamentos para obtenção da posição e orientação. Os erros presentes no método de *dead-reckoning* podem ser classificados em erros sistemáticos e erros não sistemáticos.

Os erros sistemáticos são causados basicamente por incertezas nos parâmetros geométricos utilizados para calcular a posição e orientação do robô. Para robôs atuados diferencialmente, as principais causas de erros sistemáticos são a) a diferença entre o diâmetro nominal médio e o diâmetro real das rodas; b) a diferença entre os diâmetros médios das duas rodas de tração e c) a diferença entre a distância entre rodas nominal e real.

Erros não-sistemáticos ocorrem devido a escorregamento das rodas, imperfeições no solo e fenômenos similares. Obviamente, por estarem constantemente presentes, os erros

sistemáticos são os que mais influenciam a estimativa final de posição e orientação. No entanto, através de procedimento de calibração adequados [9] é possível reduzir-se estes erros sistemáticos em até duas ordens de grandeza.

Os sensores baseados em visão computacional, por outro lado, não sofrem estes problemas. Em geral, sistemas baseados em visão operam através do reconhecimento de landmarks existentes no ambiente. Como landmarks podem ser utilizados padrões especiais como códigos de barras distribuídos pelo ambiente [44] ou características existentes naturalmente no ambiente [53].

Para robôs operando em ambientes externos, é comum a utilização de digital elevation maps (DEM) [66][67]. Um DEM é basicamente um perfil da linha do horizonte que é armazenado em um banco de dados. A localização do robô é feita comparando-se o perfil de horizonte visto pelo robô com os perfis armazenados no banco de dados.

Pode-se também utilizar visão computacional sem landmarks. Neste enfoque procura-se estimar o movimento do robô a partir do movimento das imagens captadas pela câmera [46][47][65]. No entanto, estes métodos apresentam desvantagens semelhantes as apresentadas pelo *dead-reckoning*, pois o movimento detectado pelo sistema de visão será integrado para obtenção da posição e orientação, com possível acúmulo de erros.

Outra desvantagem dos sistemas baseados em visão computacional é o tempo requerido para o processamento da imagem, o que dificulta a sua implementação em tempo real.

Dados provenientes de sonares de ultra-som podem ser processados em um tempo menor, mas a precisão de orientação não é boa [26]. Os problemas encontrados para se obter uma estimativa da posição e orientação do robô através de medidas de distância por ultra-som devem-se basicamente aos efeitos de reflexão, difração e ângulo de abertura a que estão sujeitas as ondas sonoras [39]. Um método para estimação de posição e orientação de robôs móveis é detalhado em [26]. Em geral, os métodos que utilizam sensores de ultra-som

baseiam-se no casamento de um mapa de ultra-som armazenado no robô com a resposta obtida pelos sensores.

As dificuldades inerentes ao uso de sensores de ultra-som podem ser evitadas utilizando-se medidas de distância obtidas a partir de um scanner a laser, como em [71]. Basicamente, os mesmos métodos utilizados com sonares podem ser utilizados com scanner a laser, tendo-se a vantagem de que os efeitos de reflexão, difração e ângulo de abertura do feixe são minimizados. No entanto, o custo de um scanner a laser é bem superior ao de um sistema de sonares. Além disso, apresenta inconvenientes para a montagem sobre o robô, pois não pode ser montado distribuído em torno do mesmo, como ocorre com um *array* de sonares.

Com o desenvolvimento de plataformas inerciais de estado sólido de baixo custo, este tipo de sensor também foi proposto para robôs móveis [5][40]. Como este tipo de sensor fornece informação de aceleração linear e angular, torna-se necessário uma dupla integração para obter-se posição e orientação do robô. Portanto, este também é um método de *dead-reckoning*. No entanto, ao contrário dos métodos utilizando *encoders*, o conhecimento dos parâmetros geométricos do robô não é necessário, pois o sensor fornece diretamente a aceleração linear e angular. Por outro lado, devido à necessidade do sinal de saída do sensor ser integrado duas vezes, o efeito do acúmulo dos erros torna-se mais significativo. Conseqüentemente, o uso deste tipo de sensor requer uma modelagem bastante cuidadosa dos erros.

Embora sistemas GPS (*Global Positioning System*) não sejam adequados para robôs móveis que devam operar em ambientes internos, estes sistemas quando operando de forma diferencial podem fornecer a posição do robô com uma precisão razoável, o que o torna uma boa alternativa para uso em ambientes externos. Recentemente tem sido realizados esforços no sentido de desenvolver sistemas LPS (*Local Positioning System*) [75]. Contrariamente ao

GPS, onde cada receptor determina sua posição com relação a um referencial fixo, nos sistemas LPS sendo desenvolvidos a posição de cada *tag* (dispositivos sendo rastreados) é determinada por um agente central. Portanto, para utilização deste sistema em robôs móveis seria necessário a existência de comunicação entre este agente central e o robô. Isto representa sérias limitações à autonomia do robô. Tanto devido à possibilidade de perda da comunicação quanto à complexidade introduzida caso hajam vários robôs operando na mesma área. No entanto, é razoável imaginar-se que dentro de algum tempo surgirão sistemas LPS operando de forma similar ao GPS, ou seja, de forma que os receptores possam determinar a sua própria posição.

Sabe-se que uma parcela significativa dos erros em sistemas de *dead-reckoning* deve-se ao erro de orientação [10]. Para reduzir os efeitos do erro de orientação na estimativa final de posição obtida por *dead-reckoning*, alguns trabalhos [40][60] utilizam uma bússola digital. A idéia é obter-se uma estimativa mais confiável para a orientação através da fusão de dados dos encoders e da bússola. Esta fusão de dados é realizada utilizando-se um filtro de Kalman estendido, devido as não-linearidades presentes no modelo cinemático do robô.

Na realidade, a fusão de dados de sensores operando sob princípios diferentes utilizando o filtro de Kalman estendido é um procedimento usual para a estimação de posição e orientação de robôs móveis [26][30]. Como é percebido nos parágrafos anteriores, os sensores de posição e orientação podem ser classificados em dois tipos: sensores absolutos e sensores incrementais.

Nos sensores absolutos os erros não se propagam ao longo do tempo, mas estes sensores requerem um processamento sofisticado para obter-se uma estimativa. Esta complexidade de processamento causa dois problemas principais: a) como o robô está em movimento, ao término do processamento a posição e orientação do robô sofreram alterações, portanto a estimativa obtida está defasada e b) normalmente o tempo necessário para realizar o

processamento é bem maior do que o período de amostragem do sistema de controle. Sensores incrementais, por outro lado, em geral são processados rapidamente, mas apresentam erros crescentes pois a posição e orientação são obtidas por integração.

A fusão de dados de sensores absolutos e de sensores incrementais permite se obter estimativas de posição e orientação a intervalos de tempos compatíveis com as necessidades do sistema de controle, sem que no entanto os erros cresçam de forma ilimitada. Basicamente, o efeito resultante é de que os erros dos sensores incrementais são zerados cada vez que uma leitura dos sensores absolutos está disponível.

Considerando-se que este trabalho enfoca o controle de robôs móveis com objetivo de convergência para um ponto, as trajetórias percorridas pelo robô serão tipicamente curtas. Se os parâmetros geométricos do robô forem calibrados convenientemente, os erros de *dead-reckoning* acumulados durante as trajetórias de teste serão desprezíveis. Além disso, para avaliação do desempenho do controlador, o que importa são a posição e orientação medidas pelo sensor, e que deverão convergir para os valores desejados. Se os valores medidos pelo sensor correspondem ou não aos valores reais, é uma questão que está além do escopo do problema de controle. Portanto, o método utilizado para determinação da posição e orientação do robô utilizado é basicamente *dead-reckoning*. No entanto, visando avaliar os problemas de integração das técnicas baseadas na fusão de dados utilizando o filtro de Kalman com os métodos de controle propostos, foram também implementados métodos de estimação baseados na fusão, dos dados dos *encoders* com um sistema de visão computacional e com uma bússola digital.

Neste tipo de problema é comum considerar-se que todos os sensores envolvidos possuem a mesma taxa de amostragem. Isto é, assume-se que todos os sensores são capazes de fornecer medidas periodicamente e que este período é constante e igual para todos os sensores.

No entanto, como discutido em detalhes no capítulo VII, quando são utilizados sensores com diferentes princípios físicos, é comum a situação onde cada sensor exige uma taxa de amostragem diferente. Nestes casos, a simulação do sistema, ou uma implementação da fusão de dados off-line não é muito diferente do caso com uma única taxa de amostragem. Porém, uma implementação para fusão dos dados em tempo real apresenta uma série de problemas relacionados ao processamento para aquisição de dados de cada sensor, com o sincronismo entre os sensores e destes com o filtro de Kalman.

VI.1. Método de *Dead-Reckoning*

Uma das técnicas mais simples para estimação de posição (e orientação) consiste em contar-se a quantidade de pulsos obtidos a partir dos *encoders* de cada roda em um certo intervalo de tempo. Assumindo-se que não há escorregamento e conhecendo-se o raio das rodas (r) e o número de pulsos por volta dos *encoders* (Π), pode-se calcular o deslocamento linear das rodas no intervalo de tempo

$$\Delta D_r(k) = \frac{2\pi r NP_r(k)}{\Pi} \quad (\text{VI.1})$$

$$\Delta D_l(k) = \frac{2\pi r NP_l(k)}{\Pi} \quad (\text{VI.2})$$

onde

$\Delta D_r(k), \Delta D_l(k)$ = deslocamento linear das rodas direita e esquerda no intervalo de tempo kT entre e $(k+1)T$;

$NP_r(k), NP_l(k)$ = número de pulsos obtidos nos *encoders* das rodas direita e esquerda no intervalo de tempo entre kT entre e $(k+1)T$.

Os deslocamentos linear e do robô considerando-se como referência o ponto no centro axial das rodas podem ser calculados por

$$\Delta D(k) = \frac{\Delta D_r(k) + \Delta D_l(k)}{2} \quad (\text{VI.3})$$

$$\Delta \theta(k) = \frac{\Delta D_r(k) - \Delta D_l(k)}{2b} \quad (\text{VI.4})$$

onde $2b$ é a distância axial entre as rodas.

Assumindo-se que a trajetória percorrida pelo veículo é um arco de circunferência, como mostrado na Fig. VI-1, de forma semelhante à [44] obtém-se

$$x_c(k+1) = x_c(k) + \Delta s(k) \cos\left(\theta(k) + \frac{\Delta \theta(k)}{2}\right) + d \Delta \theta(k) \sin\left(\theta(k) + \frac{\Delta \theta(k)}{2}\right) \quad (\text{VI.5})$$

$$y_c(k+1) = y_c(k) + \Delta s(k) \sin\left(\theta(k) + \frac{\Delta \theta(k)}{2}\right) - d \Delta \theta(k) \cos\left(\theta(k) + \frac{\Delta \theta(k)}{2}\right) \quad (\text{VI.6})$$

$$\theta(k+1) = \theta(k) + \Delta \theta(k) \quad (\text{VI.7})$$

onde d é a distância do centro das rodas ao centro de massa e

$$\Delta s(k) = \Delta D(k) \frac{\sin\left(\frac{\Delta \theta(k)}{2}\right)}{\frac{\Delta \theta(k)}{2}} \quad (\text{VI.8})$$

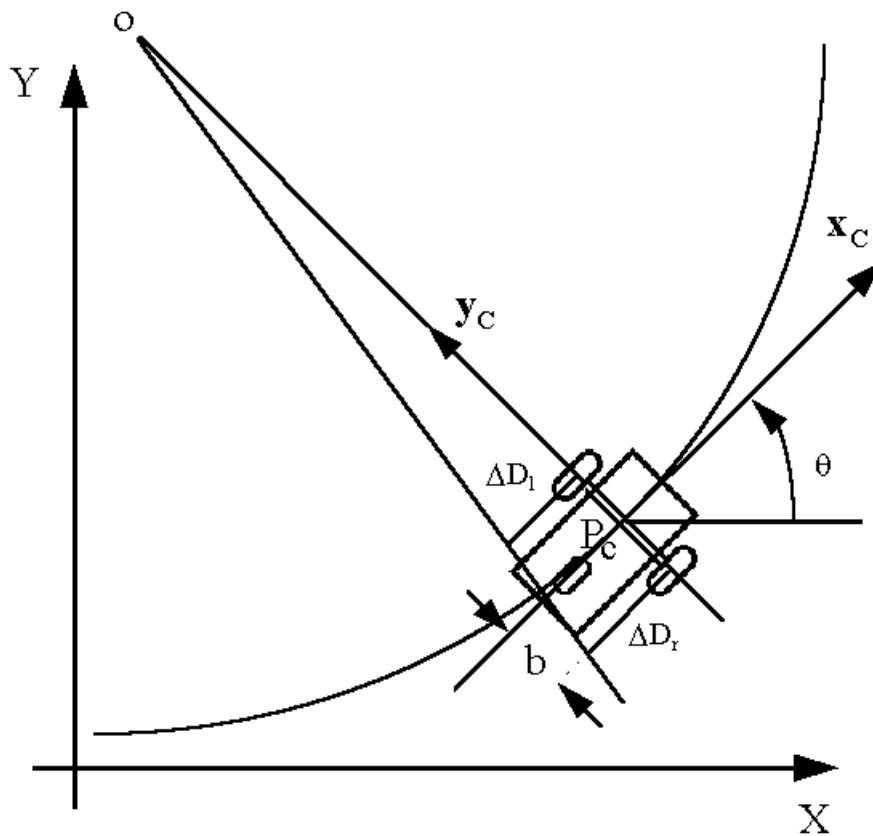


Fig. VI-1 - Trajetória circular utilizada para desenvolvimento do algoritmo de dead-reckoning.

