

Universidade Federal da Bahia - UFBA
Escola Politécnica
Programa de Pós-Graduação em Engenharia Elétrica

Dissertação de Mestrado

**Controle Preditivo de um Robô Omnidirecional com
Compensação de Atrito**

Aluno: Júlio César Lins Barreto Sobrinho
Orientador: Carlos Eduardo Trabuco Dórea
Orientador: André Gustavo Scolari Conceição
Co-orientador: Edson Roberto de Pieri

Salvador
15 de Abril de 2011

Júlio César Lins Barreto Sobrinho

**Controle Preditivo de um Robô Omnidirecional com
Compensação de Atrito**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia como requisito para a obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Carlos Eduardo Trabuco Dórea
Orientador: André Gustavo Scolari Conceição
Co-orientador: Edson Roberto de Pieri

**Universidade Federal da Bahia - UFBA
Escola Politécnica
Programa de Pós-Graduação em Engenharia Elétrica**

Salvador
15 de Abril de 2011

Dedicatória

Dedico esse trabalho a todos que estiveram envolvidos nessa minha caminhada, e em especial à minha amada tia Lajucy (in memoriam) pela dedicação e devoção a mim e aos meus objetivos. Ela foi responsável, juntamente com meus pais, pela formação do meu caráter. Apesar dela não estar mais entre nós, tenho certeza que suas orações no plano espiritual contribuíram e muito nessa conquista.

Agradecimentos

A Deus, por sempre me dar forças de onde eu menos espero e por permitir que eu supere meus limites físicos para que eu possa atingir os meus objetivos.

A toda minha família, pela confiança, dedicação e apoio incondicional.

Aos meus colegas e amigos, pela torcida e pelo incentivo.

Aos meus médicos e fisioterapeutas, que são responsáveis pela minha saúde, elemento essencial na minha caminhada.

Ao programa de Pós-graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa, ao programa Pró Engenharia UFBA/UFSC, e à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, pela provisão da bolsa de mestrado.

Ao Professor Carlos Eduardo Trabuco Dórea por estar sempre solícito e pronto para ajudar, pelas lições sobre Controle Preditivo por Variáveis de Estado sob restrições e por ter conseguido conduzir bem esse trabalho, mesmo de longe. Ao Professor André Scolari pelas noções sobre robótica e pela plataforma experimental que permitiu aplicar o controle preditivo ao robô e por ter sido responsável pela execução dos experimentos, o que foi indispensável para a realização desse trabalho. E à professora Luciana Martinez pela orientação a respeito da implementação do algoritmo dos Pontos Interiores.

Sumário

1	Introdução Geral	1
1.1	Objetivo	3
1.2	Justificativa	3
1.3	Capítulos da Dissertação	4
2	Robô Omni-Direcional	5
2.1	Estrutura do robô omnidirecional	6
2.2	Modelo	7
2.2.1	Dinâmica do Robô	8
2.3	Resultados	11
2.4	Conclusões	13
3	Controle Preditivo do Robô Omnidirecional de 3 rodas	14
3.1	Controle Preditivo Linear	14
3.2	Controle de Velocidade do Robô	17
3.3	Simulações do Controle Preditivo	20
3.3.1	Controle de Velocidades	20
3.3.2	Compensação de zona morta	21
3.3.3	Seguimento de trajetórias	23
3.4	Controle Preditivo do Robô Omnidirecional de 3 rodas considerado como um sistema afim por partes	27
3.5	Conclusões	30
4	Implementação do controlador no robô	39
4.1	Lazarus/Comunicação	39
4.2	Resultados Experimentais	39
4.2.1	Controle de Velocidades	40
4.2.2	Compensação de zona morta	42
4.2.3	Seguimento de trajetórias	44
4.3	Conclusões	46
5	Conclusão Geral	51
A	Método dos Pontos Interiores	55

Lista de Publicações

- CONCEICAO, A. G. S. ; DÓREA, C. E. T. ; BARRETO SOBRINHO, J. C. L. . Controle Preditivo de Um Robô Móvel Omnidirecional com Compensação de Atrito. In: Congresso Brasileiro de Automática, 2010, Bonito, MS. Anais do XVIII CBA, 2010. v. 1. p. 1129-1135.
- CONCEICAO, A. G. S. ; DÓREA, C. E. T. ; BARRETO SOBRINHO, J. C. L. . Predictive Control of an Omnidirectional Mobile Robot with Friction Compensation. In: 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting, 2010, São Bernardo do Campo. Anais do LARS 2010, 2010. v. 1. p. 30-35.

Lista de Figuras

2.1	Rodas Omnidirecionais.	6
2.2	Robô móvel omnidirecional de 3 rodas.	6
2.3	Diagrama de blocos do acionamento das rodas.	6
2.4	Diagrama de blocos do robô.	7
2.5	referencial do robô.	7
2.6	Comparação da velocidade v do modelo e do robô real, com $u_1 = 0V$, $u_2 = 3V$ e $u_3 = -3V$	12
2.7	Comparação da velocidade v_n do modelo e do robô real com $u_1 = -3V$, $u_2 = 1,5V$ e $u_3 = 1,5V$	12
2.8	Comparação da velocidade ω do modelo e do robô real com $u_1 = 1V$, $u_2 = 1V$ e $u_3 = 1V$	13
3.1	Estrutura básica que realiza o controle preditivo.	14
3.2	Esquema de controle.	18
3.3	Trajectoria de referência.	20
3.4	Simulação do controle de velocidades, referência $v = 1m/s$, $v_n = 0m/s$ e $\omega = 0rad/s$	21
3.5	Simulação do controle de velocidades, referência $v = 1m/s$, $v_n = 1m/s$ e $\omega = 0rad/s$	22
3.6	Simulação do controle de velocidades, referência $v = 1m/s$, $v_n = 0m/s$ e $\omega = 1rad/s$	23
3.7	Simulação do controle de velocidades, referência $v = 1m/s$, $v_n = 1m/s$ e $\omega = 1rad/s$	24
3.11	Trajectoria a ser seguida.	24
3.8	Simulação da compensação de zona morta para v (ma - malha aberta).	25
3.9	Simulação da compensação de zona morta para v_n	25
3.10	Simulação da compensação de zona morta para ω	26
3.12	Simulação do seguimento da trajetória degrau, $R = I$	31
3.13	Tensão no motor 3. Continuação da figura 3.12.	32
3.14	Simulação do seguimento da trajetória degrau, $R = 10^{-2}I$	32
3.15	Simulação do seguimento da trajetória degrau, $R = 10^{-2}I$, tensões nos motores CC.	33
3.16	Simulação do seguimento da trajetória degrau, $H_p = H_u = 5$	33
3.17	Simulação do seguimento da trajetória degrau, $H_p = H_u = 5$, continuação.	34
3.18	Seguimento das referências de velocidade, $H_p = H_u = 5$, para $R = 10^{-2}I$ (gráfico de cima) e $R = 10^{-1}I$ (gráfico de baixo).	35
3.19	Simulação do seguimento da trajetória em oito.	35
3.20	Representação geométrica do controlador no espaço de estados.	36
3.21	Simulação do controle do robô visto como um sistema afim por partes (PWA- <i>piecewise affine</i>), referência $v = 1m/s$	36
3.22	Simulação do controle do robô visto como um sistema PWA, referência $v_n = 1m/s$	36
3.23	Simulação do controle do robô visto como um sistema PWA, referência $\omega = 1rad/s$	37

3.24	Simulação do seguimento da trajetória degrau, comparação do controle preditivo com compensação de atrito com o controle do modelo visto como um sistema PWA.	37
3.25	Simulação do seguimento da trajetória degrau, comparação do controle preditivo com compensação de atrito com o controle do modelo visto como um sistema PWA, continuação.	38
4.1	<i>Software Decision.</i>	40
4.2	Experimento com a referência $v = 1m/s$.	41
4.3	Experimento com a referência $vn = 1m/s$.	42
4.4	Experimento com a referência $\omega = 1rad/s$.	42
4.5	Experimento com a referência $v = 0,5m/s$ e $\omega = 1rad/s$.	43
4.6	Experimento com a referência de $v = 0,6m/s$ e $vn = -0,6m/s$.	44
4.7	Compensação de zona morta para v .	44
4.8	Compensação de zona morta para vn .	45
4.9	Compensação de zona morta para ω .	45
4.10	Seguimento da trajetória em degrau, $v_{nav} = 0,3m/s$.	47
4.11	Componente ω da velocidade. Continuação da figura 4.10.	48
4.12	Seguimento da trajetória diagonal, $v_{nav} = 0,6m/s$.	48
4.13	Componentes de velocidade. Continuação da figura 4.12.	49
4.14	Seguimento da trajetória em oito, $v_{nav} = 0,5m/s$.	49
4.15	Seguimento da trajetória em oito, $v_{nav} = 0,5m/s$, continuação.	50