É importante ressaltar que quando $\Delta\theta=0$, a trajetória percorrida pela plataforma degenera para uma reta e portanto $\Delta s(k)=\Delta D(k)$. Isto é refletido na expressão (VI.8) pelo fato de que $\text{sen}(x)/x=1$ quando x tende a zero, não havendo portanto, indeterminação matemática. Ou seja, As expressões (VI.5) a (VI.7) foram desenvolvidas considerando-se uma trajetória circular, mas são também exatas quando a trajetória é uma reta. Existem, no entanto, erros de modelagem devido ao escorregamento das rodas, às irregularidades do solo, à não exatidão no conhecimento do raio das rodas e da distância entre elas, etc., que são estudados com maior detalhe em [9].

Na Fig. VI-2 apresenta-se uma simulação da trajetória realizada pelo robô IEEE-Rover utilizando-se o controlador proposto no capítulo III e a trajetória estimada por *dead-reckoning*.

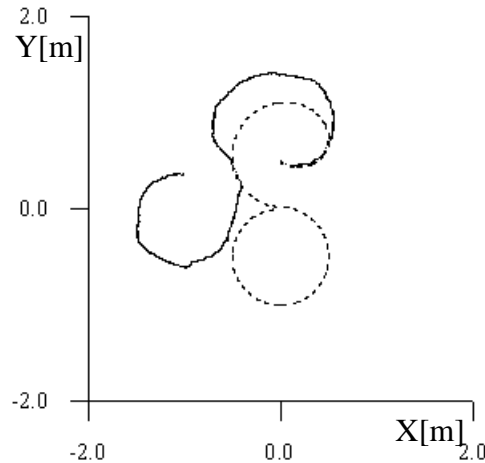


Fig. VI-2 - Trajetória do robô (pontilhada) e trajetória estimada por dead-reckoning (sólida).

Observa-se que a trajetória estimada por *dead-reckoning* coincide com a trajetória realizada apenas nos instantes iniciais. Este comportamento deve-se ao efeito integral intrínseco à esta técnica de estimação. Como as estimativas são obtidas através de somatórios, pode-se perceber que estes erros vão se acumulando. Isto faz com que esta estratégia de estimação do vetor de estado seja adequada apenas para trajetos pequenos, quando o erro acumulado pode ser desprezado.

VI.2. Visão Computacional

A estimação de posição e orientação através de visão computacional consiste em obter-se uma ou mais imagens que contenham pontos conhecidos (posição e orientação conhecidas) do ambiente. Estes pontos são identificados na imagem e conhecendo-se os parâmetros do sistema de aquisição de imagens, a posição do robô em relação à eles pode ser computada. Como a posição e orientação dos pontos de referência em relação ao sistema de coordenadas global são conhecidas, calcula-se por transformações de coordenadas a posição e orientação do robô.

O sistema de processamento de imagens utilizado neste trabalho é baseado em *landmarks* como os apresentados na Fig. VI-3, que são fixados nas paredes do ambiente de

trabalho da plataforma. A escolha deste tipo de *landmark* foi motivada por [13], pois simplifica a sua localização na imagem.

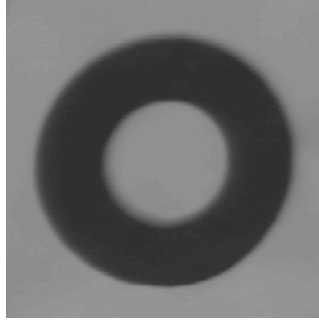


Fig. VI-3 - *Landmark utilizado pelo sistema de visão.*

Ao serem vistos em perspectiva pela plataforma, estes *landmarks* sofrem distorções nos eixos vertical e horizontal. Como os *landmarks* são fixados à altura da câmera, a distorção no eixo vertical é devida apenas à distância da câmera ao *landmark*, enquanto a distorção no eixo horizontal se dá devido à distância e à orientação da câmera com relação ao *landmark*. No caso geral, o *landmark* aparecerá na imagem como uma elipse. A posição e os eixos desta elipse são detectados na imagem através da transformada de Hough generalizada [4][78].

A posição e orientação do *landmark* em relação à câmera então será dada pela transformação de perspectiva [27]

$$Z = \lambda \left(1 + \frac{r_l}{a_y} \right) \quad (\text{VI.9})$$

$$\Theta = \pm a \cos \left(\frac{a_x}{a_y} \right) \quad (\text{VI.10})$$

onde Z é distância da câmera ao *landmark*, λ é a distância focal da câmera, r_l é raio do *landmark*, a_x e a_y são respectivamente os semi-eixos horizontal e vertical da elipse e Θ é a orientação da câmera com relação ao *landmark*.

A seguir, determina-se a matriz de transformação homogênea [20] entre o *landmark* e a câmera

$${}^vT_1 = \begin{bmatrix} -\text{sen}\Theta & \cos\Theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \cos\Theta & \text{sen}\Theta & 0 & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{VI.11})$$

A matriz de transformação homogênea entre o sistema de coordenadas global e o sistema de coordenadas centrado em x_c e y_c será dada por

$${}^gT_c = {}^gT_1 \cdot {}^1T_v \cdot {}^vT_c \quad (\text{VI.12})$$

onde gT_1 é a matriz de transformação homogênea entre o sistema de coordenadas localizado no centro do *landmark* e o sistema de coordenadas global, 1T_v é a matriz de transformação homogênea entre a câmera e o *landmark* e vT_c é a matriz de transformação homogênea entre sistema de coordenadas centrado em x_c e y_c e a câmera.

Como gT_c é constante pois depende somente da posição do *landmark* sendo observado, e vT_c depende apenas da geometria da plataforma e ${}^1T_v = ({}^vT_1)^{-1}$ tem-se que

$${}^gT_c = {}^gT_1 \cdot ({}^vT_1)^{-1} \cdot {}^vT_c \quad (\text{VI.13})$$

Por outro lado, da Fig. II- pode-se facilmente obter

$${}^gT_c = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 & x_c \\ \text{sen}\theta & \cos\theta & 0 & y_c \\ 0 & 0 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{VI.14})$$

onde d_1 é a altura do ponto x_c, y_c . Portanto, tem-se que

$$x_c = ({}^gT_c)_{14} \quad (\text{VI.15})$$

$$y_c = ({}^gT_c)_{24} \quad (\text{VI.16})$$

e

$$\theta = \arctan 2 \left(({}^gT_c)_{11}, ({}^gT_c)_{21} \right) \quad (\text{VI.17})$$

É importante observar, que devido à ambigüidade do sinal de Θ em (VI.10), serão obtidos dois conjuntos de valores x_c , y_c e θ para um mesmo *landmark*. Esta ambigüidade pode ser removida observando-se dois ou mais *landmarks*, veja-se [13][44] para maiores detalhes.

Utilizando-se as expressões (VI.15)-(VI.17), pode-se reconstituir a posição e orientação do robô. O erro de estimação de x_c , y_c e θ neste caso não é cumulativo como no caso anterior, mas o tempo necessário para estimação via processamento de imagens é maior, forçando o sistema de controle a usar uma estimativa não muito atual do estado ou uma taxa de amostragem muito baixa.

VI.2.1. Fusão de Dados

Como o método de *dead-reckoning* e o processamento de imagens possuem características diferentes parece razoável supor-se que a fusão dos dados obtidos pelos dois sistemas produzirá uma estimativa de posição e orientação de melhor qualidade. Esta fusão de dados pode ser feita utilizando-se o filtro de Kalman estendido.

Definindo-se

$$U(k) = \begin{bmatrix} \Delta D_r(k) \\ \Delta D_l(k) \end{bmatrix} \quad (VI.18)$$

e

$$X(k) = \begin{bmatrix} x_c(k) \\ y_c(k) \\ \theta(k) \end{bmatrix} \quad (VI.19)$$

pode-se escrever (VI.5) - (VI.7) na forma

$$X(k+1) = f(X(k), U(k) + \omega(k)) \quad (VI.20)$$

onde $\omega(k)$ representa perturbações como escorregamento das rodas, incertezas nos parâmetros do modelo e erros nos *encoders*.

A expressão (VI.20), representa a posição e orientação reais do robô, pois as perturbações foram levadas em consideração através de $\omega(k)$ enquanto as expressões (VI.5)-(VI.7) representam a posição do veículo sem considerar as perturbações, portanto incluem erro.

A saída do sistema de processamento de imagens, dada pelas expressões (VI.15) - (VI.17) é o vetor

$$\mathbf{y}(k) = \begin{bmatrix} x_c \\ y_c \\ \theta \end{bmatrix} + \mathbf{v}(k) \quad (\text{VI.21})$$

considerado como uma observação do processo (VI.20) que contém um erro de observação

$$\mathbf{v}(k) = \begin{bmatrix} \text{erro}_{x_c} \\ \text{erro}_{y_c} \\ \text{erro}_{\phi} \end{bmatrix} \quad (\text{VI.22})$$

denominado ruído de observação.

Fazendo-se as suposições usuais, isto é, $\omega(k)$ e $\mathbf{v}(k)$ são ruídos gaussianos com média zero e $E \begin{bmatrix} \omega(k) \\ \mathbf{v}(k) \end{bmatrix} \begin{bmatrix} \omega(n)^T & \mathbf{v}(n)^T \end{bmatrix} = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} \delta(k-n)$, com $Q \geq 0$ e $R > 0$, $E[\omega(k)X(n)^T] = 0$, para $k > n$, e $E[\mathbf{v}(k)X(n)^T] = 0$, para $k > n$, o filtro de Kalman estendido pode ser utilizado para combinar as estimativas de posição obtidas pelos sistemas de *dead-reckoning* e processamento de imagens.

Expandindo-se $f(\cdot)$ em (VI.20) em série de Taylor em torno do ponto $\bar{\mathbf{X}}(k)$, $U(k)$ onde $\bar{\mathbf{X}}(k)$ é a estimativa de $X(k)$ no instante k e desprezando os termos de ordem superior, resulta

$$f(X(k), U(k) + \omega(k)) = f(\bar{\mathbf{X}}(k), U(k)) + F_k(X(k) - \bar{\mathbf{X}}(k)) + G_k \omega(k) \quad (\text{VI.23})$$

onde

$$F_k = \frac{\partial f(X(k), U(k))}{\partial X(k)} \bigg|_{X(k)=\bar{X}(k)} = \begin{bmatrix} 1 & 0 & F_{13} \\ 0 & 1 & F_{23} \\ 0 & 0 & 1 \end{bmatrix} \quad (VI.24)$$

e

$$G_k = \frac{\partial f(X(k), U(k))}{\partial U(k)} \bigg|_{X(k)=\bar{X}(k)} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \\ G_{31} & G_{32} \end{bmatrix} \quad (VI.25)$$

são dados por

$$F_{13} = d\Delta\theta(k)C - \Delta sS c \quad (VI.26)$$

$$F_{23} = d\Delta\theta(k)S + \Delta sC \quad (VI.27)$$

$$G_{11} = N(cC - sS) - MU_2(k)Cs + \frac{d}{2b} \left(S + \frac{\Delta\theta(k)}{2} C \right) \quad (VI.28)$$

$$G_{12} = -N(cC - sS) + MU_1(k)Cs - \frac{d}{2b} \left(S + \frac{\Delta\theta(k)}{2} C \right) \quad (VI.29)$$

$$G_{21} = N(Cs + cS) - MU_2(k)sS + \frac{d}{2b} \left(-C + \frac{\Delta\theta(k)}{2} S \right) \quad (VI.30)$$

$$G_{22} = -N(Cs + cS) + MU_1(k)sS - \frac{d}{2b} \left(-C + \frac{\Delta\theta(k)}{2} S \right) \quad (VI.31)$$

$$G_{31} = \frac{1}{2b} \quad (VI.32)$$

$$G_{32} = -\frac{1}{2b} \quad (VI.33)$$

sendo

$$s = \sin\left(\frac{\Delta\theta(k)}{2}\right) \quad (VI.34)$$

$$c = \cos\left(\frac{\Delta\theta(k)}{2}\right) \quad (VI.35)$$

$$C = \cos\left(\theta(k) + \frac{\Delta\theta(k)}{2}\right) \quad (VI.36)$$

$$S = \sin\left(\theta(k) + \frac{\Delta\theta(k)}{2}\right) \quad (\text{VI.37})$$

$$M = \frac{4b}{U_1(k) - U_2(k)} \quad (\text{VI.38})$$

$$N = \frac{U_1(k) + U_2(k)}{2(U_1(k) - U_2(k))} \quad (\text{VI.39})$$

quando $\Delta\theta=0$ as expressões (VI.26)-(VI.31) reduzem-se a

$$F_{13} = -\Delta D \sin(\theta(k)) \quad (\text{VI.40})$$

$$F_{23} = \Delta D \cos(\theta(k)) \quad (\text{VI.41})$$

$$G_{11} = \frac{1}{2} \left(\cos(\theta(k)) - \frac{\Delta D}{2b} \sin(\theta(k)) \right) + \frac{d}{2b} \sin(\theta(k)) \quad (\text{VI.42})$$

$$G_{12} = \frac{1}{2} \left(\cos(\theta(k)) + \frac{\Delta D}{2b} \sin(\theta(k)) \right) - \frac{d}{2b} \sin(\theta(k)) \quad (\text{VI.43})$$

$$G_{21} = \frac{1}{2} \left(\sin(\theta(k)) + \frac{\Delta D}{2b} \cos(\theta(k)) \right) - \frac{d}{2b} \cos(\theta(k)) \quad (\text{VI.44})$$

$$G_{22} = \frac{1}{2} \left(\sin(\theta(k)) - \frac{\Delta D}{2b} \cos(\theta(k)) \right) + \frac{d}{2b} \cos(\theta(k)) \quad (\text{VI.45})$$

substituindo-se (VI.24) e (VI.25) em (VI.20)

$$\mathbf{X}(k+1) = \mathbf{F}_k \mathbf{X}(k) + \mathbf{G}_k \omega(k) + f(\bar{\mathbf{X}}(k), U(k)) - \mathbf{F}_k \bar{\mathbf{X}}(k) \quad (\text{VI.46})$$

que juntamente com (VI.21) forma um sistema linear variante no tempo com entrada $f(\bar{\mathbf{X}}(k), U(k)) - \mathbf{F}_k \bar{\mathbf{X}}(k)$, que pode ser perfeitamente determinada no instante k e portanto é uma entrada determinística. Portanto pode-se utilizar o seguinte algoritmo para o filtro de Kalman estendido [23]

$$\mathbf{K}(k) = (\mathbf{F}_k \mathbf{P}(k) \mathbf{F}_k^T + \mathbf{G}_k \mathbf{S}) (\mathbf{P}(k) + \mathbf{R})^{-1} \quad (\text{VI.47})$$

$$\bar{\mathbf{X}}(k+1) = f(\bar{\mathbf{X}}(k), U(k)) + \mathbf{K}(k) (y(k+1) - \bar{\mathbf{X}}(k)) \quad (\text{VI.48})$$

$$\mathbf{P}(k+1) = \mathbf{F}_k \mathbf{P}(k) \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^T - \mathbf{K}(k) (\mathbf{P}(k) + \mathbf{R}) \mathbf{K}(k)^T \quad (\text{VI.49})$$

com condições iniciais $P(0)$ e $\bar{x}(0)$

Na Fig. VI.4 pode-se ver um diagrama de blocos do método de estimação de posição e orientação utilizando fusão de dados através do filtro de Kalman estendido.

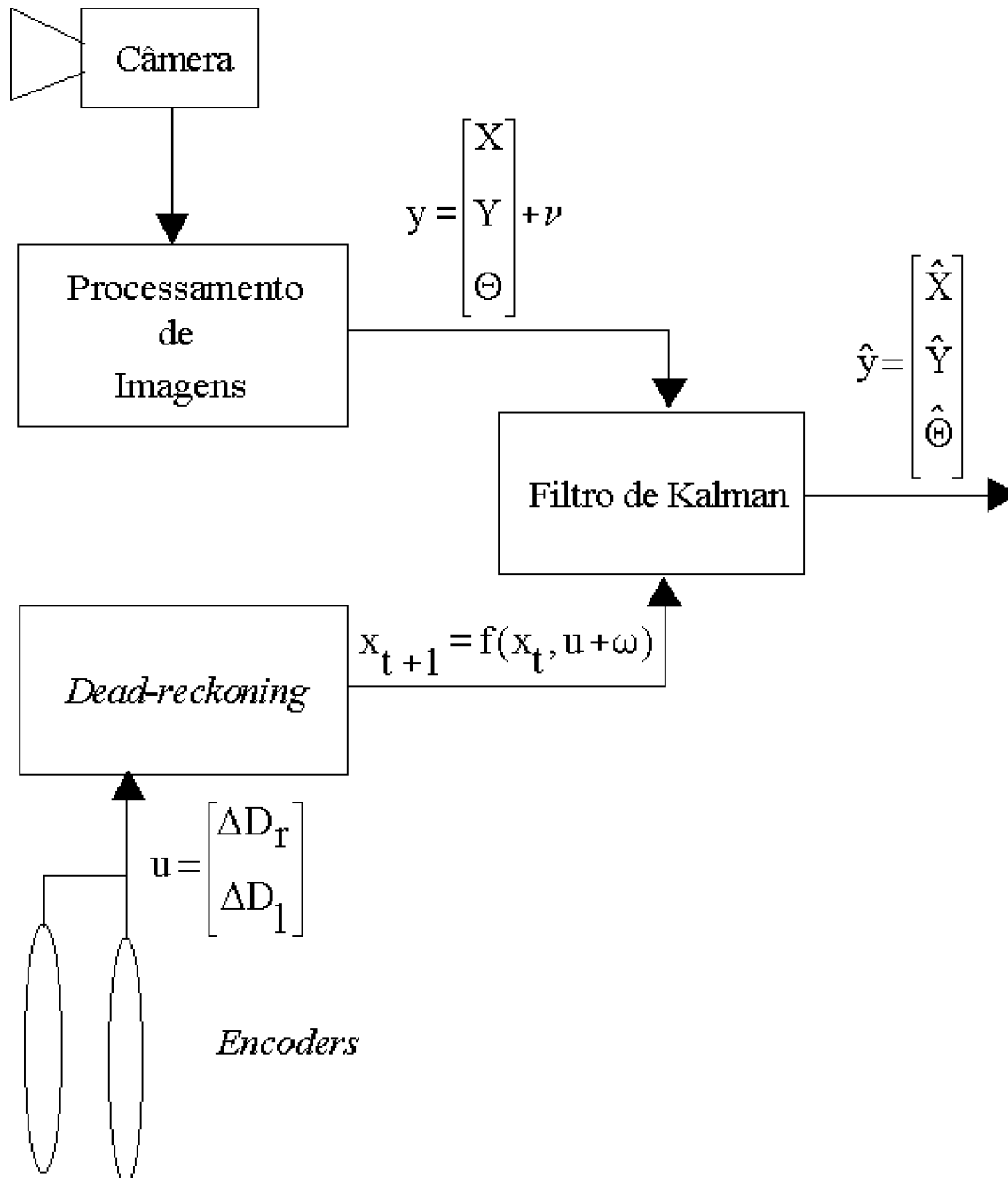


Fig. VI-4 - Diagrama de blocos da fusão de dados obtidos por visão computacional e encoders.

VI.2.2. Resultados de Simulação

A Fig. VI-5 mostra a trajetória descrita pelo robô IEEE-Rover utilizando-se o controlador desenvolvido no capítulo III com a posição e orientação estimadas através de dados obtidos por visão computacional e *dead-reckoning*, fundidos através do filtro de Kalman estendido.

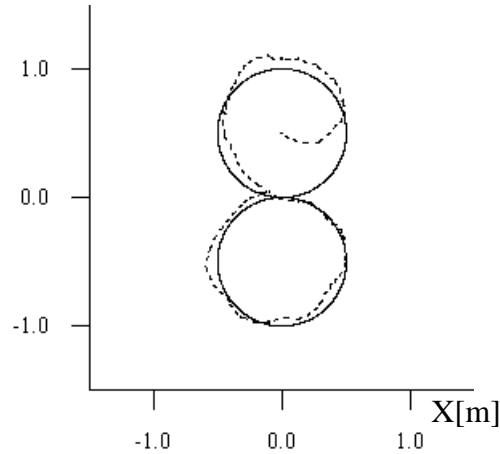


Fig. VI-5 - Trajetória de referência (sólida) e trajetória estimada por fusão de dados (tracejada).

VI.3. Bússola Digital

O cálculo de orientação baseado na leitura de uma bússola digital $\theta_b(k)$ apresenta erros, ou seja, o valor verdadeiro $\theta_t(k)$ é corrompido por o ruído de medida $v(k)$, isto é,

$$\theta_b(k) = \theta_t(k) + v(k) \quad (\text{VI.50})$$

Neste experimento foi utilizada a bússola digital *Vector 2X Compass Module* fabricada pela *Precision Navigation, Inc.*, cujas especificações estão em [56].

A integração de ambos os sistemas, odômetro e bússola, foi realizada mediante um filtro de Kalman linear. A orientação calculada com base nos *encoders*, $\theta(k)$, pode ser modelada como o valor verdadeiro $\theta_t(k)$ mais um erro de orientação $\delta\theta(k)$, isto é,

$$\theta(k) = \theta_t(k) + \delta\theta(k) \quad (\text{VI.51})$$

O processo de observação $y(k)$, é a diferença de orientação entre ambos os sistemas, (VI.51) e (VI.50), portanto,

$$y(k) = \delta\theta(k) + v(k) \quad (\text{VI.52})$$

Supondo-se que um ruído branco $w(k)$ é adicionado ao incremento de orientação a cada instante de tempo, a orientação calculada pelos *encoders* no instante $k+1$ será dada por

$$\theta(k+1) = \theta(k) + \Delta\theta_t(k) + w(k) \quad (\text{VI.53})$$

onde $\Delta\theta_t(k)$ é o valor verdadeiro do incremento de orientação.

De (VI.7), assumindo-se que as magnitudes são verdadeiras, tem-se

$$\theta_t(k+1) = \theta_t(k) + \Delta\theta_t(k) \quad (\text{VI.54})$$

subtraindo-se (V.53) de (V.54) e utilizando-se (VI.51) obtém-se

$$\delta\theta(k+1) = \delta\theta(k) + w(k) \quad (\text{VI.55})$$

que constitui a equação de estado para o filtro de Kalman.

Assim, o filtro de Kalman linear é escalar com os ruídos de medida, $v(k)$, e de estado, $w(k)$, não-correlacionados, permitindo que se estime o erro de orientação, para a seguir subtraí-lo do valor calculado a partir dos *encoders*, como mostrado na Fig. VI-6. A orientação corrigida é realimentada para calcular a posição, de modo a se reduzir o acúmulo de erro.

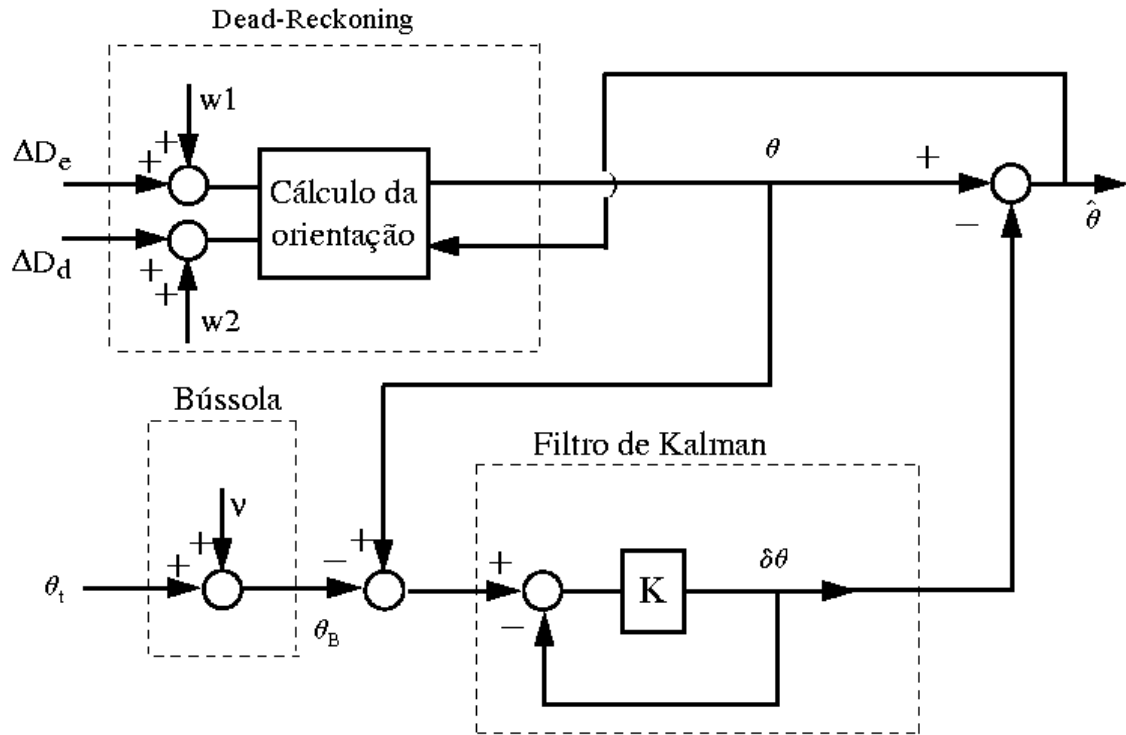


Fig. VI-6 - Diagrama de Blocos da Fusão de Dados da Bússola Digital e Encoders.

VI.3.1. Resultados de Simulação

A estimativa de posição e orientação através da fusão de dados de *dead-reckoning* e de uma bússola digital via filtro de Kalman foi simulada para uma trajetória circular com 1.37 m de raio. A Fig. VI-7 mostra a trajetória determinada por *dead-reckoning* (odômetro) e a trajetória estimada utilizando-se o filtro de Kalman, para uma velocidade constante do veículo de 0.14 m/s. A Fig. VI-8 mostra as diferentes orientações durante o movimento. Utilizou-se um período de amostragem de 50ms para os encoders e de 200 ms para a bússola. Um ruído de medida com desvio padrão de $4^\circ/s$ foi utilizado, e o desvio padrão do ruído em cada encoder é de 10% do valor máximo medido.

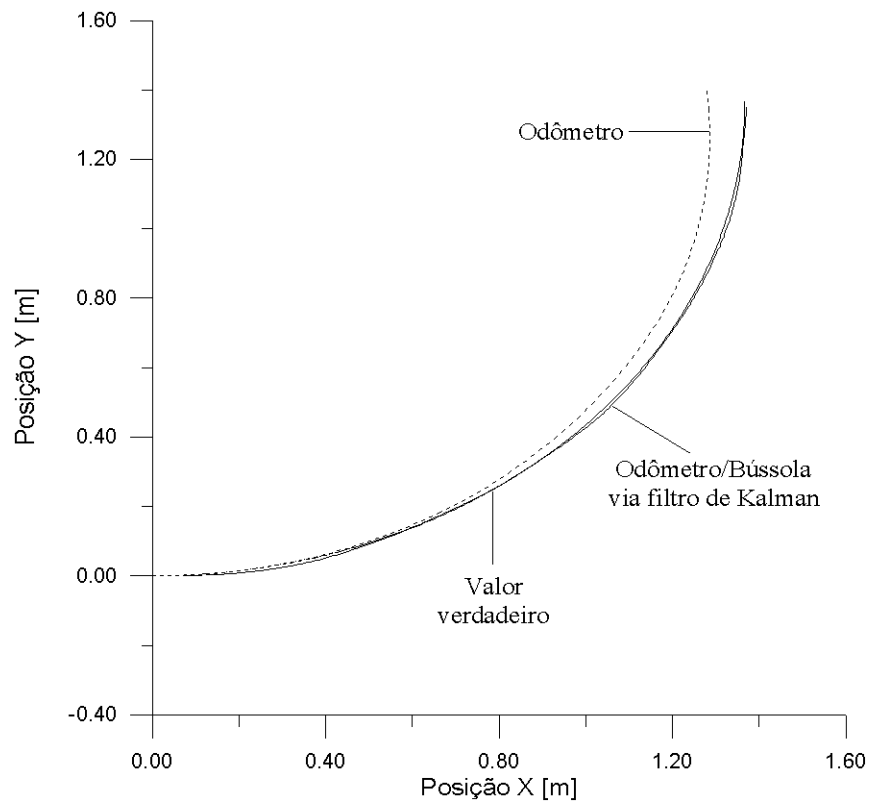


Fig. VI-7 - Simulação do odômetro e da fusão de dados odômetro/bússola.