Lista de Tabelas

2.1	Parâmetros do modelo [Maxon, 2002].	10
3.1	Parâmetros da simulação e tempo total de navegação.	27

Lista de Símbolos e Siglas

PID - Proporcional Integral Derivativo
PD - Proporcional Derivativo
 x - vetor de estados
 u - vetor de ações de controle
 y - vetor de saídas medidas
 z - vetor de saídas a serem controladas
 r - vetor de referências
 A, B, C - matrizes do espaço de estados
 C_y e C_z - matriz C do espaço de estados relativa à saída medida e controlada, respectivamente
 A_d, B_d, C_d - matrizes do espaço de estados discreto
 \hat{z} - vetor de saídas previstas
 \hat{x} - vetor de estados previstos
 \hat{u} - vetor de ações de controle previstas
 $V(k)$ - função custo
 H_p - horizonte de predição
 H_u - horizonte de controle
 H_w - atraso
 T_s - Período de amostragem
 $Q(i)$ - matrizes simétricas semi-definidas positivas
 $R(i)$ são matrizes simétricas definidas positivas
 Q - matriz de ponderação do erro de rastreamento, dimensão igual ao número de saídas do sistema
 R - matriz de ponderação do esforço de controle, dimensão igual ao número de entradas do sistema
 v_{nav} - velocidade de navegação
 \bar{u} - vetor de ações de controle no equilíbrio
 \bar{x} - referência do estado
 $\hat{Z}(k)$ - sequência de saídas previstas
 $T(k)$ - sequência de referências previstas
 $\hat{U}(k)$ - sequência de entradas previstas
 Q_{custo} - matriz particionada diagonal, os elementos da diagonal principal são iguais a Q
 R_{custo} - matriz particionada diagonal, os elementos da diagonal principal são iguais a R
 Ψ e Θ - matrizes da predição
 $\epsilon(k)$ - Valor atual da diferença entre a referência $T(k)$ e $\Psi x(k)$
 G - matriz que relaciona as forças nas rodas com as forças no robô
 H - termo quadrático da função custo
 F - matriz de restrições
 f - vetor que contém as restrições nas ações de controle
MPC - *Model Predictive Control* (Controle Preditivo baseado em modelo)
NMPC - *Nonlinear Model Predictive Control* (Controle Preditivo não-linear baseado em modelo)
PWA - *Piecewise Affine* (Afim por partes)
MPT - *Multi Parametric Toolbox*
mp-LP - problema linear multi paramétrico
mp-QP - problema quadrático multi paramétrico
QP - *Quadratic Programming* (Programação Quadrática)
CC - corrente contínua
UAVs - veículos aéreos não tripulados
AUVs - veículos autônomos subaquáticos
 v e vn - velocidades na direção direta e normal ao plano do movimento do robô, respectivamente

ma - malha aberta
 $F_v(t)$ e $F_{vn}(t)$ - forças de tração no robô relativas a v e vn
 Γ - torque de rotação do robô
 $\omega(t)$ - velocidade angular do robô
 \bar{v} , \bar{v}_n e $\bar{\omega}$ - velocidades de referência
 e_{vx} , e_{vy} , e_ω - velocidades de referência no sistema de coordenadas da terra
 M - massa do robô
 J - momento de inércia do robô
 B_v , $B_v n$ e B_ω - coeficientes de atrito viscoso relativos a cada componente de velocidade e a ω , respectivamente
 C_v , $C_v n$ e C_ω - coeficientes de atrito de Coulomb
 $f_i(t)$ - as forças de tração das rodas do robô
 $T_i(t)$ - torques rotacionais das rodas
 r_i - raios das rodas
 l_i - reduções nos motores
 b - distância entre o centro do robô e as rodas
 L_{a_i} - indutância de armadura
 R_{a_i} - resistência de armadura
 i_{a_i} - corrente de armadura
 i_{max} - corrente máxima
 K_{v_i} - constante EMF
 K_{t_i} - constante de torque
 $x(t)$ - coordenada x
 $y(t)$ - coordenada y
 $\theta(t)$ - ângulo de orientação
 x_r - coordenada x do centro de massa P do robô
 y_r - coordenada y do centro de massa P do robô
 \bar{x}_r - coordenada x da trajetória de referência
 \bar{y}_r - coordenada y da trajetória de referência
 $\bar{\theta}$ - orientação da trajetória de referência
 $R(\theta)$ - matriz de rotação
 v_1 , v_2 , v_3 e v_4 - velocidades das rodas
 u_1 , u_2 e u_3 - tensão nos motores 1, 2 e 3
 u_i - tensão no motor i
 u_f - parcela das ações de controle utilizada para compensar as não-linearidades
 u_c - parcela relativa ao controle propriamente dito
 K - matriz que representa as não-linearidades da planta contínua
 K_d - matriz das não-linearidades da planta discretizada
 I - matriz identidade
 LTI - *Linear Time Invariant* (Linear invariante no tempo)
 a_i - Valor do elemento i de $signal(x)$
 f_i - Valor do termo independente da dinâmica i
 D_i - uma região no espaço de estados
 Q_f - Matriz de ponderação do erro, valor terminal
 $A_i e B_i$ - matrizes A e B do espaço de estados para a dinâmica i
 W_i - limite da restrição, sistema PWA
 L_i - Valor que multiplica x_k na restrição do custo para o modelo PWA
 E_i - Valor que multiplica u_k na restrição do custo para o modelo PWA
 X_{set} - conjunto poliédrico dos estados para uma determinada dinâmica
 $w(k)$ - parte não-linear do espaço de estados
 f_{dyn} - função dinâmica que obtém $x(k+1)$ a partir de $u(k)$, $x(k)$ e $w(k)$
 g_{dyn} - função dinâmica que obtém $y(k)$ a partir de $u(k)$ e $x(k)$

u_{min} e u_{max} - limites de variação de $u(k)$
 $\Delta u(k)$ - variação de u entre os instantes $k - 1$ e k
 Δu_{min} e Δu_{max} - limites de variação de $\Delta u(k)$
 y_{min} e y_{max} - limites de variação de $y(k)$
 T_{set} - união de todos conjuntos poliédricos
 I_f - número de dinâmicas
 I - conjunto das dinâmicas
 CLMR - *Car Like Mobile Robot* (Robô móvel em forma de carro)
 WMR - *Wheeled Mobile Robot* (Robô Móvel com rodas)
 GPS - *Global Positioning System* (Sistema de Posicionamento Global)
 c_E - restrições de igualdade
 c_I - restrições de desigualdade
 s - vetor de variáveis de folga
 s_i - elemento i de s
 μ - o parâmetro de barreira
 A_E e A_I - Jacobianas das funções c_E e c_I
 y e z - multiplicadores de Lagrange
 S e Z - matrizes diagonais, cujos elementos são dados pelos vetores s e z
 σ - constante que multiplica μ
 e - vetor unitário que multiplica μ
 $f(x)$ - uma função hipotética a ser otimizada, onde x é a variável do problema
 $\nabla f(x)$ - gradiente de f
 ∇_{xx}^2 - Hessiana
 g_- - vetor gradiente da função a ser minimizada.
 D - matriz que multiplica u_{aum} na inequação das restrições de desigualdade
 d - vetor de restrições de desigualdade
 A_{eq} - matriz que multiplica u_{aum} na equação das restrições de igualdade
 b_{eq} - vetor de restrições de igualdade
 u_{aum} - u aumentado
 u_{aum}^+ , s^+ , y^+ e z^+ - valores atualizados de u_{aum} , s , y e z
 α - comprimento do passo (em cada direção)
 α_y^{max} , α_s^{max} e α_z^{max} - α máximo de y , s e z
 p - vetor de variáveis independentes na equação matricial do algoritmo dos pontos interiores
 pu_{aum} , ps , py e pz - vetores que compõem o vetor de variáveis independentes na equação matricial do algoritmo dos pontos interiores
 τ - parâmetro utilizado para que se possa obter o α máximo

Resumo

Esse trabalho tem como principal objetivo a implementação e análise do desempenho de um algoritmo de controle preditivo aplicado a um modelo não-linear de um Robô Móvel Omnidirecional de 3 rodas, que deve ser controlado para seguir trajetórias de referência. Primeiramente foi feito um estudo do modelo do robô, incluindo uma descrição dos procedimentos para a estimação dos parâmetros do modelo, que foi validado através da realização de experimentos. O algoritmo MPC (*Model Predictive Control*) básico foi adaptado para incorporar o tratamento das não-linearidades oriundas das forças de atrito do robô com o solo. O robô também foi representado por um modelo afim por partes (PWA) e uma lei de controle *off-line* foi obtida, utilizando para isso o toolbox MPT (*Multi Parametric Toolbox*) do *Matlab/Simulink*, com o intuito de comparar as duas técnicas. O algoritmo MPC aqui desenvolvido realiza um controle de velocidade e o modelo da cinemática inversa é utilizado para gerar a referência de velocidade a partir da posição atual do robô. Foi proposto, portanto, um sistema de controle em cascata para a resolução do problema. O controlador foi testado em simulações e experimentos, utilizando para isso o ambiente Lazarus.

Palavras-chave: Robôs móveis, Sistemas de Controle, Modelagem, Controle Preditivo, MPC (*Model Predictive Control*), sistema PWA (Afim por partes), MPT (*Multi Parametric Toolbox*).

Abstract

This work's main objective is the implementation and performance analysis of a predictive control algorithm that was applied to a Nonlinear Model of a 3 wheel Omnidirectional Mobile Robot, that should be controlled to follow reference trajectories. Firstly, the robot's model was studied, including a description of the procedures to estimate the model parameters, this model was validated by the realization of experiments. The basic MPC (Model Predictive Control) algorithm was adapted to include the treatment of the nonlinearities from the friction forces between the robot and the floor. A piecewise affine (PWA) model was also used to represent the robot and an offline control law was obtained, using the MPT (Multi Parametric Toolbox) from Matlab-Simulink, aiming to compare the techniques. The developed MPC algorithm implements a velocity control and the inverse kinematics model is used to generate the reference velocities from the robot's current position. The proposed problem is solved using a cascade control system. The controller was tested in simulations and experiments using the Lazarus environment. Keywords: Mobile Robots, Control Systems, Modelling, Predictive Control, MPC (Model Predictive Control), PWA (Piecewise Affine) System, MPT (Multi Parametric Toolbox).

Capítulo 1

Introdução Geral

Recentemente, na área de robótica, muita atenção tem se voltado para os robôs móveis omnidirecionais, já que eles têm mobilidade no plano e podem se mover em qualquer direção, sem reorientação. Ao contrário de robôs não-holonômicos, como os robôs nas configurações Ackerman e diferencial, que têm que executar primeiro o movimento de rotação, para depois desempenhar uma velocidade de translação, robôs omni têm um maior poder de manobra e são muito utilizados em aplicações em ambientes dinâmicos. Exemplos importantes de aplicações que utilizam esses robôs são o transporte de cargas em fábricas e o futebol de robôs.