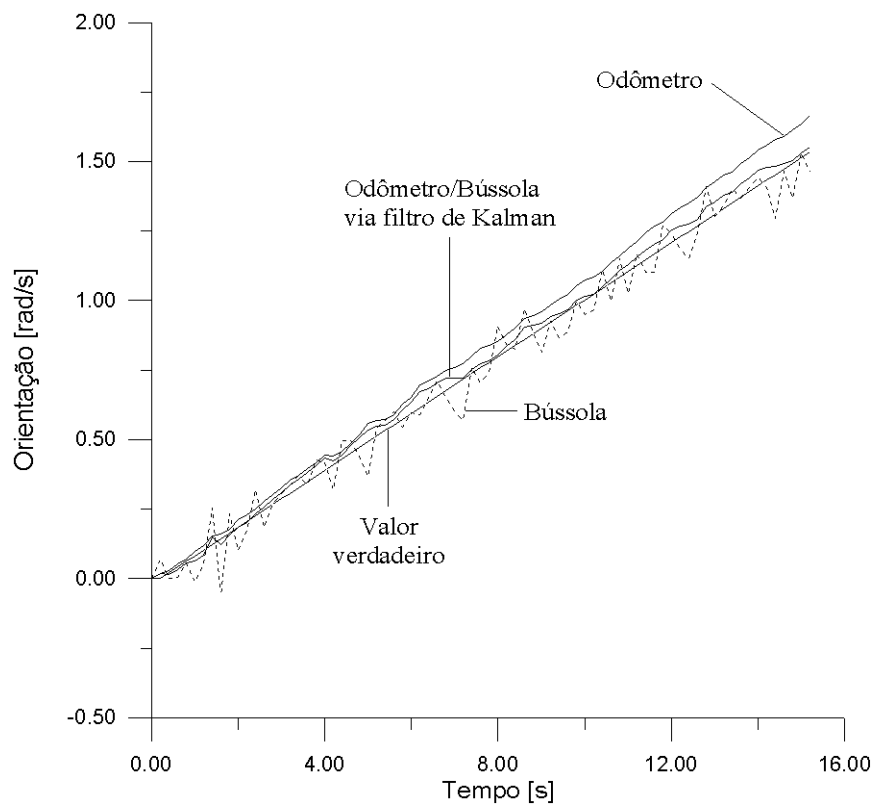


Fig. VI-8 - Simulação da orientação.

VI.3.2. Resultados Experimentais

Navegou-se em uma trajetória circular, com raio mantido constante via elo rígido, de um ângulo de 90° , com os mesmos parâmetros utilizados na simulação. A trajetória nominal percorrida pelo veículo foi então calculada. A Fig VI-9 mostra a trajetória nominal e as posições calculadas com base na integração odômetro/bússola e com o odômetro apenas.

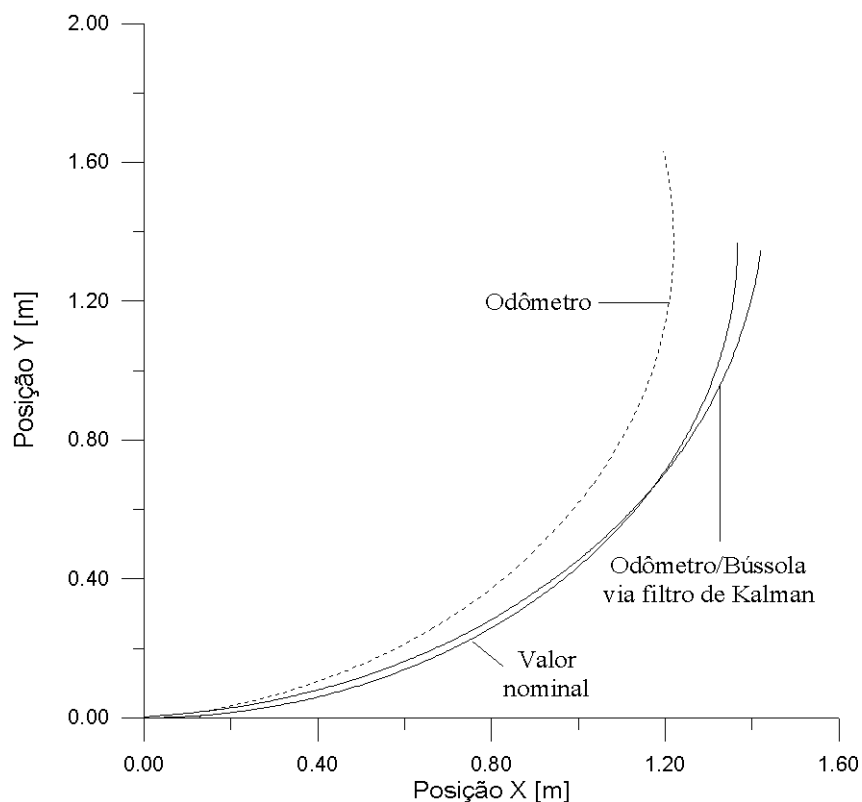


Fig. VI-9 - Posição calculada pelo odômetro e pela fusão odômetro/bússola em tempo real.

Note-se que a posição estimada oscila em torno da trajetória nominal crescendo com o tempo (devido à acumulação do erro de posição), mas sempre está mais perto do valor nominal do que o odômetro. A Fig. VI-10 mostra as várias orientações relativas a esta realização.

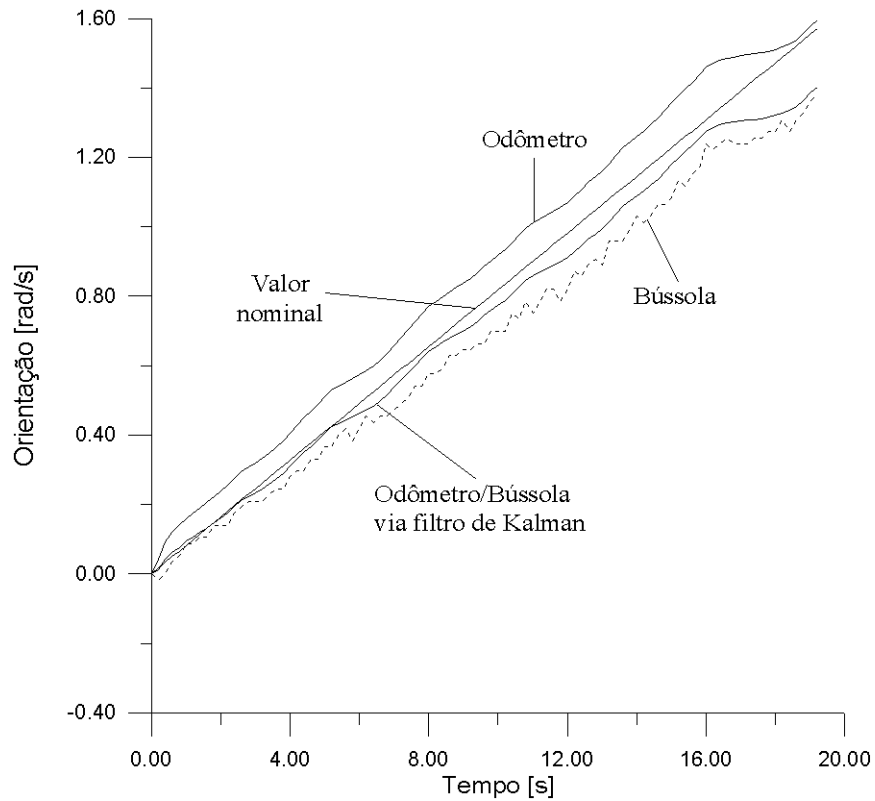


Fig. VI-10 - Orientações obtidas em tempo real.

VI.4. Conclusão

Foram apresentados neste capítulo métodos para estimação de posição e orientação para robôs móveis, utilizando fusão de dados, via filtro de Kalman. Foram apresentados resultados simulados da fusão de dados obtidos por *dead-reckoning* e dados obtidos por visão computacional. Foram também mostrados resultados simulados e obtidos experimentalmente da fusão de dados obtidos por *dead-reckoning* e dados de orientação obtidos por uma bússola digital.

Capítulo VII

Implementação

Os algoritmos apresentados nos capítulos anteriores foram implementados em dois protótipos desenvolvidos no ITA-IEEE, denominados IEEE-Rover e Twil. A seguir será descrita a arquitetura destes robôs.

VII.1. Descrição do *Hardware* do IEEE-Rover

Na Figura VII-1 pode ser vista uma fotografia do robô IEEE-Rover.

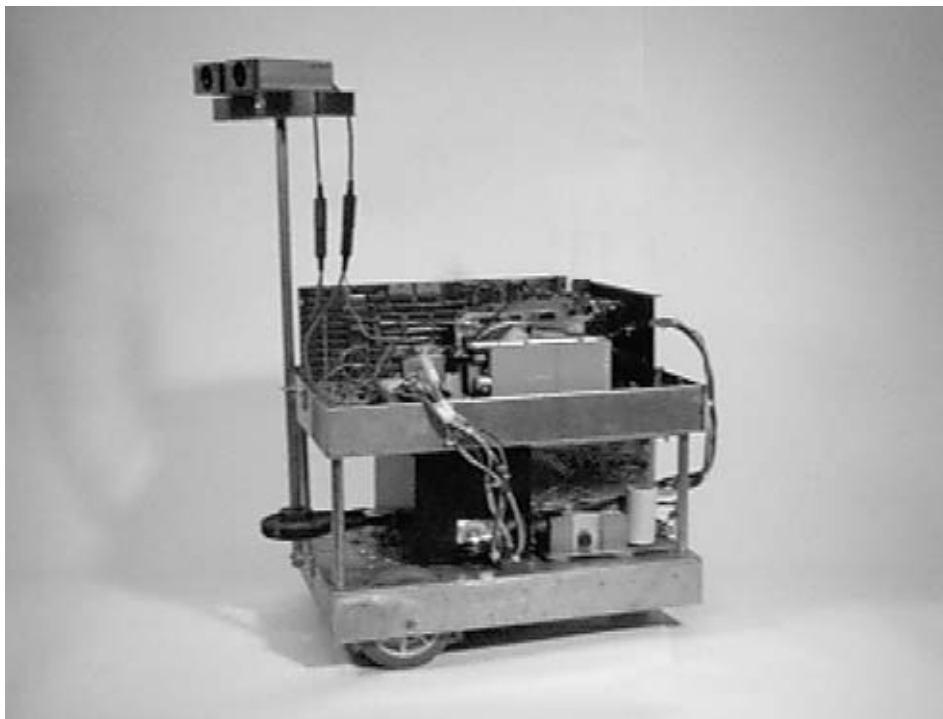


Fig. VII-1 - Fotografia do IEEE-Rover.

Este robô, do tipo (2,0), é atuado em modo diferencial através das rodas dianteiras. Na parte traseira existe uma roda livre apenas para apoio. Duas câmeras de vídeo estão montadas em uma torre que pode ser orientada independentemente da plataforma.

Na figura VII-2 pode ser visto um diagrama de blocos do sistema.

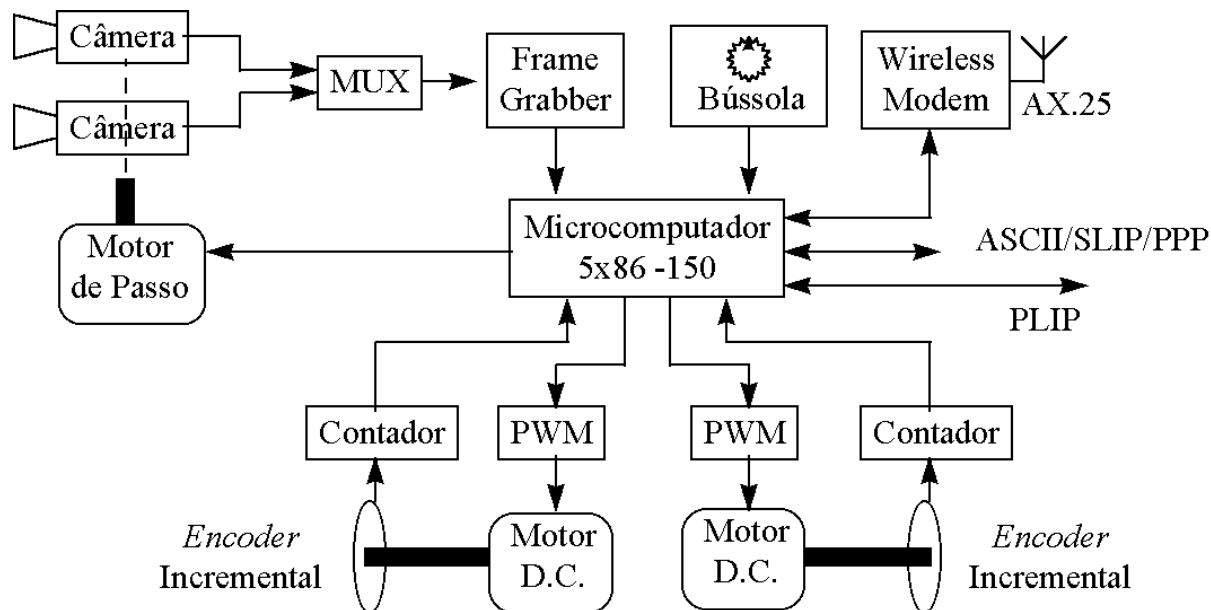


Fig. VII-2 - Diagrama de Blocos do IEEE-Rover.

O IEEE-Rover é controlado por um computador baseado no microprocessador 5x86 com *clock* de 150MHz.

Os motores das rodas dianteiras são acionados através de circuitos PWM. Diferentemente dos circuitos convencionais, os PWMs utilizados no IEEE-Rover são totalmente digitais, não empregando a clássica técnica de comparação de um sinal analógico com uma onda triangular. Desta forma evita-se o uso de conversores D/A. Os PWMs existentes no IEEE-Rover são implementados utilizando-se um *timer* programável 82C54. Este *chip* possui quatro contadores (decrementes) programáveis de 16 bits. Todos os contadores utilizam um *clock* de 10MHz. O contador 0 é utilizado no modo 2 (*rate generator*). Neste modo o contador gera um pulso na saída sempre que a contagem atinge 0. Os contadores 1 e 2 são utilizados no modo 1 (*hardware retriggerable one-shot*). Neste modo a saída do contador vai para nível lógico baixo quando do início da contagem e retorna para nível lógico alto ao final da contagem. Os dois PWMs necessários para acionamento dos motores são obtidos conectando-se a saída do contador 0 às entradas de disparo dos contadores 1 e 2. Desta forma, quando o valor programado nos contadores 1 e/ou 2 for tal que

a contagem chegue a 0 antes do que a contagem do timer 0, a largura do pulso na saída destes contadores (1 e 2) será proporcional ao valor da contagem.

Portanto, o valor programado no contador 0 determina a frequência de operação dos PWMs enquanto os valores programados nos contadores 1 e 2 determinam a largura de pulso dos PWMs conectados aos motores direito e esquerdo, respectivamente.

É importante notar que a frequência de operação dos PWMs pode ser modificada por *software*. Quanto maior esta frequência, menores serão os erros de quantização do sinal gerado. Por outro lado, a potência dissipada nos transistores que acionam os motores será maior, reduzindo a autonomia das baterias. Normalmente os PWMs do IEEE-Rover operam a 30Khz. Esta frequência de operação permite que o sinal de saída seja quantizado em 333 níveis sem provocar grandes perdas de potência nos transistores. Além disso, o ruído gerado não será audível, pois está fora da faixa de frequências percebida pelo ouvido humano. A possibilidade de alterar a frequência de operação dos PWMs por *software* é interessante por que permite que se utilize frequências de operação baixas (poupando energia) para longos percursos, que em geral não necessitam grande precisão de posicionamento e frequências mais altas (gastando mais energia) quando é necessário um posicionamento mais preciso.

Os motores são instrumentados com *encoders* incrementais de 1000ppr. Estes *encoders* possuem dois canais em quadratura, portanto são capazes de detectar também o sentido de giro do motor. Um benefício adicional da decodificação dos sinais gerados por *encoders* com dois canais é que a resolução é multiplicada por quatro [10], obtendo-se portanto, uma resolução efetiva de 4000ppr. Isto pode ser mais facilmente entendido observando-se a figura VII-3.

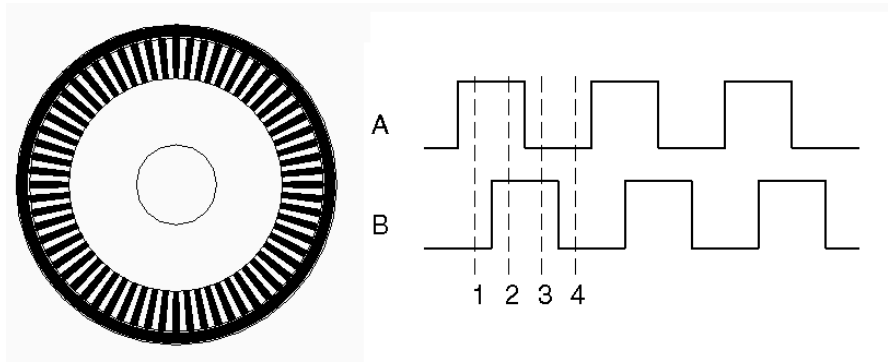


Fig. VII-3 - Disco de Encoder e Sinais Gerados.

A fase relativa entre os canais A e B pode ser usada para determinar-se a direção de rotação. Os quatro estados possíveis para os sinais A e B permitem que a resolução final seja quatro vezes a resolução do disco.

Os pulsos dos *encoders* são injetados em contadores bi-direcionais de 16 bits operando em complemento 2, devido ao fato dos *encoders* serem capazes de gerar informação sobre o sentido de giro dos motores. Foi verificado experimentalmente que a sintonização das malhas internas de controle de velocidade dos motores torna-se bastante difícil se esta informação não estiver disponível.

A torre com as câmeras de vídeo é movimentada através de um motor de passo de $1,8^\circ$ por passo. Estas câmeras estão conectadas através de um multiplexador de vídeo à uma placa de digitalização de imagens DT2851 [16], capaz de digitalizar imagens com resolução de $512 \times 480 \times 256$ em tempo real e de realizar várias operações nas imagens através de *look-up tables*. Existe ainda um *micro-switch* para detectar quando a torre está em sua orientação de referência.

Os sensores do IEEE-Rover incluem também uma bússola digital [56], com resolução de 1° e precisão de 2° , com uma taxa de amostragem máxima de 5Hz.

O computador montado no IEEE-Rover não possui teclado nem monitor de vídeo. A operação do sistema é feita através de uma conexão com outro computador. Existem quatro métodos de conexão que podem ser utilizados:

a) terminal via linha serial: neste método um terminal RS-232 ou outro computador executando um programa emulador de terminal conecta-se através de um cabo *null-modem*. A vantagem deste método é a simplicidade. No entanto, não é possível fazer a transferência de arquivos entre os dois sistemas. Além disso, a maioria dos emuladores de terminais tem problemas em tratar alguns caracteres de controle normalmente utilizados em alguns programas, principalmente editores de texto que operam em tela cheia. Este método é utilizado basicamente para configurar-se o *software* de rede para permitir que os outros métodos sejam utilizados.

b) conexão TCP/IP via SLIP ou PPP: neste método é utilizado o mesmo cabo null-modem da opção anterior para estabelecer uma conexão SLIP (*Serial Line IP*) ou PPP (*Point-to-Point Protocol*) entre o IEEE-Rover e outro computador. Os protocolos TCP/IP (*Transfer Control Protocol/Internet Protocol*) podem então ser utilizados para fornecer vários serviços entre o IEEE-Rover e o computador do operador, sendo particularmente úteis os serviços de telnet e ftp.

c) conexão TCP/IP via PLIP: Este método é basicamente o mesmo do caso anterior, porém a conexão SLIP é substituída por uma conexão PLIP (*Parallel Line IP*) utilizando um cabo *null-printer*, portanto a taxa de transferência de dados pode ir bem além dos 115200 bps permitidos pela porta serial existente. Com uma conexão PLIP pode-se obter taxas de mais de 100KB/s. A grande desvantagem deste método está no fato do protocolo PLIP não ser um padrão oficial como o PPP, nem ao menos um padrão de fato, como o SLIP.

d) conexão TCP/IP via AX.25: neste método são utilizados modems de RF [57] para estabelecer-se uma conexão AX.25 entre o IEEE-Rover e o computador do operador. Os pacotes TCP/IP são encapsulados em pacotes AX.25. A utilização do protocolo AX.25 é necessária porque a transmissão via rádio envolve vários fenômenos não presentes quando a comunicação se dá via cabo, como sombreamento, interferência, estações fora de alcance, etc.

Estes métodos não são mutuamente exclusivos. Pode-se estabelecer várias conexões simultaneamente, com qualquer combinação de métodos. A única restrição são os dispositivos de hardware existentes. Os métodos baseados no protocolo TCP/IP permitem que o IEEE-Rover seja acessado através de uma rede local, eventualmente com vários robôs ou até mesmo através da Internet.

O protocolo AX.25, tal como implementado, permite que um único computador comunique-se com vários robôs ou ainda que os vários robôs comuniquem-se entre si através de um único canal de R.F. Os modems de R.F. utilizados operam em *half-duplex* com uma taxa de até 34800 bps. Como todos os modems operam na mesma frequência (433.13 MHz), torna-se necessário implementar uma política de acesso ao meio de transmissão para evitar que mais de um modem transmita ao mesmo tempo, o que pode causar a perda da mensagem. No IEEE-Rover é utilizada a técnica CSMA (*Carrier Sense Medium Access*) **p**-persistente. Não é utilizada detecção de colisão (CSMA-CD) porque quando a transmissão é feita por R.F. é difícil garantir que todas as estações envolvidas na colisão serão capazes de detectá-la [68].

Em linhas gerais a técnica CSMA **p**-persistente consiste no seguinte [68]: quando uma estação deseja transmitir ela verifica se o meio de transmissão está livre, isto é, se não existe portadora presente. Se o meio de transmissão estiver livre, a estação inicia a transmissão com probabilidade **p**. Com probabilidade $(1 - p)$ ou se o meio de transmissão não estiver livre a transmissão é retardada até o *slot* de tempo seguinte. Este processo é repetido até que a transmissão seja feita. No IEEE-Rover é utilizado $p=0.25$ e um *slot* de tempo de 10 ms.

VII.2. Descrição do Software do IEEE-Rover

A fusão de dados de diversos sensores utilizando o filtro de Kalman é uma técnica que tem sido bastante utilizada para obter-se uma estimativa mais confiável das grandezas medidas. É comum considerar-se, nestes casos, que todos os sensores envolvidos possuem a

mesma taxa de amostragem. Isto é, assume-se que todos os sensores são capazes de fornecer medidas periodicamente e que este período é constante e igual para todos os sensores.

No entanto, quando são utilizados sensores baseados em diferentes princípios físicos, é comum a situação onde cada sensor exige uma taxa de amostragem diferente. Nestes casos, a simulação do sistema, ou uma implementação da fusão de dados *off-line* não é muito diferente do caso com uma única taxa de amostragem. Porém, uma implementação em tempo real apresenta uma série de problemas relacionados ao processamento para aquisição de dados de cada sensor, com o sincronismo entre os sensores e destes com o filtro de Kalman.

Conforme comentado na seção VII.1 o IEEE-Rover está equipado com *encoders*, sensores de visão e uma bússola digital. O diagrama de temporização para acionamento da bússola utilizada pode ser visto na Fig. VII-4. O símbolo ✓ indica processamento pelo microcomputador. Este processamento consiste apenas da leitura ou escrita de alguns bits nas portas de I/O, portanto é executado em no máximo alguns μ s. Os números indicam os *delays* necessários em ms. O símbolo ■ indica o tempo restante dentro do período de amostragem da bússola de 200ms. Note-se que embora o processamento para leitura da bússola demore em torno de 135ms, o processamento é necessário apenas durante breves intervalos de alguns μ s. A maior parte do tempo necessário para leitura da bússola é constituído por *delays*.

✓	10	✓	80-100	10	✓	5	✓	1	✓	1	...	1	✓	5	✓	■
disparo			espera	habilitação			Leitura (16 bits)							Liberação		livre

Fig. VII-4 - Diagrama de temporização da bússola.

Este diagrama de temporização pode facilmente ser implementado em *software*. Porém, para implementação em tempo real do sistema proposto é necessário também efetuar a leitura dos *encoders*, cujo diagrama de temporização pode ser visto na Fig.VII-5. Novamente, o

símbolo ✓ indica processamento pelo microcomputador, da ordem de alguns μ s, e o símbolo ■ indica o tempo restante dentro do período de amostragem dos *encoders*, que é de 50ms.

✓	■
Leitura	livre

Fig. VII-5 - Diagrama de temporização dos *encoders*.

Percebe-se, então, que a leitura da bússola não pode ser feita pelas técnicas convencionais de *busy-wait*, pois, devido ao seu tempo de processamento em torno de 135ms, isto faria com que os *encoders* não pudessem ser processados a cada 50 ms. Necessita-se portanto de um ambiente multitarefa. Neste caso, a leitura da bússola e a leitura dos *encoders* são feitas por tarefas periódicas diferentes, cada uma com seu tempo de amostragem. Quando uma tarefa necessita efetuar um *delay* ela é posta para *dormir* durante este tempo, liberando o processador para as demais tarefas.

Este ambiente multitarefa foi implementado utilizando-se o sistema operacional Linux[79]. É importante notar, porém, que as tarefas não foram implementadas como processos, mas sim como *threads*[69]. Processos operam com espaços de endereçamento diferentes entre si, enquanto *threads* compartilham o mesmo espaço de endereço, além de vários outros recursos. Uma consequência direta deste fato é que a troca de dados entre *threads* é feita mais facilmente do que entre processos.

O programa implementado possui quatro *threads*: 1) leitura da bússola, com período de amostragem de 200ms; 2) leitura dos *encoders*, com período de amostragem de 50ms; 3) fusão de dados, sem período de amostragem fixo, mas sincronizado com os dois *threads* anteriores; 4) interface com o usuário, com período de amostragem de 1s. Este *thread* possui a menor prioridade e portanto executa apenas quando os demais *threads* não necessitam utilizar o processador.