Em um robô omnidirecional, um modelo dinâmico descreve as relações entre as forças das rodas e o movimento do mesmo com a tensão aplicada a cada motor acoplado às rodas (entradas) e a velocidade ou posição (saídas). Geralmente são estimados o momento de inércia e os coeficientes de atrito viscoso e de Coulomb, a massa, o raio das rodas e a distância entre o centro do robô e as rodas são medidos, e os valores dos parâmetros dos motores, como as reduções, e as resistências e indutâncias de armadura, são fornecidos pelo fabricante. Mas as incertezas no modelo tornam o projeto de controle mais complexo. Ainda podem ser considerados os atritos e a saturação dos atuadores. Já que as velocidades das rodas, que são impostas pelos motores, não podem ultrapassar os limites de saturação, a saturação dos atuadores pode afetar o desempenho do robô, podendo fazer o movimento ser instável [Li and Zell, 2007]. A depender da aplicação desejada pode-se realizar algumas simplificações [Conceição et al., 2009], como aproximar o modelo estático ou dinâmico de atrito, supor que a massa é uniforme, não modelar a zona morta dos motores e as variações no atrito do solo e considerar as rodas indeformáveis. Assumindo que não ocorre escorregamento, que os sensores têm uma boa precisão e que o ambiente é suficientemente plano, modelos cinemáticos são utilizados para simular o comportamento do robô já que estruturas mais simples são utilizadas. A dinâmica dos atuadores é ignorada, ou seja, as velocidades desejadas são atingidas imediatamente. Entretanto, dependendo da velocidade do robô, a dinâmica dos atuadores pode limitar, ou até degradar o desempenho em situações reais.

O objetivo desse trabalho é resolver o problema de controle de seguimento de trajetória para um robô móvel omnidirecional. Existem na literatura, vários trabalhos que tratam esse problema para diferentes tipos de robôs móveis utilizando diferentes técnicas de controle [Hwang and Chang, 2007], [Low and Wang, 2008], [Gang et al., 2005], [Li and Zell, 2007], [Zhao et al., 2009], [Raffo et al., 2009], [Shim et al., 2006], [Anderson and Stone, 2007], [Lim et al., 2008], [Kang and Hedrick, 2009].

O problema de rastreamento com obstáculos é resolvido para um CLMR (*Car Like Mobile Robot* - Robô móvel em forma de carro) em um espaço inteligente usando o controle descentralizado misto H_2/H_∞ [Hwang and Chang, 2007]. Nessa técnica um comando de referência para o controlador é planejado *on-line* através da informação da planta fornecida por uma câmera. Essa câmera fornece a posição e a orientação de um determinado obstáculo. As vantagens dessa técnica são o baixo consumo de energia, com erro de rastreamento pequeno, atenuação de perturbação na saída e um melhor desempenho de controle. Na literatura, o

problema de rastreamento também é resolvido para robôs móveis WMR (*Wheeled Mobile Robot*). Em outro trabalho foi realizado um controle baseado em GPS associado a uma técnica de compensação de escorregamento e derrapagem [Low and Wang, 2008]. Em uma outra abordagem é apresentada uma metodologia de controle adaptativo de falha, também para o robô móvel WMR. Esse controle é baseado em modelo e é utilizado para detecção de faltas e isolação [Ji et al., 2003]. Para robôs móveis diferenciais, o problema de controle de orientação pode ser resolvido através da sincronização de duas rodas, usando um algoritmo de controle adaptativo acoplado. O controlador utiliza a tecnologia de acoplamento cruzado em uma arquitetura de controle adaptativo e garante que os erros de rastreamento e de sincronização das rodas convirjam para zero [Gang et al., 2005].

Existem também trabalhos tratando do problema de controle de robôs omnidirecionais. Um método baseado no modelo cinemático linearizado de entrada-saída também pode ser utilizado. A saturação e a dinâmica dos atuadores degradam o desempenho do controlador, portanto a lei de controle proposta considera esses efeitos e garante a estabilidade do sistema de controle em malha fechada [Li and Zell, 2007]. Também foi proposto um controle das forças internas e do movimento de robôs móveis omnidirecionais, usando um controlador PD [Zhao et al., 2009].

Recentemente, um grande interesse tem sido demonstrado em técnicas de controle preditivo aplicadas ao problema de seguimento de trajetórias para robôs móveis, como pode ser visto em [Raffo et al., 2009], com desvio de obstáculos em [Lim et al., 2008] e [Shim et al., 2006], com diversos mecanismos de locomoção em [Anderson and Stone, 2007] e [Kang and Hedrick, 2009]. O conceito de predição é ideal para aplicações onde se conhecem as referências futuras.

O sucesso do uso de técnicas de Controle Preditivo depende da disponibilidade de um modelo matematicamente preciso do sistema em questão. No caso de navegação de robôs, o atrito é uma das principais fontes de comportamento não-linear, que dificulta a obtenção de um modelo adequado e torna difícil o controle. Neste sentido, vários modelos de atrito têm sido propostos na literatura. Em um trabalho o modelo engloba o comportamento do atrito incluindo o efeito Stribeck, a histerese, características de mola por *stiction*, e atrito estático variante [Canudas de Wit et al., 1995]. Os parâmetros de tais modelos são, entretanto, muito difíceis para estimar com precisão [Ray, 2009]. Mais do que isso, os modelos são bastante complexos, requerendo, como consequência, complexos métodos de projeto de controle.

Um modelo simplificado de um robô omnidirecional de 4 rodas foi usado em [Conceição et al., 2009], onde métodos para estimar os parâmetros do modelo foram propostos e validados através de resultados experimentais. Este modelo também foi usado em [Conceição et al., 2007], onde métodos de otimização numérica foram aplicados diretamente no projeto de um controlador NMPC (*Nonlinear Model Predictive Control*). A principal desvantagem de técnicas de controle preditivo não-linear é o fato de que é necessário resolver um problema de otimização não convexo, o que é uma tarefa complexa em termos de métodos numéricos de otimização, e a existência de ótimos locais.

Em uma aplicação foi utilizada uma técnica de controle preditivo não linear, NMPC, que resolve o problema de seguimento de trajetórias para uma formação de robôs omnidirecionais [Kanjawanishkul and Zell, 2008]. Em outro trabalho o NMPC foi aplicado ao problema de seguimento de trajetória de um robô omnidirecional de 4 rodas. Hoje em dia, com a evolução da capacidade de processamento dos computadores, é possível usar algoritmos de otimização em tempo real. Portanto, no artigo de [Conceição et al., 2007], métodos de otimização numérica são usados para obter a minimização da função custo do controlador. Na literatura utilizou-se o algoritmo MPC (*Model Predictive Control*) para resolver o mesmo problema. Dado o erro do modelo dinâmico obtido a partir do estado do robô e da trajetória de referência, o algoritmo MPC foi aplicado para projetar uma lei de controle que considera explicitamente a taxa de progressão de um veículo virtual a ser seguido ao longo da trajetória. Dados os erros entre a cinemática do robô e a cinemática virtual do veículo, linearizados em torno de pontos de operação, a taxa de progressão pode ser obtida através do MPC. Com esse sistema linearizado variante no tempo, o problema de otimização pode ser transformado em um problema de programação

quadrática (QP). Já que o problema se torna convexo, pode-se garantir uma solução global [Kanjawanishkul and Zell, 2009].

Nesse trabalho será elaborado um algoritmo de controle preditivo linear, o MPC, capaz de controlar um robô móvel omnidirecional de 3 rodas, para que esse robô siga uma trajetória de referência, utilizando-se uma técnica de compensação de atrito. Aqui o controle preditivo é aplicado no controle de velocidade da base móvel e as referências de velocidade são geradas pela cinemática inversa, realizando-se, portanto, um controle em cascata. Foi utilizado um modelo que considera a dinâmica do robô, inclusive atritos viscosos e de Coulomb, na estrutura do controlador preditivo. Os graus de liberdade nas entradas podem ser usados para compensar o atrito, permitindo o uso de controle preditivo linear sob restrições.

Na mesma estrutura em cascata, também foi utilizada uma técnica de controle preditivo não-linear utilizando a ferramenta MPT (*Multi-Parametric Toolbox*) que pode ser utilizada para o controle de sistemas PWA (*Piecewise Affine* - Afim por partes), como é o caso do modelo em estudo. A aplicação de dois algoritmos diferentes tem como objetivo comparar um controlador que calcula a ação de controle *on-line*, ou seja, em tempo real como é o caso do MPC, com um controlador *off-line* como é o caso do controle utilizando a MPT.

1.1 Objetivo

O objetivo desse trabalho é fazer com que um robô omnidirecional de 3 rodas siga trajetórias no espaço. Para isso uma técnica de controle preditivo será utilizada. Visando um bom desempenho do controlador que será desenvolvido, o modelo de simulação que será utilizado deve representar o robô da melhor maneira possível. Uma vez obtidos os parâmetros do modelo, estes foram utilizados para a obtenção das matrizes do espaço de estados que representa o processo. Depois disso, o modelo obtido foi validado através de testes em malha aberta.

Os graus de liberdade nas entradas podem ser usados para compensar o atrito, permitindo o uso do controle preditivo linear sob restrições. A solução do problema de controle preditivo depende da utilização de um algoritmo de otimização, para isso o Algoritmo dos Pontos Interiores foi implementado. Foi utilizado um controle em cascata, onde o controle preditivo é aplicado na malha interna, que realiza o controle de velocidades. Simulações foram realizadas com o objetivo de provar que a técnica consegue resolver o problema proposto e também para verificar a influência dos parâmetros de sintonia no desempenho do controlador.

Uma plataforma experimental deve ser utilizada para implementar o controlador ao robô real. Nessa etapa foi feito um estudo de como é feita a interface do algoritmo de controle preditivo com o robô. Experimentos foram realizados para testar o desempenho do controle aplicado à plataforma.

1.2 Justificativa

Esse trabalho envolve 2 áreas do conhecimento muito importantes na atualidade: a robótica e o controle preditivo. As principais vantagens do controle preditivo, que é muito utilizado na indústria, em relação ao controle clássico, é o fato de que ele trabalha mais facilmente com problemas multivariáveis, trata restrições na dinâmica, que podem ser no estado ou na entrada de controle e tem melhores características de robustez já que se adapta bem a perturbações, não-linearidades e erros de modelagem. Controladores tradicionais como o PID (Proporcional Integral Derivativo) não compensam adequadamente efeitos do atrito nem tratam adequadamente restrições [Maciejowski, 2002]. Como a técnica utilizada é inerentemente discreta é necessária a implementação de um algoritmo de otimização para resolver o problema de controle. Este é resolvido através da minimização de uma função custo.