O sincronismo entre os *threads* é realizado através de semáforos[7]. Semáforos são variáveis controladas pelo sistema operacional sobre as quais são definidas as operações *post* e *wait* (além da inicialização). A operação *wait* testa o valor do semáforo. Se o semáforo é zero, o processo (ou *thread*, neste caso) é suspenso. Caso contrário, o valor do semáforo é decrementado e o processo continua normalmente. A operação *post* acorda um processo que esteja suspenso no semáforo ou, caso não haja nenhum processo nesta condição, o valor do semáforo é incrementado.

Aqui semáforos são utilizados para garantir: a) que a operação do sistema em tempo real comece apenas após todos os *threads* terem completado a sua inicialização, e b) que a fusão de dados ocorre apenas quando os dados da bússola e dos *encoders* estão disponíveis, sincronizando a operação entre os *threads* de leitura dos sensores e o *thread* de fusão de dados. Em linhas gerais, os *threads* de leituras dos sensores seguem o algoritmo resumido abaixo (os semáforos são inicializados com valor zero):

```
void compass(struct cps_param *data)
{
    // inicialização
    sem_post(&data->ready);
    sem_wait(&data->start);
    for(;;)
    {
        // leitura do sensor
        sem_post(&data->avail);
        // suspender até próximo período de amostragem
        sem_wait(&data->ack);
    }
}
```



```
}
```

O *thread* de fusão de dados executa o seguinte algoritmo:

```
void fusion(struct fus_param *data)
{
    // inicialização
    int c=3; // c = contador de sincronismo
    sem_post(&data->ready);
    sem_wait(&data->start);
    for(;;)
    {
        sem_wait(&(data->enc_data->avail));
        // processamento dos encoders
        sem_post(&(data->enc_data->ack));
        if(c==0)
        {
            c=4
            sem_wait(&(data->cps_data->avail));
            // processamento da bússola
            sem_post(&(data->cps_data->ack));
            // filtragem
        }
        c--;
        // controle dos motores
    }
}
```

Na Fig. VII-6 são mostrados os períodos de amostragem da bússola e dos *encoders*. O nível lógico do sinal é invertido a cada amostragem. A amostragem da bússola não é realizada em sincronismo com a amostragem dos *encoders*, mas é mantida a razão de uma amostragem da bússola para cada quatro amostragens dos *encoders*.

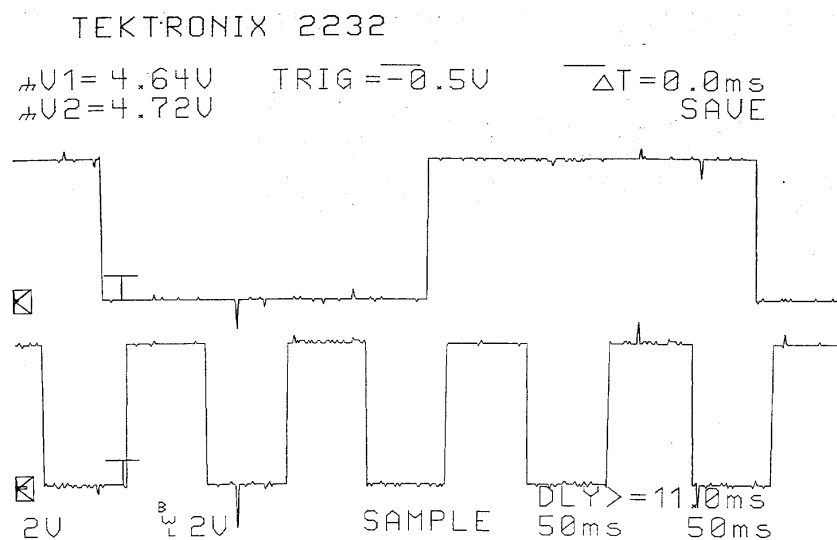


Fig. VII-6- Períodos de amostragem da bússola e dos *encoders*.

O sincronismo é feito no *thread* de fusão de dados, como pode ser verificado na Fig. VII-7, onde é mostrado o sinal gerado pelo *thread* de leitura da bússola e o sinal gerado pelo *thread* de fusão de dados. Novamente tem-se uma inversão do nível lógico do sinal por *loop* da rotina. O período de execução da fusão de dados não é mantido exatamente em 50ms, mas varia ligeiramente para forçar o sincronismo entre os sensores.

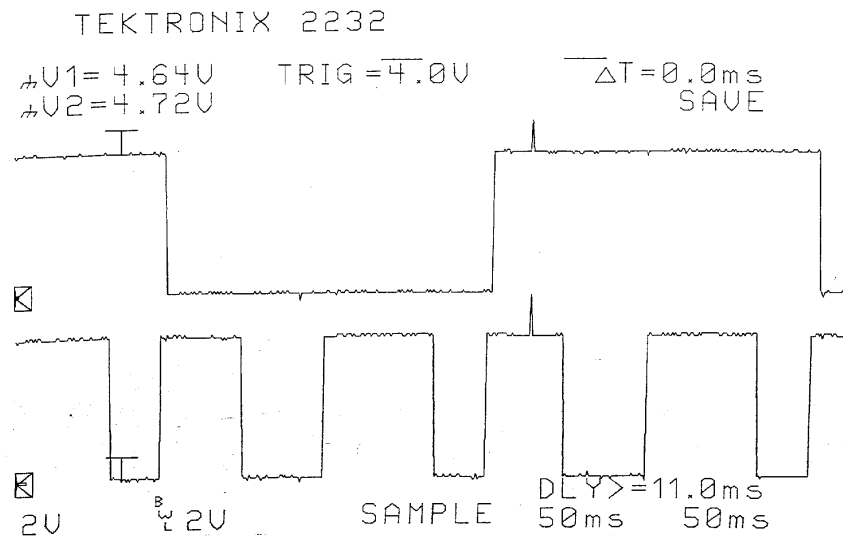


Fig. VII-7- Temporização da bússola e da fusão de dados.

VII.3. O Robô Móvel Twil

O robô móvel Twil foi desenvolvido a partir do IEEE-Rover, e portanto é bastante similar. Nesta seção serão descritas as diferenças entre os dois robôs. Nas Figs. VII-8 e VII-9 podem ser vistos uma fotografia e um diagrama de blocos do robô móvel Twil. O diagrama de blocos do IEEE-Rover é basicamente o mesmo. A diferença é que ele não possui sonares nem joystick.

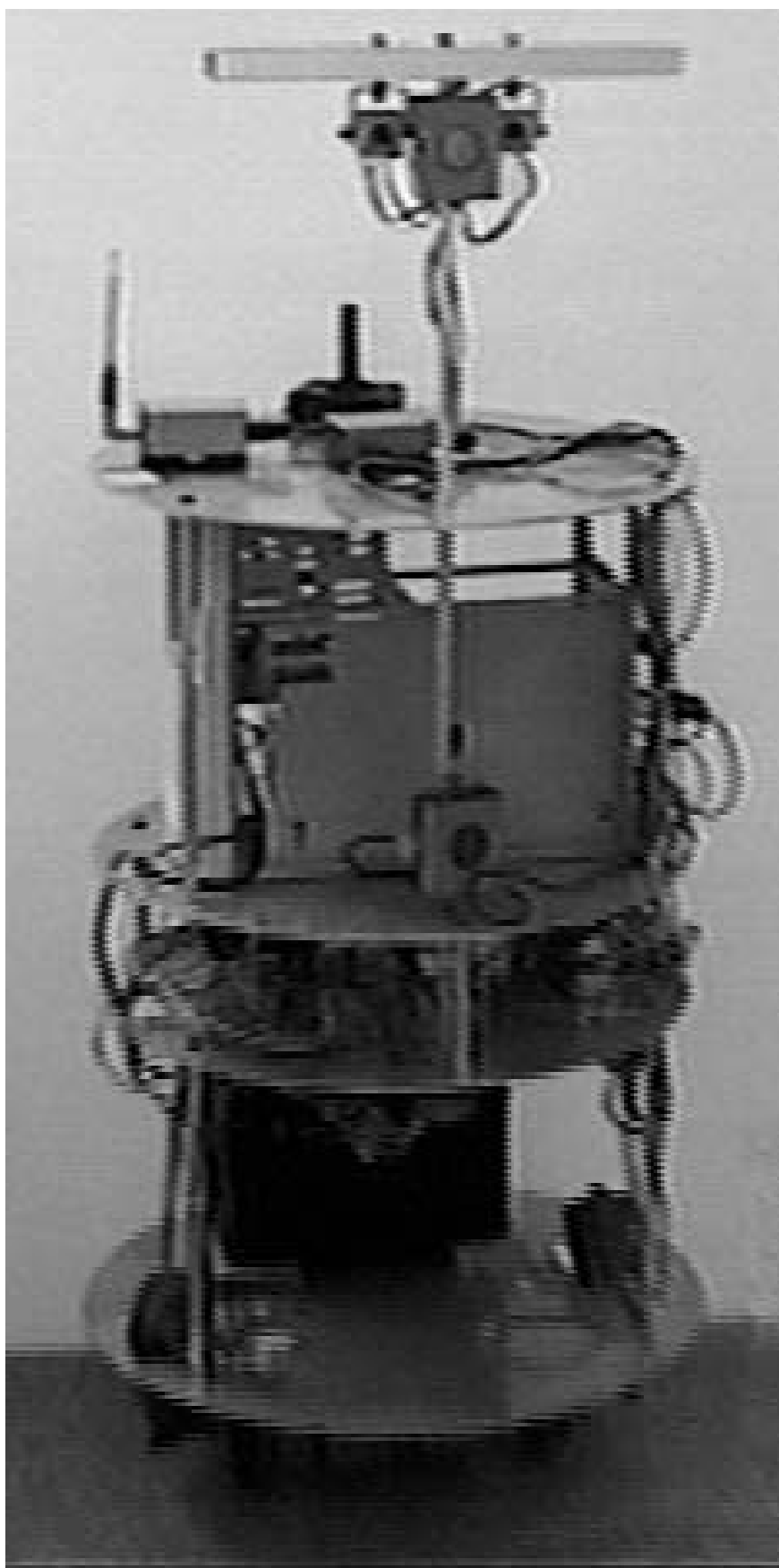


Fig. VII-8 - Fotografia do robô móvel Twil.

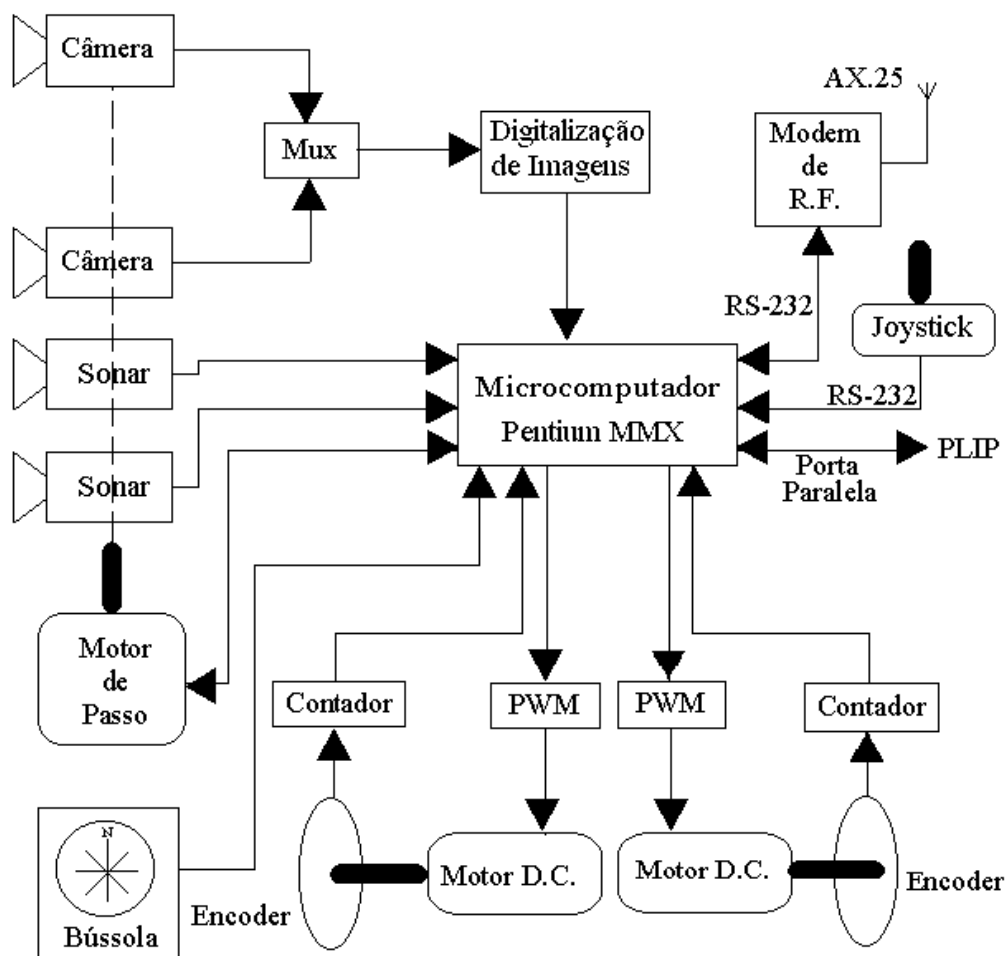


Fig. VII-9 - Diagrama de blocos do robô móvel Twil

Cada roda possui um *encoder* incremental que permite a medição do deslocamento angular das rodas. Duas câmeras de vídeo estão montadas na frente da plataforma, formando um sistema de visão stereo e podem ser movimentadas de modo a focalizar pontos do ambiente interessantes para controle e navegação. Na mesma torre que suporta as câmeras de vídeo existem dois sonares de ultra-som, montados em alturas diferentes. Os sensores do robô incluem ainda uma bússola digital.

O robô pode comunicar-se com outros computadores através de um modem de Rádio Frequência utilizando o protocolo AX.25 ou através de uma interface paralela utilizando o protocolo PLIP.

Embora os diagramas de blocos de ambos robôs sejam similares, a implementação de alguns blocos difere. O Twil é controlado por um computador baseado no microprocessador Pentium MMX de 233 MHz, enquanto o IEEE-Rover é controlado por um computador baseado no microprocessador 5x86 com *clock* de 150MHz. A placa de digitalização de imagens do IEEE-Rover possui maior resolução, mas no Twil o acesso à imagem digitalizada é mais eficiente, pois a placa de vídeo armazena a imagem na própria memória do *host*, ou invés de armazenar em memória própria. Os *encoders* do Twil possuem uma resolução maior (8192ppr) do que as do IEEE-Rover (4000ppr), o que lhe garante uma melhor precisão de posicionamento.

A necessidade de um *joystick* no Twil deve-se a seu peso (45Kg), que torna inconveniente a sua movimentação manual. Por ser bem mais leve (5.6Kg), o IEEE-Rover dispensa este acessório. A interface RS232 não utilizada pelo *joystick*, pode então ser utilizada para estabelecer uma comunicação serial utilizando-se o protocolo SLIP ou PPP. No entanto, estes protocolos não apresentam vantagem, pois necessitam de uma conexão por cabo, como o PLIP e a taxa de transferência de dados é comparável a obtida com o AX.25.

Utilizando-se as suas facilidades de comunicação, o robô Twil pode ser acessado através da Internet, utilizando-se *browsers* convencionais. Na Fig. VII-10 é mostrada uma das páginas que podem ser acessadas, mostrando dados obtidos em tempo real.

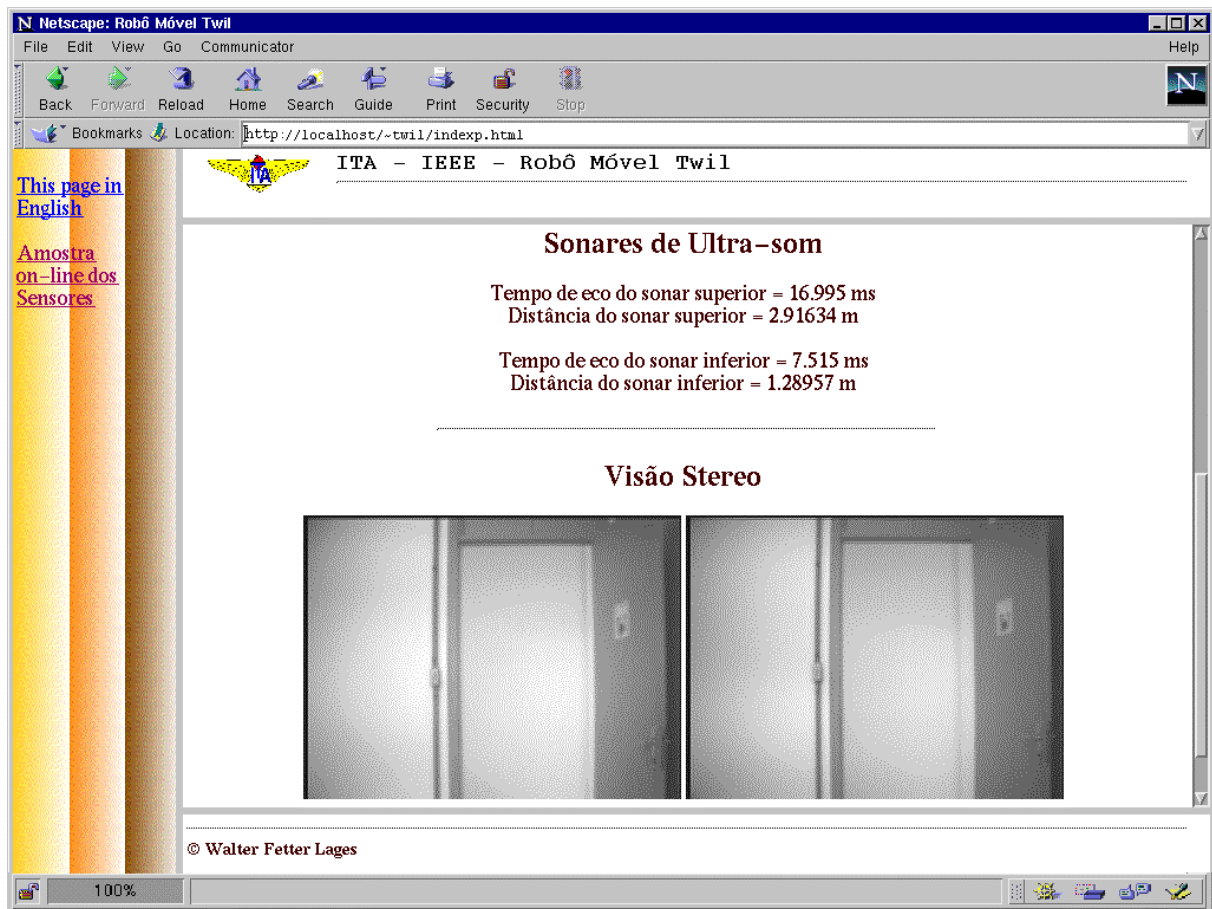


Fig. VII-10 - Amostra em tempo real dos sensores do robô Twil, acessado através da Internet utilizando-se um browser convencional.

VII.4. Conclusão

Neste capítulo foram descritos dois protótipos implementados para testar as técnicas desenvolvidas neste trabalho. Alguns aspectos de implementação, que geralmente não são considerados quando o sistema é simulado foram discutidos. A plataforma de *software* desenvolvida, utilizando o sistema operacional Linux e uma biblioteca de *threads* permite que algoritmos de controle em tempo real sejam implementados de forma bastante confortável. Além disso, esta arquitetura permite que os robôs sejam acessados via Internet.

Capítulo VIII

Conclusões

Neste trabalho foram desenvolvidos métodos de controle de robôs móveis utilizando-se o modelo dinâmico completo do robô. O objetivo é abordar o problema de estabilização da posição e orientação do robô em um ponto, também denominado problema de *parking*.

Inicialmente, os modelos cinemático e dinâmico de robôs móveis com rodas foram desenvolvidos com base em [14]. Através das definições dos graus de mobilidade e dirigibilidade, conclui-se que é possível a existência de apenas cinco classes de modelos de robôs móveis com rodas.

A seguir, de forma semelhante a [77], é proposta a utilização de métodos de linearização por realimentação para desenvolvimento de uma lei controle estabilizante. Como forma de generalizar os resultados obtidos por [77], a aplicabilidade deste método propostos às diversas classes de robôs móveis com rodas foi investigada. Conclui-se que a a linearização por realimentação não é adequada para controle de robôs das classes (1,1) e (1,2). Na realidade, esta técnica só permite que se controle completamente robôs da classe (3,0). Para robôs das classes (2,0) e (2,1) é possível controlar apenas a posição. Estas restrições podem ser contornadas utilizando-se linearização dinâmica. No entanto, isto só pode ser feito sob a hipótese de que a velocidade robô não é nula. Esta hipótese contradiz a premissa inicial que é estabilizar o robô em determinado ponto.

Para contornar estas limitações, procurou-se, conforme sugerido em [42][43], utilizar transformações de coordenadas não suaves, seguidas de uma realimentação suave invariante no tempo. A idéia nesta abordagem é fazer uma transformação de coordenadas de forma que o sistema transformado seja descontínuo em malha aberta, conforme em [3], mas em malha

fechada torne-se contínuo. Utiliza-se aqui uma abordagem semelhante a [1], que utiliza uma transformação para coordenadas polares.

Mostrou-se que os modelos de robôs pertencentes a uma determinada classe podem ser transformados para modelos equivalentes pertencentes a qualquer outra classe com o mesmo grau de manobrabilidade. Este resultado permitiu que as leis de controle desenvolvidas para as classes (3,0) e (2,0) fossem utilizadas para controlar robôs pertencentes a qualquer classe.

Em [1] é considerado apenas o modelo cinemático de um robô da classe (2,0). Portanto, também foi feita a generalização para permitir a sua aplicação à todas as classes de robôs com rodas. Como esta abordagem não considera o modelo dinâmico, foi proposta, como forma de viabilizar a sua implementação, a utilização de estratégias de *backstepping* [19]. Duas estratégias de *backstepping* foram desenvolvidas. A primeira, utilizando controladores PID, não considera explicitamente o modelo dinâmico. A segunda, permite a consideração do modelo dinâmico do robô, através da utilização de um controlador linearizante semelhante ao empregado anteriormente para controlar o modelo completo do robô.

Tal como ocorre com os robôs manipuladores, os modelos cinemáticos e dinâmicos possuem estrutura bem conhecida, mas os parâmetros destes modelos apresentam problemas. Os parâmetros cinemáticos devem ter seus valores conhecidos com boa precisão, pois interferem diretamente nos erros de determinação de posição e orientação por *dead-reckoning*, que são acumulativos. A fusão de dados de diversos sensores operando sob princípios físicos diferentes, permite uma redução substancial na taxa de crescimento dos erros. Mais especificamente, foram discutidos dois métodos de fusão de dados obtidos a partir de *encoders*, visão computacional e bússola digital.

Já os parâmetros dinâmicos, por serem variantes com a carga útil do robô, não podem ser determinados por calibração, ensejando o desenvolvimento de versões adaptativas dos

controladores propostos. Estes controladores foram desenvolvidos com base em [52]. A versão adaptativa do controlador linearizante segue basicamente os resultados apresentados em [52], sendo porém necessário algum trabalho de forma a parametrizar o modelo completo do robô móvel de forma que este seja linear nos parâmetros desconhecidos.

A versão adaptativa do controlador *backstepping* foi obtida através da extensão dos resultados apresentados em [52] para sistemas do tipo MIMO com grau relativo 1. O fato do sistema a ser controlado pelo controlador *backstepping* apresentar grau relativo 1 possibilita que a prova de estabilidade do sistema em malha fechada, isto é, processo, controle e algoritmo de adaptação seja realizada em conjunto.

Os algoritmos de controle desenvolvidos neste trabalho foram validados através de simulações e da implementação de dois protótipos. O desenvolvimento de protótipos é bastante útil no sentido de apontar problemas que não ocorrem em simulação. Notadamente, a determinação da posição e orientação real do robô, que embora em simulação sejam conhecidas com exatidão, na implementação experimental devem ser estimadas através de medidas indiretas, freqüentemente utilizando fusão de dados de vários sensores. As temporizações que devem ser obedecidas para obtenção dos dados destes sensores também são geralmente negligenciadas em simulação. Na implementação real, ocorrem vários problemas de sincronismo e inversões de prioridades entre os sensores, devido ao período de amostragem de cada sensor ser diferente dos demais e dos períodos de amostragem dos controladores. A solução destes problema exigiu que fosse utilizado um sistema multitarefa de tempo real.

Em trabalhos futuros, seria interessante o desenvolvimento de modelos e controladores similares aos propostos aqui para robôs com trailers, bem como o desenvolvimento de times assimétricos, compostos de robôs representantes das diversas classes discutidas. A

implementação do protocolo AX.25 desenvolvida permitiria a comunicação entre os robôs, de forma que estes colaborem entre si para execução de uma tarefa comum.

Outra possibilidade, é o estudo de métodos de estimação de posição e orientação que beneficiem-se da linearização do modelo do robô móvel obtida através de dos métodos aqui propostos. Como o modelo do robô torna-se linear, é razoável supor-se que a fusão de dados poderia ser realizada utilizando-se o filtro de Kalman linear ao invés do filtro estendido.

O desempenho dos diversos controladores propostos neste trabalho não foi comparado. A comparação de desempenho de controladores para robôs é uma tarefa dificultada pelo fato de que, em geral, o desempenho do controlador depende da trajetória de referência. Alguns controladores comportam-se melhor para determinados tipos de trajetória. Além disso, para que seja possível comparar o desempenho de controladores é necessário que os seus parâmetros sejam otimizados. Neste trabalho também não foram realizados esforços no sentido de otimizar os parâmetros dos controladores utilizados. Portanto, uma comparação do desempenho dos controladores não seria apropriada.

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*, 2(1):27-35, Mar. 1995.
- [2] ASAMI, S., Robots in Japan: Present and Future. *IEEE Robotics & Automation Magazine*, 1(2):22-26, June 1994.
- [3] ASTOLFI, A., Discontinuous Control of Nonholonomic Systems. *ETH Technical Report AUT95-09*, 1995. Disponível *on-line* em <ftp://ftp.aut.ee.ethz.ch/pub/reports/postscript/AUT95-09.ps>.
- [4] BALLARD, D. H., Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 13:111-122, 1981.
- [5] BARSHAN B. & DURRANT-WHYTE, H., Inertial Navigation Systems for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 11(3): 328-342, June 1995.
- [6] BAUMGARTNER, E. T. & SKAAR, S. B., An Autonomous Vision-Based Mobile Robot. *IEEE Transactions on Automatic Control*, 39(3):493-502, Mar. 1994.
- [7] BEN-ARI, M., Principles of Concurrent Programming. Englewood Cliffs, Prentice-Hall, 1982.
- [8] BLOCH, A. M.; REYHANOGLU, M. & MCCLAMROCH, N. H., Control and Stabilization of Nonholonomic Dynamic Systems. *IEEE Transactions on Automatic Control*, 37(11): 1746-1756, Nov. 1992.
- [9] BORENSTEIN, J. & FENG, L., UMBMark - A Method for Measuring, Comparing and Correcting Dead-Reckoning Errors in Mobile Robots. *Technical Report UM-MEAM-94-22*, University of Michigan, 1994.