É necessário um modelo de simulação que seja o mais preciso possível, já que as matrizes do espaço de estados são utilizadas no algoritmo de controle que foi implementado. Por mais

eficiente que seja a técnica de controle, se o modelo não representar bem o processo, o controlador não terá um bom desempenho. Foi utilizado um modelo simplificado para que fosse possível a aplicação de controle preditivo, já que refinar mais o modelo exigiria uma técnica de controle mais complexa.

A opção pelo controle preditivo linear sob restrições utilizando uma técnica de compensação de atrito se deve ao fato de que assim o problema de otimização se torna convexo, ou seja, passará a ter apenas 1 mínimo local, que obviamente coincide com o mínimo global. Aplicar a técnica utilizada diretamente ao modelo não-linear do robô geraria um desempenho pobre do sistema de controle, ao passo de que desconsiderar os efeitos do atrito seria considerar uma condição muito distante da realidade. Por outro lado, utilizar o controle preditivo não-linear exigiria a solução de um problema não-convexo.

A implementação na plataforma experimental se justifica pelo fato de que o sistema real sofre a influência de vários fatores que não são observados na simulação, logo é necessária uma análise profunda do desempenho do controle aplicado ao robô real.

1.3 Capítulos da Dissertação

No capítulo 2, o modelo será descrito em detalhes. Será explicitada a estrutura física do robô, bem como os referenciais utilizados e as equações matemáticas envolvidas. Por fim virá uma breve descrição da estimação dos parâmetros e será feita uma validação do modelo obtido.

No capítulo 3, será feita a descrição do controle preditivo. Serão dadas as bases teóricas dessa técnica, e será explicado como ela foi aplicada ao problema em estudo, tanto para o MPC linear com compensação de atrito como para a implementação que considera a abordagem PWA. Além disso, serão realizadas simulações para analisar o desempenho do controle em cascata aplicado ao modelo obtido no capítulo 2 considerando as duas técnicas, realizando ao fim do capítulo, uma análise comparativa.

No capítulo 4, será descrita a implementação do controle no robô real. Primeiramente o ambiente utilizado, o Lazarus [Baeseman et al., 1999a], e a comunicação com o robô serão descritos. Por fim será feita uma análise dos resultados experimentais, tanto para o seguimento das trajetórias quanto para o controle de velocidade com compensação do atrito e para a compensação de zona morta.

Na conclusão, será feito um resumo das principais contribuições do trabalho bem como uma análise geral dos resultados das simulações e dos experimentos.

No apêndice A, o algoritmo implementado para minimizar a função custo do controle preditivo, o algoritmo dos Pontos Interiores, será estudado.

Capítulo 2

Robô Omni-Direcional

Robôs móveis são capazes de se movimentar e de interagir em ambientes conhecidos ou desconhecidos, dinâmicos ou estáticos, ao contrário dos manipuladores industriais que são presos a uma superfície fixa. Eles podem ser classificados quanto ao ambiente em que se movem:

- Robôs terrestres. Normalmente têm rodas, mas podem possuir duas ou mais pernas;
- Robôs aéreos, mais conhecidos como veículos aéreos não tripulados (UAVs);
- Robôs subaquáticos, mais conhecidos como veículos autônomos subaquáticos (AUVs).

Neste contexto, os robôs móveis são o foco de diversas pesquisas atuais, tanto que a maior parte das grandes universidades e centros de pesquisa possuem um ou mais laboratórios de robótica. Também são encontrados na indústria, em instalações militares, em ambientes de segurança, e como produtos, seja de entretenimento ou para trabalhos domésticos. Dentre os robôs móveis existentes, destacam-se os seguintes tipos:

- CLMR: *Car Like Mobile Robot*. Possui dois eixos com duas rodas cada, as rodas traseiras ficam paralelas ao chassi, e as da frente se movem simultaneamente para um dos lados [Hwang and Chang, 2007];
- Robôs diferenciais: possuem pelo menos um eixo direcional com duas rodas, além de rodas fixas. Para esses robôs, o foco é o controle de sincronização entre as rodas do eixo [Gang et al., 2005];
- Robôs humanóides, que têm formato e modo de locomoção similar ao ser humano;
- Robôs omnidirecionais. Possuem rodas uniformemente distribuídas no espaço e podem realizar movimentos de translação e rotação simultaneamente e independentemente [Conceição et al., 2009].

Para a aplicação de seguimento de trajetórias, escolheu-se o robô móvel omnidirecional. Este robô possui a vantagem de ter os movimentos de rotação e translação independentes [Conceição et al., 2009], ou seja, ele pode se mover a cada instante em qualquer direção, sem reorientação. Ao contrário de robôs não-holonômicos, como robôs em formato de carro, que executam primeiro o movimento de rotação, para depois desempenhar uma velocidade de translação. Portanto os robôs holonômicos possuem um maior poder de manobra [Li and Zell, 2007]. Nos robôs não-holonômicos, algum tipo de restrição é imposta a um ou mais estados do sistema e com isso ele não pode se movimentar em todas as direções do espaço do problema, ao contrário dos holonômicos, que não possuem restrições na cinemática.

Este capítulo descreve a modelagem de um robô móvel omnidirecional de 3 rodas, através da sua dinâmica e cinemática.

2.1 Estrutura do robô omnidirecional

Robôs com esse tipo de locomoção têm se tornado cada vez mais populares em aplicações de robôs móveis, pois possuem vantagens em relação a outros tipos de robô [Kanjawanishkul and Zell, 2008]. O robô utilizado possui 3 rodas suecas mostradas na figura 2.1 e é montado sobre uma base móvel, que acomoda todo o *hardware* do robô, na qual as rodas se distribuem uniformemente espaçadas de 120° como ilustrado na figura 2.2. A cada roda, estão ligados um encoder para medir a sua velocidade e a cada motor, um drive de potência com limitação de corrente que o aciona. Os motores possuem uma caixa redutora que reduz a velocidade por uma razão de 19:1 fazendo com que o eixo da roda tenha um torque maior que o eixo do motor, como pode ser visto na figura 2.3. Em experimentos verificou-se que o robô atinge uma velocidade de navegação de no máximo 2 m/s. O sistema computacional embarcado do robô possui um microcontrolador que controla os sensores e atuadores, como mostra o diagrama de blocos da figura 2.4.



Figura 2.1: Rodas Omnidirecionais.



(a) Base do robô real.

(b) Robô montado.

Figura 2.2: Robô móvel omnidirecional de 3 rodas.

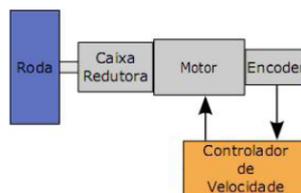


Figura 2.3: Diagrama de blocos do acionamento das rodas.

O primeiro passo para o projeto do controlador é obter um modelo que represente o comportamento da planta. Elementos não-lineares de saturação e atritos relativos a movimentação do robô foram incluídos no modelo.

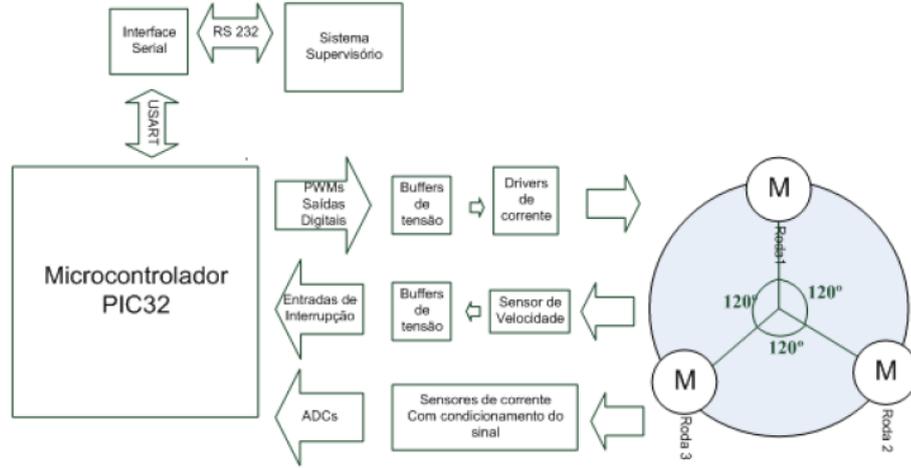


Figura 2.4: Diagrama de blocos do robô.

Os motores de corrente contínua (motor CC), associados às rodas impõem restrições de dinâmica, e os atritos relacionados às forças que provocam os movimentos do robô são responsáveis pelas não-linearidades do modelo, como será visto a seguir.

2.2 Modelo

O modelo em estudo foi obtido baseado na dinâmica e cinemática do robô e nos motores CC. Os elementos não-lineares de saturação são as limitações de corrente nos motores e da tensão aplicada, além do atrito relativo à equação cinemática do robô. Os referenciais do robô e da terra podem ser vistos na figura 2.5.

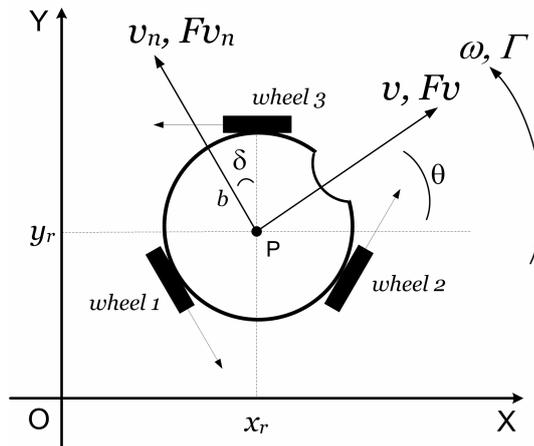


Figura 2.5: referencial do robô.

A conversão entre o referencial do robô e o da terra é feita pela seguinte matriz de rotação, onde θ é o ângulo entre os dois referenciais:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1)$$

2.2.1 Dinâmica do Robô

Pela segunda lei de Newton e considerando o referencial do robô, o modelo dinâmico é dado por [Conceição et al., 2009]:

$$F_v(t) - B_v v(t) - C_v \text{signal}(v(t)) = M \frac{dv(t)}{dt}, \quad (2.2)$$

$$F_{v_n}(t) - B_{v_n} v_n(t) - C_{v_n} \text{signal}(v_n(t)) = M \frac{dv_n(t)}{dt}, \quad (2.3)$$

$$\Gamma(t) - B_w w(t) - C_w \text{signal}(w(t)) = J \frac{dw(t)}{dt}, \quad (2.4)$$

onde F_v e F_{v_n} são as forças de tração no robô relativas a v e v_n que são as velocidades na direção direta e normal ao plano do movimento do robô, respectivamente, Γ é o torque de rotação, $\omega(t)$ é a velocidade angular, M é a massa, J é o momento de inércia, B_v , B_{v_n} e B_w são os coeficientes de atrito viscoso e C_v , C_{v_n} e C_w são os coeficientes de atrito de Coulomb relativos às componentes v e v_n e a w , respectivamente.

Fisicamente, os atritos viscosos representam uma força retardadora. São relações lineares entre as forças de tração e as componentes da velocidade. Os atritos de Coulomb representam uma força retardadora cuja amplitude não varia com a velocidade, mas o seu sinal muda de acordo com o sentido da velocidade.

As relações entre as forças de tração no robô e nas rodas são determinadas pelas relações 2.5, 2.6 e 2.7.

$$F_v(t) = \cos(\delta)(f_2(t) - f_3(t)), \quad (2.5)$$

$$F_{v_n}(t) = -f_1(t) + \text{sen}(\delta)f_2(t) + \text{sen}(\delta)f_3(t), \quad (2.6)$$

$$\Gamma(t) = (f_1(t) + f_2(t) + f_3(t))b, \quad (2.7)$$

em notação matricial tem-se:

$$\begin{bmatrix} F_v \\ F_{v_n} \\ \Gamma \end{bmatrix} = G \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} 0 & \cos(\delta) & -\cos(\delta) \\ -1 & \text{sen}(\delta) & \text{sen}(\delta) \\ b & b & b \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \quad (2.8)$$

onde os $f_i(t)$ são as forças de tração das rodas e b é a distância entre o centro do robô e as rodas. $f_i(t)$ é dada por:

$$f_i(t) = \frac{T_i(t)}{r_i}, \quad (2.9)$$

onde os $T_i(t)$ são os torques rotacionais das rodas e os r_i são os raios das rodas.

Cada roda é acionada independentemente por um motor de corrente contínua. Para cada motor CC, a dinâmica é descrita pelas relações 2.10 e 2.11.

$$u_i(t) = L_{a_i} \frac{di_{a_i}(t)}{dt} - R_{a_i} i_{a_i}(t) + K_{v_i} \omega_{m_i}(t), \quad (2.10)$$

$$T_i(t) = l_i K_{t_i} i_{a_i}(t), 0 \leq i_{a_i}(t) \leq i_{max}, \quad (2.11)$$

onde i refere-se ao motor responsável pelo acionamento da roda i , $i = 1, 2, 3$. A corrente de armadura é limitada para economizar bateria, onde i_{max} é um parâmetro da planta. K_t é a constante de torque, K_v é a constante de tensão e a tensão de armadura é $u(t)$, com restrições do sinal de entrada: $-6V \leq u(t) \leq 6V$ para $t \geq 0$.

Escrevendo as equações do modelo na forma de espaço de estados tem-se:

$$\dot{x}(t) = Ax(t) + Bu(t) + Ksignal(x(t)), \quad (2.12)$$

$$y(t) = Cx(t), \quad (2.13)$$

onde

$$A = -GQK_t l r^{-1} G^T - \begin{bmatrix} \frac{B_v}{M} & 0 & 0 \\ 0 & \frac{B_{v_n}}{M} & 0 \\ 0 & 0 & \frac{B_\omega}{J} \end{bmatrix},$$

$$B = GQ, C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ e } K = \begin{bmatrix} -\frac{C_v}{M} & 0 & 0 \\ 0 & -\frac{C_{v_n}}{M} & 0 \\ 0 & 0 & -\frac{C_\omega}{J} \end{bmatrix}$$

, onde

$$Q = \begin{bmatrix} \frac{l_1 K_{t1}}{M R_{a1} r_1} & 0 & 0 \\ 0 & \frac{l_2 K_{t2}}{M R_{a2} r_2} & 0 \\ 0 & 0 & \frac{l_3 K_{t3}}{J R_{a3} r_3} \end{bmatrix}, K_t = \begin{bmatrix} K_{t1} & 0 & 0 \\ 0 & K_{t2} & 0 \\ 0 & 0 & K_{t3} \end{bmatrix}$$

$$l = \begin{bmatrix} l_1 & 0 & 0 \\ 0 & l_2 & 0 \\ 0 & 0 & l_3 \end{bmatrix} \text{ e } r = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix}.$$

Observando-se a equação 2.12, pode-se notar que a não-linearidade reside no termo $Ksignal(x(t))$. A matriz G já foi definida na equação 2.8.

A função $signal(x)$, que aparece na equação 2.12, é definida pela relação 2.14.

$$signal(x) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \\ -1 & \text{se } x < 0. \end{cases} \quad (2.14)$$

A estimação dos parâmetros dinâmicos é feita baseada nas medidas da corrente nos motores e medidas de velocidade do robô. Elementos não-lineares de saturação e atritos relacionados à equação do movimento do robô foram incluídos no modelo, onde o método dos mínimos quadrados linear foi usado para estimar os coeficientes de atrito viscoso e de Coulomb e o momento de

inércia do robô. Aplicando velocidades v , v_n e ω constantes ao robô, os termos derivativos das equações (2.2)-(2.4) se anulam, e então o problema se reduz a estimação dos parâmetros de 3 equações lineares [Conceição et al., 2009].

O momento de inércia foi estimado através da variação da velocidade com as forças de tração iguais a zero. A estimação dos parâmetros do robô utilizado nesta dissertação foi feita pelo aluno Tiago Ribeiro Trindade do Programa de Pós Graduação em Engenharia Elétrica (PPGEE), utilizando a metodologia definida em [Conceição et al., 2009]. Todos os procedimentos de estimação dos parâmetros para um modelo de robô similar podem ser analisados no trabalho de [Conceição et al., 2009].

Os valores de l , L_a , R_a , K_v e K_t são obtidos das especificações dos motores utilizados. A tabela 2.1 contém os parâmetros do robô utilizado e dos motores.

Tabela 2.1: Parâmetros do modelo [Maxon, 2002].

<i>Símbolo</i>	<i>Descrição</i>	<i>Valor</i>
$B_v(N/m/s)$	coeficiente de atrito viscoso em relação a v	2
$B_{v_n}(N/m/s)$	coeficiente de atrito viscoso em relação a v_n	1,5
$B_\omega(N/rad/s)$	coeficiente de atrito viscoso em relação a ω	0,024
$C_v(N)$	coeficiente de atrito de coulomb em relação a v	1,2
$C_{v_n}(N)$	coeficiente de atrito de coulomb em relação a v_n	0,8
$C_\omega(N.m)$	coeficiente de atrito de coulomb em relação a ω	0,0035
$b(m)$	raio do robô	0,1
$M(kg)$	massa do robô	1,258
$J(kg.m^2)$	momento de inércia do robô	0,025
δ	ângulo	30°
$r_1, r_2, r_3(m)$	raio das rodas	0,035
l_1, l_2, l_3	redução	19:1
$L_{a1...3}(H)$	indutância de armadura	0,00011
$R_{a1...3}(\Omega)$	resistência de armadura	1,66
$K_{v1...3}(Volts/rad/s)$	constantes emf	0,0059
$K_{t1...3}(N.m/A)$	constantes de torque	0,0059

Com os valores dos parâmetros, pode-se calcular as matrizes A, B e K, obtendo:

$$A = \begin{bmatrix} -8,7428 & 0 & 0 \\ 0 & -8,3454 & 0 \\ 0 & 0 & -8,0656 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 1,2894 & -1,2894 \\ -1,4889 & 0,7444 & 0,7444 \\ 9,1906 & 9,1906 & 9,1906 \end{bmatrix} e$$

$$K = \begin{bmatrix} -0,9539 & 0 & 0 \\ 0 & -0,6359 & 0 \\ 0 & 0 & -0,0172 \end{bmatrix}.$$

É importante frisar que o modelo obtido é simplificado, já que as rodas são ditas indeformáveis, não são previstas variações nos coeficientes de atrito e a massa do conjunto robô-base é considerada uniforme.

Para demonstrar o quanto o modelo é fiel ao processo, fez-se experimentos para comparar as saídas em malha aberta do robô real e do modelo para um mesmo sinal de entrada. Já que o sinal de saída é colhido a cada período de amostragem de $50ms$, o espaço de estados foi discretizado como segue:

$$\dot{x}((k+1)Ts) = A_d x(kTs) + \begin{bmatrix} B_d & K_d \end{bmatrix} u_{aum}(kTs)$$

$$y(kTs) = C_d x(kTs) + \begin{bmatrix} D_d & 0 \end{bmatrix} u_{aum}(kTs)$$

, onde:

$$A_d = e^{At},$$

$$B_d = (\int e^{ATs} dt)B,$$

$$K_d = (\int e^{ATs} dt)K,$$

$$C_d = C,$$

$$D_d = D \text{ e}$$

$$u_{aum} = \begin{bmatrix} u(t) \\ signal(x(t)) \end{bmatrix}$$

e supondo que

$$signal(x(kTs)) = signal(x((k+1)Ts)).$$

Com isso, as matrizes assumem os seguintes valores:

$$Ad = \begin{bmatrix} 0,6459 & 0 & 0 \\ 0 & 0,6588 & 0 \\ 0 & 0 & 0,6681 \end{bmatrix},$$

$$Bd = \begin{bmatrix} 0 & 0,0522 & -0,0522 \\ -0,0609 & 0,0304 & 0,0304 \\ 0,3782 & 0,3782 & 0,3782 \end{bmatrix},$$

$$Cd = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ e}$$

$$Kd = \begin{bmatrix} -0,0386 & 0 & 0 \\ 0 & -0,0260 & 0 \\ 0 & 0 & -0,0007 \end{bmatrix}.$$

2.3 Resultados

Para analisar o desempenho do modelo do robô, foi realizado um teste para $u_1 = 0V$, $u_2 = 3V$ e $u_3 = -3V$, com esses valores de tensão, o robô se move na direção de v , já que enquanto a roda traseira permanece parada as rodas dianteiras giram em sentidos opostos com a mesma velocidade, portanto durante todo o teste $v_n = 0m/s$ e $\omega = 0rad/s$. A velocidade do robô foi obtida através dos sensores de velocidade do aparato experimental. Da figura 2.6, pode-se perceber que no regime transitório a velocidade do robô na direção direta foi bem próxima à do modelo de simulação, e no regime permanente a saída do robô oscilou em torno do valor permanente da velocidade do modelo. As oscilações em regime permanente se devem principalmente ao fato de que os coeficientes de atrito do terreno não são constantes, e esse fato não está englobado no modelo da seção anterior.