- [10] BORENSTEIN, J.; EVERETT, H. R. & FENG, L., Where Am I? - Sensors and Methods for Mobile Robot Positioning. University of Michigan Technical Report, Apr. 1996.
- [11] BOZMA, Ö. & KUC, R., A Physical Model-Based Analysis of Heterogeneous Environments Using Sonar - ENDURA Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):497-506, May 1994.
- [12] BROCKETT, R. W., Asymptotic Stability and Feedback Stabilization. in *Differential Geometric Control Theory*. Brockett, Millman & Sussman, Eds, Boston - MA: Birkhauser, 1983, p. 181-208.
- [13] BYLER, E.; WENDELL, C.; HOFF, W. & LAYNE, D., Autonomous Hazardous Waste Drum Inspection Vehicle. *IEEE Robotics & Automation Magazine*, 2(1):6-17, Mar. 1995.
- [14] CAMPION, G.; BASTIN, G. & D'ANDRÉA-NOVEL, B., Structural Properties and Classification of Kinematic and Dynamical Models of Wheeled Mobile Robots. *IEEE Transactions on Robotics and Automation*, 12(1):47-62, Feb. 1996.
- [15] CANUDAS de WIT, C. & SØRDAL, O. J., Exponential Stabilization of Mobile Robots with Nonholonomic Constraints. *IEEE Transactions on Automatic Control*, 37(11):1791-1797, Nov. 1992.
- [16] DATA TRANSLATION, User Manual for DT2851 High Resolution Frame Grabber, *Document UM-05234J*, 1991.
- [17] FENG, L.; KOREN, Y. & BORENSTEIN, J., Cross-Coupling Motion Controller for Mobile Robots. *IEEE Control Systems Magazine*, 13(6):35-43, Dec. 1993.
- [18] FENTON, R. E., IVHS/AHS: Driving into the Future. *IEEE Control Systems Magazine*, 14(6):13-20, Dec. 1994.
- [19] FIERRO, R. & LEWIS, F. L., Control of a Nonholonomic Mobile Robot Using Neural Networks. *IEEE Transactions on Neural Networks*, 9(4):589-600, May 1998

- [20] FU, K. S.; GONZALEZ, R. C. & LEE, C. S. G., Robotics: Control, Sensing, Vision and Intelligence. McGraw-Hill, 1987.
- [21] GAT, E.; DESAI, R.; IVLEV, R.; LOCH, J & MILLER, D., Behavior Control for Robotic Exploration of Planetary Surfaces. *IEEE Transactions on Robotics and Automation*, 10(4):490-503, Aug. 1994.
- [22] GIRI, F.; IOANNOU, P. A. & AHMED-ZAID, F., A Stable Indirect Control Scheme for First-Order Plants with no Prior Knowledge on the Parameters. *IEEE Transactions on Automatic Control*, 38(5): 766-771, May 1993.
- [23] GOODWIN, G. & SIN, K., Adaptive Filtering Prediction and Control. Englewood Cliffs, Prentice-Hall, 1984.
- [24] HEDRICK, J. K.; TOMIZIKA, M. & VARAIYA, P., Control Issues in Automated Highway Systems. *IEEE Control System Magazine*, 14(6):21-32, Dec. 1994.
- [25] HEMERLY, E. M. & RODRIGUES, C. C., Guiagem de Veículos Autônomos Utilizando Sensor de Visão. In: *10º Congresso Brasileiro de Automática*, Rio de Janeiro, 1994.
- [26] HOLLESTEIN, A. A., Aufdatierung der Position und der Orientierung eines mobilen Roboters. Zürich, 1992, ETH Doktor Abhandlung.
- [27] HORN, B., Robot Vision. Cambridge, MA, MIT Press, 1986.
- [28] ISIDORI, A., Nonlinear Control Systems. Berlin, Springer-Verlag, 1995.
- [29] KELLY, A., Essential Kinematics for Autonomous Vehicles. Carnegie Mellon University Technical Report CMU-RI-TR-94-14 disponível *on-line* em <ftp://reports.adm.cs.cmu.edu/usr/anon/robotics/CMU-RI-TR-94-14.ps.Z>, May, 1994.
- [30] KNOLL, A., Sensordatenfusion für Anwendungen in der Robotik. Technische Bericht KIFS 94 Universität Bielefeld, 1994.
- [31] KOLMANOVSKY, I. & MCCLAMROCH, H. H., Developments In Nonholonomic Control Problems. *IEEE Control Systems Magazine*, 15(6):20-36, Dec. 1995.

- [32] KUC, R. & SIEGEL, M. W., Physically Based Simulation Model for Acoustic Sensor Robot Navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(6):766-778, Nov. 1987.
- [33] LAGES, W. F. & HEMERLY, E. M., Adaptive Linearizing Control of Mobile Robots, In: Proceedings of the 5th IFAC Workshop on Intelligent Manufacturing Systems, Gramado - RS, Nov. 1998.
- [34] LAGES, W. F. & HEMERLY, E. M., Controle em Coordenadas Polares de Robôs Móveis com Rodas. In: Anais do XII Congresso Brasileiro de Automática, vol. II, pp. 605-610, Uberlândia - MG, set. 1998.
- [35] LAGES, W. F. & HEMERLY, E. M., Smooth Time-invariant Control of Wheeled Mobile Robots. In: *Proceedings of The XIII International Conference on Systems Science*, Technical University of Wroclaw. Wroclaw, Poland, 1998.
- [36] LAGES, W. F.; HEMERLY, E. M. & PEREIRA, L. F. A., Controle Linearizante de uma Plataforma Móvel Empregando Realimentação Visual. In: *Anais do XI Congresso Brasileiro de Automática*, pp. 537-542, São Paulo, 1996.
- [37] LAPRADE, A. & LAMPERT-TORRES, G., Controle de um Veículo Utilizando a Teoria dos Conjuntos Nebulosos. In: *Anais do I Simpósio Brasileiro de Automação Inteligente*, Rio Claro, 1993.
- [38] LATOMBE, J. C., Robot Motion Planning. Boston, Kluwer Academic Publishers, 1993.
- [39] LEONARD, J. J. & DURRANT-WHYTE, H. F., Direct Sonar Sensing for Mobile Robot Navigation. Kluwer, Norwell, Ma, 1992.
- [40] LOBO, J.; LUCAS, P.; DIAS, J. & TRAÇA DE ALMEIDA, A., Inertial Navigation System for Mobile Land Vehicles. In: *Proceedings of the IEEE International Symposium on Industrial Electronics* Vol. 2, New York, NY, USA, 1995, p.843-848.

- [41] MAIGRET, P., Reactive Planning and Control with Mobile Robots. *IEEE Control Systems Magazine*, 12(3):95-100, June 1992.
- [42] MCCLAMROCH, N. H. & KOLMANOVSKY, I., Developments in Nonholonomic Control Problems. *IEEE Control Systems Magazine*, 15(6):20-36, Dec. 1995.
- [43] MCCLOSKEY, R. T. & MURRAY, R. M., Exponential Stabilization of Driftless NonLinear Control Systems Usinsing Homogeneous Feedback. *IEEE Transactions on Automatic Control*, 42(5):614-628, May 1997.
- [44] MURATA, S. & HIROSE, T., On Board Locating System Using Real-Time Image Processing for a Self-Navigating Vehicle. *IEEE Transactions on Industrial Electronics*, 40(1):145-153, Feb. 1993.
- [45] MURRAY, R. M. & SASTRY, S. S., Nonholonomic Motion Planning: Steering Using Sinusoids, *IEEE Transactions on Automatic Control*, 38(5): 700-715, May 1993.
- [46] NEGAHDARIPOUR, S. & HORN, B. K. P., Direct Passive Navigation. *IEEE Transactions. on Pattern Analysis and Machine Intelligence*, 9(1):168-176, Jan. 1987.
- [47] NEGAHDARIPOUR, S.; HAYASHI, B. Y. & ALOIMONOS, Y., Direct Motion Stereo for Passive Navigation. *IEEE Transactions on Robotics and Automation*, 11(6):829-843, Dec. 1995.
- [48] NELSON, W. L., Continuous Steering-Function Control of Robot Carts. *IEEE Transactions on Industrial Electronics*, 36(3): 330-337, Aug. 1989.
- [49] NIELSEN, R. O., Sonar Signal Processing. Norwood , Artech House, 1991.
- [50] NIJMEIER, H & VAN DER SHAFT, A., Nonlinear Dynamical Contol Systems. New York, Springer-Verlag, 1990.

- [51] OELEN, W.; BERGHUIS, H.; NIJMEIJER, H. & CANUDAS DE WIT, C., Hybrid Stabilizing Control on a Real Mobile Robot. *IEEE Robotics & Automation Magazine*, 15(3):16-23, June 1995.
- [52] PEREIRA, L. F. A., Controle Adaptativo de Sistemas Não-Lineares Linearizáveis por Realimentação de Estados. *Tese de Doutorado, ITA*, São José dos Campos, 1995.
- [53] PERKINS, S. & HAYES, G., Real Time Optical Flow Based Range Sensing in Mobile Robots. Department of Artificial Intelligence, University of Edinburgh. Disponível *on-line*.
- [54] POMERLEAU, D. A., Neural Network Perception for Mobile Robot Guidance. Boston, Kluwer, 1993.
- [55] POPOV, V. P., Hyperstability of Control Systems. Berlin, Springer-Verlag, 1973.
- [56] PRECISION NAVIGATION, INC., Vector 2X Compass Module Application Notes Version 1.03, jan. 1996.
- [57] RADIO-TECH, BIM-XXX-RS232 Transceiver Data Sheet
- [58] REEDS, J. A. & SHEPP, L. A., Optimal Paths for a Car that Goes both Forwards and Backwards. *Pacific Journal of Mathematics*, 145(2): 367-393, 1990.
- [59] SANDI LORA, F. A.; HEMERLY E. M. & LAGES, W. F., Sistema para Navegação e Guiagem de Robô Móveis Autônomos, *Revista SBA Controle e Automação*, 9(3):107-118, set., out., nov., dez. 1998
- [60] SANDI LORA, F. A.; HEMERLY, E. M. & LAGES, W. F., Estimação em Tempo Real de Posição e Orientação de Robôs Móveis Utilizando Sensores com Diferentes Taxas de Amostragem. In: *Anais do 3º Simpósio Brasileiro de Automação Inteligente*, Vitória - ES, set. 1997.
- [61] SASTRY, S. S & BODSON, M., Adaptive Control: Stability, Convergence and Robustness. New Jersey, Englewood Cliffs, 1989.

- [62] SASTRY, S. S. & ISIDORI, A., Adaptive Control of Linearizable Systems. *IEEE Transactions on Automatic Control*, 34(11):1123-1131, Nov. 1989.
- [63] SCHRAFT, R. D., Mechatronics and Robotics for Service Applications. *IEEE Robotis & Automation Magazine*, 1(4):31-35, Dec. 1994.
- [64] SIMONS, REID G., Structured Control for Autonomous Robots. *IEEE Transactions on Robotics and Automation*, 10(1):34-43, Feb. 1994.
- [65] SOATTO, S., FREZZA, R. & PERONA, P., Motion Estimation via Dynamic Vision. *IEEE Transactions on Automatic Control* 41(3):393-413, Mar. 1996.
- [66] TALLURI, R. & AGGARWAL, J. K., Image/Map Correspondence for Mobile Robot Self-Location Using Computer Graphics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):597-601, June 1993.
- [67] TALLURI, R. & AGGARWAL, J. K., Position Estimation for an Autonomous Mobile Robot in an Outdoor Environment. *IEEE Transactions on Robotics and Automation*, 8(5):575-584, Oct. 1992.
- [68] TANENBAUM, A., Computer Networks. Englewood Cliffs, Prentice-Hall, 1988.
- [69] TANENBAUM, A., Modern Operating Systems. Englewood Cliffs, Prentice-Hall, 1992.
- [70] THUILOT, B.; D'ANDRÉA-NOVEL, B. & MICAELLI, A., Modeling and Feedback Control of Mobile Robots Equipped with Several Steering Wheels. *IEEE Transactions on Robotics and Automation*, 12(3): 375-390, June 1996.
- [71] VESTLI, S. J., Fast, Accurate and Robust Estimation of Mobile Robot Position and Orientation. Dissertação de Doutorado submetida ao ETH, Zurique, 1995.
- [72] WALSH, G.; TILBURY, D.; SASTRY, S.; MURRAY, R. & LAUMOND, J. P., Stabilization of Trajectories for Systems with Nonholonomic Constraints. *IEEE Transactions on Automatic Control*, 39(1):216-222, Jan. 1994.

- [73] WEISBIN, C. R. & LAVERY, D., NASA Rover and Telerobotics Technology Program. *IEEE Robotics & Automation Magazine*, 1(4):14-21, Dec. 1994.
- [74] WENG, J. , COHEN, P. & REBIBO, N., Motion and Structure Estimation from Stereo Image Sequences. *IEEE Transactions on Robotics and Automation*, 8(3):362-382, June 1992.
- [75] WERB, J. & LANZI, C., Designing a Positioning Systems for Finding Things and People Indoors. *IEEE Spectrum*, 35(9):71-78, Sep. 1998.
- [76] YAGI, Y.; KAWATO, S. & TSUJI, S., Real-Time Omnidirectional Image Sensor (COPIS) for Vision-Guided Navigation. *IEEE Transactions on Robotics and Automation*, 10(1):11-22, Feb. 1994.
- [77] YAMAMOTO, Y. & YUN, X., Coordinating Locomotion and Manipulation of a Mobile Manipulator. *IEEE Transactions on Automatic Control*, 39(6): 1326-1332, June 1994.
- [78] YLÄ-JÄÄSKI, A. & KIRYATI, N., Adaptive Termination of Voting in the Probabilistic Circular Hough Transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):911-915, Sep. 1994.
- [79] YODAIKEN, V., Cheap Operating Systems Research and Teaching with Linux. disponível on-line em <http://luz.cs.nmt.edu/~rtlinux>.
- [80] YUH, J., Learning Control for Underwater Robotic Vehicles. *IEEE Control Systems Magazine*, 14(2):39-46, Apr. 1994.
- [81] ZORPETTE, G., Autopilots of the Deep. *IEEE Spectrum*, 34(8):38-44 Aug. 1994.