A figura 2.7 mostra os resultados de simulação e experimentais com $u_1 = -3V$, $u_2 = 1,5V$ e $u_3 = 1,5V$, com esses valores de tensão, o robô se move na direção de vn , já que enquanto a roda traseira gira no sentido anti-horário as rodas dianteiras giram em sentido horário com

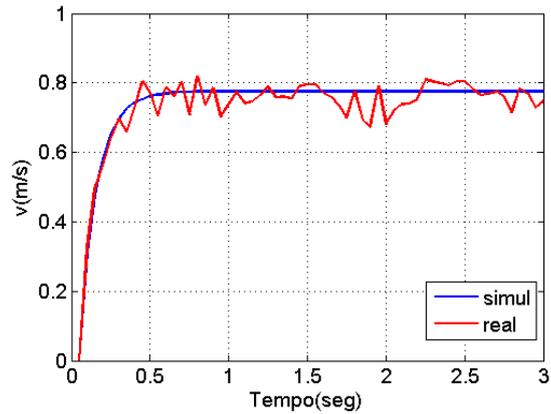


Figura 2.6: Comparação da velocidade v do modelo e do robô real, com $u_1 = 0V$, $u_2 = 3V$ e $u_3 = -3V$.

a mesma velocidade, portanto durante todo o teste $v = 0m/s$ e $\omega = 0rad/s$. Percebe-se que a saída do robô real tem um regime transitório similar ao do modelo e no regime permanente a saída do robô real oscila em torno do valor final da saída do modelo.

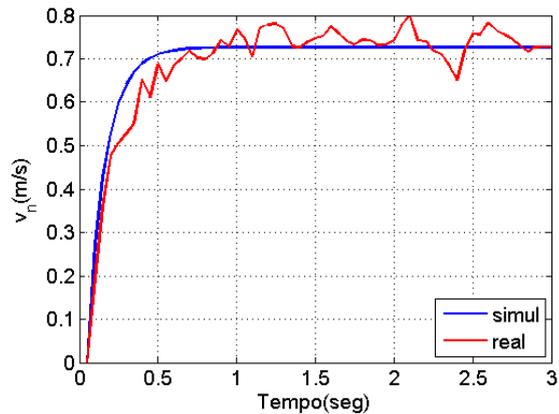


Figura 2.7: Comparação da velocidade v_n do modelo e do robô real com $u_1 = -3V$, $u_2 = 1,5V$ e $u_3 = 1,5V$.

A figura 2.8 mostra os resultados de simulação e experimentais com $u_1 = 1V$, $u_2 = 1V$ e $u_3 = 1V$, com esses valores de tensão o robô gira com ω constante, já que as três rodas giram no sentido horário com a mesma velocidade, portanto durante todo o teste $v = v_n = 0m/s$. Percebe-se que a saída do robô real é bem próxima à do modelo e no regime permanente o ω do robô real oscila em torno do valor permanente do modelo, e essa oscilação tem uma amplitude pequena.

Considerando que o desempenho dos testes experimentais sofre a influência das irregularidades do terreno que geram variações nos coeficientes de atrito, os resultados obtidos para os testes em malha aberta podem ser considerados satisfatórios, já que no regime permanente os valores da saída do robô oscilaram com valor médio próximo do valor em regime permanente da saída do modelo. Portanto esse modelo pode ser considerado uma boa aproximação do robô omnidirecional de 3 rodas.

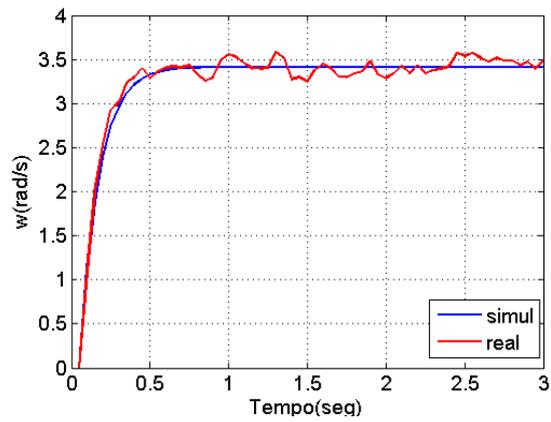


Figura 2.8: Comparação da velocidade ω do modelo e do robô real com $u_1 = 1V$, $u_2 = 1V$ e $u_3 = 1V$.

2.4 Conclusões

Neste capítulo, foi feita uma descrição do robô, depois os métodos de estimação dos parâmetros do modelo de simulação foram descritos, e por fim esse modelo foi validado.

No próximo capítulo, o algoritmo de controle preditivo será aplicado a esse modelo, e será feita uma análise do seu desempenho. Para a aplicação dessa técnica de controle é imprescindível que o modelo utilizado represente bem o processo a controlar, e de acordo com os resultados da validação, esse modelo atende bem a esse critério.

Capítulo 3

Controle Preditivo do Robô Omnidirecional de 3 rodas

Nas últimas décadas, o Controle Preditivo tem sido cada vez mais aplicado na indústria, principalmente no setor petroquímico, e estudado na academia. Os pioneiros no desenvolvimento dessa técnica foram Richalet, em 1978 [Richalet et al., 1978] e Cutler e Ramaker, em 1979 [Cutler and Ramaker, 1979]. Esse capítulo trata dos princípios associados ao controle preditivo como base para a rotina que foi elaborada para implementá-lo. O controle preditivo utiliza as informações do presente e do passado e prevê o futuro, e isso o torna mais eficiente que outras estratégias de controle. As ações de controle são calculadas através da minimização de uma função custo. Os valores futuros da diferença entre a saída e a referência são minimizados e quando o modelo representa fielmente o processo a ser controlado e não está sujeito a distúrbios, esse erro será zerado, a menos que o horizonte de predição seja pequeno o suficiente para instabilizar o sistema. A figura 3.1 mostra a estrutura básica que realiza o controle preditivo.

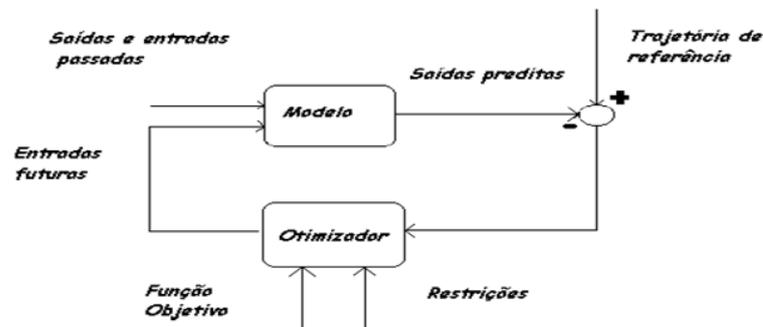


Figura 3.1: Estrutura básica que realiza o controle preditivo.

3.1 Controle Preditivo Linear

A representação de um sistema LTI em espaço de estados é como segue:

$$x(k+1) = Ax(k) + Bu(k), \quad (3.1)$$

$$y(k) = C_y x(k), \quad (3.2)$$

$$z(k) = C_z x(k), \quad (3.3)$$

onde y é o vetor de saídas medidas e z é o vetor de saídas a serem controladas.

O controle aqui implementado, tem como objetivo minimizar a seguinte função custo:

$$V(k) = \sum_{i=H_w}^{H_p} (Q(i)^T (\hat{z}(k+i|k) - r(k+1|k)) Q(i) + \sum_{i=0}^{H_u} (R(i)^T (u(k+i|k) - \bar{u}(k+1|k)) R(i)), \quad (3.4)$$

onde H_p é o horizonte de predição, H_u é o horizonte de controle, H_w é o atraso, considerado zero para o caso aqui estudado, $Q(i)$ é a matriz de ponderação do erro de rastreamento e $R(i)$ é a matriz de ponderação do esforço de controle, ambas no instante i . $Q(i)$ são matrizes simétricas semi-definidas positivas e $R(i)$ são matrizes simétricas definidas positivas. \hat{z} é a saída predita. Deseja-se que \hat{z} convirja para r e u para \bar{u} , valor da entrada no equilíbrio que é obtido da seguinte relação:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \bar{x} \\ \bar{u} \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}, \quad (3.5)$$

onde \bar{x} é o estado de equilíbrio.

A etapa da predição pode ser representada pelas seguintes equações matriciais [Maciejowski, 2002]:

$$\begin{bmatrix} \hat{x}(k+1|k) \\ \dots \\ \hat{x}(k+H_u|k) \\ \hat{x}(k+H_u+1|k) \\ \dots \\ \hat{x}(k+H_p|k) \end{bmatrix} = \begin{bmatrix} A \\ \dots \\ A^{H_u} \\ A^{H_u+1} \\ \dots \\ A^{H_p} \end{bmatrix} x(k) + \begin{bmatrix} B & \dots & 0 \\ AB & \dots & 0 \\ \dots & \dots & \dots \\ A^{H_u} B & \dots & B \\ A^{H_u+1} B & \dots & (A+I)B \\ \dots & \dots & \dots \\ A^{H_p-1} B & \dots & (A^{H_p-H_u} + \dots + A + I)B \end{bmatrix} \begin{bmatrix} \hat{u}(k|k) \\ \dots \\ \hat{u}(k+H_u-1|k) \end{bmatrix}, \quad (3.6)$$

$$\begin{bmatrix} \hat{z}(k+1|k) \\ \dots \\ \hat{z}(k+H_p|k) \end{bmatrix} = \begin{bmatrix} C_z & 0 & \dots & 0 \\ 0 & C_z & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & C_z \end{bmatrix} \begin{bmatrix} \hat{x}(k+1|k) \\ \dots \\ \hat{x}(k+H_p|k) \end{bmatrix}. \quad (3.7)$$

Finalmente, pode-se partir para a solução do problema de controle preditivo propriamente dito, para isso pode-se escrever a função custo como segue:

$$V(k) = (Q_{custo}^T (\hat{Z}(k) - T(k)) Q_{custo}) + (R_{custo}^T (U(k) - \bar{U}(k)) R_{custo}) \quad (3.8)$$

onde:

$$\hat{Z}(k) = \begin{bmatrix} \hat{z}(k + H_w|k) \\ \dots \\ \hat{z}(k + H_p|k) \end{bmatrix}, \hat{T}(k) = \begin{bmatrix} \hat{r}(k + H_w|k) \\ \dots \\ \hat{r}(k + H_p|k) \end{bmatrix},$$

$$\hat{U}(k) = \begin{bmatrix} \hat{u}(k|k) \\ \dots \\ \hat{u}(k + H_u - 1|k) \end{bmatrix},$$

e as matrizes de ponderação Q_{custo} e R_{custo} são dadas por:

$$Q_{custo} = \begin{bmatrix} Q(H_w) & 0 & \dots & 0 \\ 0 & Q(H_w + 1) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & Q(H_p) \end{bmatrix}, \quad (3.9)$$

$$R_{custo} = \begin{bmatrix} R(0) & 0 & \dots & 0 \\ 0 & R(1) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & R(H_u - 1) \end{bmatrix}. \quad (3.10)$$

Substituindo a primeira equação da predição na segunda, tem-se que:

$$\hat{Z}(k) = \Psi x(k) + \Theta \hat{U}(k). \quad (3.11)$$

As matrizes Ψ e Θ são definidas como segue:

$$\Psi = \begin{bmatrix} C_z & 0 & \dots & 0 \\ 0 & C_z & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & C_z \end{bmatrix} \begin{bmatrix} A \\ \dots \\ A^{H_u} \\ A^{H_u+1} \\ \dots \\ A^{H_p} \end{bmatrix},$$

$$\Theta = \begin{bmatrix} C_z B & \dots & 0 \\ C_z A B & \dots & 0 \\ \dots & \dots & \dots \\ C_z A^{H_u} B & \dots & C_z B \\ C_z A^{H_u+1} B & \dots & C_z (A + I) B \\ \dots & \dots & \dots \\ C_z A^{H_p-1} B & \dots & C_z (A^{H_p-H_u} + \dots + A + I) B \end{bmatrix}.$$

A sintonia do controlador preditivo depende dos horizontes de predição e controle (H_p e H_u) e das matrizes Q e R . Quanto menores H_p e H_u mais difícil será o rastreamento, já que o controlador teria que alcançar o objetivo em menos passos. Pôde-se verificar em testes que a partir de determinados valores para os horizontes, um aumento desses valores já não surte mais efeito na sintonia. Quanto menores os elementos da matriz R , mais rápido é o transitório da resposta, já que se pondera menos o esforço de controle e assim o controle gera ações de controle maiores, tornando o controle preditivo mais agressivo. Quanto maior Q , maior o sobre-sinal (diferença entre o valor máximo e o valor final da resposta) e menor o tempo de resposta, ocorre uma maior ponderação do erro, portanto ele converge para zero mais rapidamente e em troca, o sobre-sinal é maior. A redução do valor de R compensa o efeito do sobre-sinal, portanto na indústria normalmente a matriz Q fica constante e a sintonia é feita somente através de R .

Considera-se que as restrições têm o formato de inequações, como segue:

$$F \begin{bmatrix} \hat{U}(k) \\ 1 \end{bmatrix} \leq 0, \quad (3.12)$$

onde $\hat{U}(k) = [\hat{u}(k|k)^T, \dots, \hat{u}(k + H_u - 1|k)^T]^T$, definido analogamente a $U(k)$.
Assumindo F na forma:

$$F = [F_1, F_2, \dots, F_{H_u}, f],$$

onde cada F_i tem dimensões $q \times m$, e f dimensões $q \times 1$, então a equação assume a forma:

$$\sum_{i=1}^{H_u} F_i \hat{u}(k + i - 1|k) + f \leq 0.$$

A ultima relação obtida pode ainda ser exibida da seguinte forma:

$$F\hat{U} \leq -f. \quad (3.13)$$

Para implementar o problema de controle preditivo com restrições precisa-se resolver o seguinte problema quadrático:

$$V(k)_{min} = \min \hat{U}(k)^T H \hat{U}(k) - g_-^T \hat{U}(k), \quad (3.14)$$

onde

$$H = \Theta^T Q_{custo} \Theta \quad \text{e} \quad g_- = 2[\epsilon(k)^T Q_{custo} \Theta + \bar{U}(k)^T R_{custo}],$$

com

$$\epsilon(k) = \hat{T}(k) - \Psi \hat{x}(k),$$

sujeito à restrição de desigualdade da equação 3.13. O problema aqui abordado é convexo, o que garante que o mínimo obtido é global [1].

Para o caso com restrições observa-se que, deve-se ter H_p e H_u maiores à medida que as restrições são mais rigorosas para que se consiga rastrear a referência, já que as restrições limitam as ações de controle. A solução do problema de otimização (QP) pode requerer um grande esforço computacional, já que se trabalha com matrizes de ordem superior, e pode apresentar problemas numéricos a depender de quão rigorosas são as restrições. Esse problema ocorre porque o número de variáveis a otimizar aumenta.

3.2 Controle de Velocidade do Robô

A estratégia do Controle Preditivo caracteriza-se por usar um modelo matemático do sistema a controlar, para obter uma predição das saídas futuras, baseada nos seus valores atuais e passados e nas possíveis ações de controle futuras. Tais ações são calculadas de modo a otimizar um índice de desempenho do sistema [Camacho and Bordons, 2004]. A filosofia de Controle Preditivo ajusta-se, assim, muito bem ao problema de forçar um robô-móvel a seguir uma trajetória pré-definida. Como a referência futura (trajetória que o robô deve realizar no seu ambiente de

navegação) é obtida a partir do conhecimento do modelo cinemático do robô, o controle é capaz de fazer com que o robô reaja antecipadamente a uma mudança na orientação da trajetória.

O esquema de controle aqui proposto baseia-se em uma estrutura em cascata, como mostrado na Figura 3.2. Este esquema é semelhante ao usado em [Liu et al., 2008], onde a malha de controle interna é destinada ao controle da dinâmica do robô, na qual as referências de velocidade são fornecidas pelo controlador da malha externa. Este último é, na verdade, um bloco que implementa a cinemática inversa do robô, que gera as referências de velocidade a partir da medição da posição do robô. O controlador interno deve, então, ser projetado como um controlador de velocidade do sistema, como expresso pela equação 2.12.

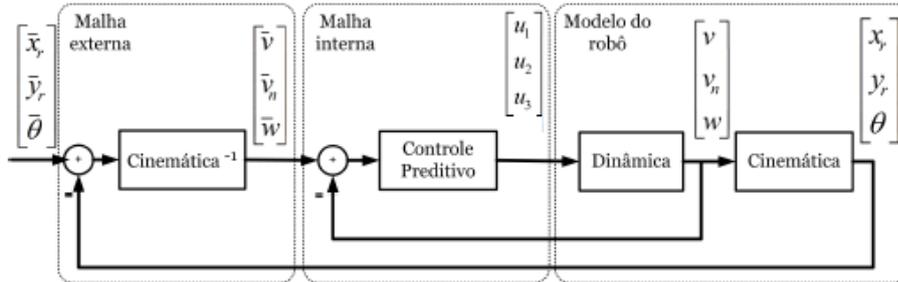


Figura 3.2: Esquema de controle.

Neste trabalho propõe-se o uso de Controle Preditivo na malha interna, o qual apresenta as seguintes vantagens [Camacho and Bordons, 2004, Maciejowski, 2002]:

- Calcula-se uma ação de controle ótima em relação à predição do comportamento futuro do sistema, o que se adequa com perfeição ao rastreamento de trajetórias previamente definidas;
- Restrições nas variáveis de estado e de controle são explicitamente levadas em conta na concepção do controle;
- A ação de controle ótima é re-calculada a cada intervalo de amostragem, baseada em novas medições (princípio do horizonte móvel), o que torna o controle menos sensível a efeitos de perturbações e imperfeições no modelo matemático.

O Controle Preditivo é mais facilmente projetado a partir de modelos de tempo discreto. A discretização do modelo contínuo desenvolvido no Capítulo 2 (relação 2.12) leva ao seguinte modelo de tempo discreto:

$$x(k+1) = A_d x(k) + B_d u(k) + K_d \text{signal}(x(k)). \quad (3.15)$$

Devido a não-linearidades associadas ao termo do atrito de Coulomb em (3.15), a aplicação direta de Controle Preditivo Linear pode resultar em um desempenho pobre do sistema. Por outro lado, técnicas de Controle Preditivo Não-Linear requerem a solução em tempo real de problemas de otimização não-convexos, cujas soluções numéricas são geralmente muito difíceis. Verifica-se, porém, que, considerando o controle de velocidade, robôs omnidirecionais de 3 rodas possuem 3 entradas de controle e apenas 3 variáveis de estado. Portanto, parte dos graus de liberdade no controle pode ser usada para compensar as não-linearidades. Considere-se, então, a seguinte ação de controle:

$$u(k) = u_f(k) + u_c(k),$$

onde f vem de *friction compensation* e c vem de *control*, sendo $u_f(k)$ uma realimentação linearizante, tal que:

$$B_d u_f(k) = -K_d \text{signal}(x(k)). \quad (3.16)$$

Como $B_d \in \mathbb{R}^{3 \times 3}$ e $\text{posto}(B_d) = 3$, a equação acima possui solução única qualquer que seja $x(k)$. A partir desta escolha, a dinâmica do sistema torna-se linear em relação a $u_c(k)$:

$$x(k+1) = A_d x(k) + B_d u_c(k). \quad (3.17)$$

Logo, o Controle Preditivo Linear poderia, em princípio, ser aplicado ao sistema linear (3.17). No entanto, a entrada de controle $u(k)$ é sujeita às seguintes restrições:

$$|u_i(k)| \leq 6, \text{ para } i = 1, 2, 3.$$

Devido ao termo linearizante (3.16), as ações de controle futuras teriam que satisfazer às seguintes restrições não-lineares, ao longo do horizonte de controle H_u :

$$B_d u_f(k+j) = -K_d \text{signal}(x(k+j)) \text{ para } j = 1, \dots, H_u. \quad (3.18)$$

Pode-se contornar esta dificuldade impondo-se a restrição de controle apenas para $j = 0$, visto que $x(k)$ é o vetor conhecido de velocidades medidas no instante k . O vetor de referências de velocidades é fornecido pelo bloco da cinemática inversa.

A função custo do problema de controle preditivo pode ser formulada como segue:

$$V(k) = \sum_{i=0}^{H_p} ((x(k+i) - \bar{x})^T Q (x(k+i) - \bar{x})) + \sum_{i=0}^{H_u} ((u(k+j) - \bar{u})^T R (u(k+j) - \bar{u})) \quad (3.19)$$

sujeito às restrições descritas anteriormente.

O problema de otimização acima é um problema de programação quadrática, para o qual métodos numéricos eficientes são disponíveis. O índice de desempenho é escolhido de modo que o estado x convirja para \bar{x} (estado de referência), o mais rapidamente possível. Para o caso em estudo, tem-se que $\bar{x} = r$, já que a matriz C é a identidade 3×3 (equação (3.5)). No algoritmo proposto, as matrizes $Q(i)$ e $R(i)$ das diagonais principais de Q_{custo} e R_{custo} , vide equações (3.9) e (3.10), são iguais entre si. Conforme discutido anteriormente, para evitar restrições não-lineares no problema de otimização, as restrições na ação de controle foram impostas apenas para o instante k . Do ponto de vista do Controle Preditivo, isto significa a adoção de uma abordagem “otimista”, pois uma ação de controle futura infactível pode ser calculada ($u(k+j)$, $j > 0$), a qual não poderia ser aplicada ao robô. No entanto, as restrições físicas nas tensões de entrada dos motores CC são satisfeitas, visto que apenas a primeira ação de controle ($u(k)$) é de fato aplicada ao robô.

A trajetória de referência é definida como um conjunto de pontos no sistema de coordenadas do mundo (OXY), conforme a Figura 3.3:

$$W(k+j) = [\bar{x}_r(k+j) \ \bar{y}_r(k+j) \ \bar{\theta}(k+j)]^T, \ j = 0, 1, \dots, H_p - 1.$$

Então, a partir da corrente posição e orientação do robô, é necessário calcular as velocidades desejadas para os próximos H_p períodos de tempo. O vetor de velocidades de referência é dado por:

$$\bar{x}(k+j|k) = [\bar{v}(k+j|k) \ \bar{v}_n(k+j|k) \ \bar{w}(k+j|k)]^T, \ j = 0, 1, \dots, H_p - 1,$$

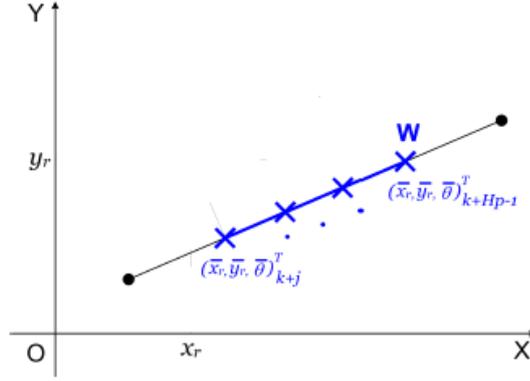


Figura 3.3: Trajetória de referência.

onde j é um passo de predição da velocidade do robô no instante k , e é definido a partir das equações (3.20, 3.21, 3.22). $\epsilon(k) = [x_r(k) \ y_r(k) \ \theta(k)]^T$ é o vetor que representa a localização e a orientação do robô no instante de tempo k e v_{nav} é a velocidade linear de navegação do robô, a qual é um parâmetro.

$$\begin{bmatrix} \bar{v}(k+j|k) \\ \bar{v}_n(k+j|k) \\ \bar{w}(k+j|k) \end{bmatrix} = \begin{bmatrix} \cos(\theta(k)) & \sin(\theta(k)) & 0 \\ -\sin(\theta(k)) & \cos(\theta(k)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix}, \quad (3.20)$$

$$\begin{bmatrix} e_{vx} \\ e_{vy} \\ e_w \end{bmatrix} = \begin{bmatrix} v_{nav} \cos(\varphi) \\ v_{nav} \sin(\varphi) \\ \bar{\theta}(k+j|k) - \theta(k) \end{bmatrix}, \quad (3.21)$$

$$\varphi = \text{atan2}(\bar{y}_r(k+j|k) - y_r(k), \bar{x}_r(k+j|k) - x_r(k)), \quad (3.22)$$

onde $\text{atan2}(y, x)$ é o ângulo em radianos entre o eixo x positivo e o vetor que liga a origem ao ponto (x, y) .

3.3 Simulações do Controle Preditivo

Foram feitas simulações para testar o desempenho do Algoritmo de Controle Preditivo, primeiramente para o controle de velocidades, depois para a compensação de zona morta e finalmente para o seguimento de trajetórias.

3.3.1 Controle de Velocidades

Nesta etapa será feita uma análise dos valores de v , vn , ω , u_1 , u_2 e u_3 . Foram utilizados os seguintes parâmetros de sintonia:

$$R = \begin{pmatrix} 10^{-2} & 0 & 0 \\ 0 & 10^{-2} & 0 \\ 0 & 0 & 10^{-2} \end{pmatrix} \quad \text{e} \quad H_p = H_u = 3.$$

As simulações dessa sub-seção foram feitas com valores diferentes da matriz Q portanto o valor utilizado em cada uma delas está no título das figuras.

Para a referência de $v = 1m/s$, $vn = 0m/s$ e $\omega = 0rad/s$, são obtidos os gráficos da figura 3.4. A referência foi seguida, sem erro de regime perceptível e com tempo de resposta da ordem

de 0, 15s, que corresponde a 3 períodos de amostragem, denotando que o objetivo foi atingido. A tensão u_1 permaneceu em 0V e u_2 e u_3 permanecem em valores iguais em módulo e opostos, como deve ser quando o robô se move para a frente, as duas rodas dianteiras devem girar em sentidos opostos e a roda traseira deve ficar parada.

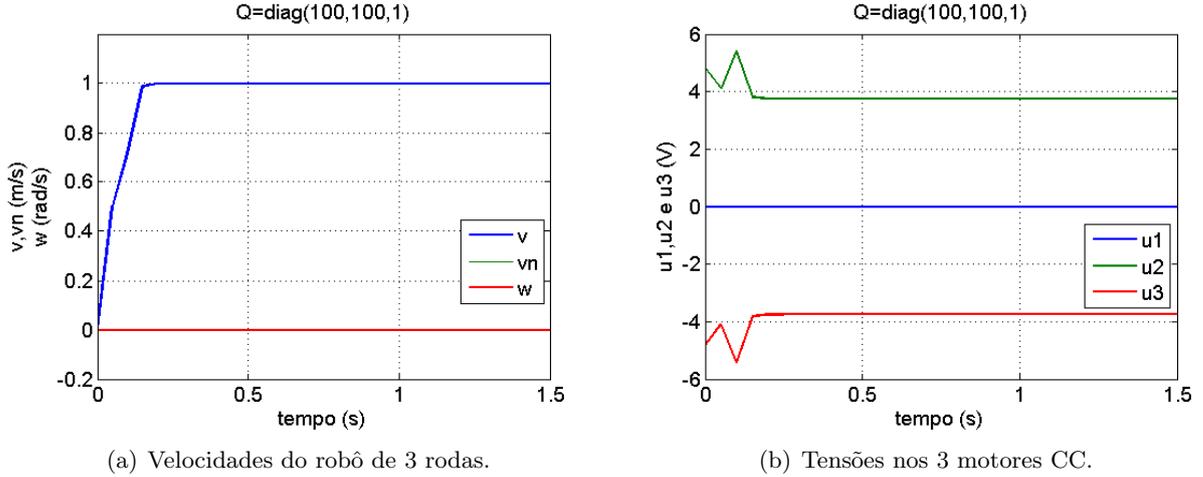


Figura 3.4: Simulação do controle de velocidades, referência $v = 1m/s$, $vn = 0m/s$ e $\omega = 0rad/s$.

Para a referência de $v = 1m/s$, $vn = 1m/s$ e $\omega = 0rad/s$, são obtidos os gráficos da figura 3.5. A referência foi seguida sem erro de regime significativo e com um tempo de resposta de 0,4s. Nessa simulação ω oscila, mas retorna para $0rad/s$ após o regime transitório. Dos gráficos das ações de controle, verifica-se que o motor 2 saturou por alguns períodos de amostragem, portanto as restrições são respeitadas. Com essas velocidades, o robô se move em diagonal.

Para a referência de $v = 1m/s$, $\omega = 1rad/s$ e $vn = 0m/s$, são obtidos os gráficos da figura 3.6. A referência foi seguida com um tempo de resposta de 0,2s. Observou-se que não ocorre saturação dos motores, denotando que essa referência é mais fácil de ser seguida. Com essas velocidades, o robô realiza uma curva em relação ao eixo x.

Para a referência de $v = 1m/s$, $vn = 1m/s$ e $\omega = 1rad/s$, são obtidos os gráficos da figura 3.7. As referências foram seguidas, com erros de regime com valores de 0,006m/s, 0,018m/s e 0,035rad/s para v , vn e ω respectivamente, e observou-se um tempo de resposta de 0,5s. Dos gráficos das ações de controle, pode-se perceber que a tensão no motor 2 saturou ao longo de toda a simulação, o que explica o fato de que os valores das velocidades não se aproximaram ainda mais da referência. Quando a simulação atinge o valor de referência, tem-se $\bar{u} = [-3,72908 \quad 6.06575 \quad -1,45723]^T V$, valor que ultrapassa o limite de tensão de 6V.

Para a grande maioria das referências, o controlador teve um desempenho ideal, de modo que, para estas, não houve um erro de regime significativo. Já para a referência de $v = 1m/s$, $vn = 1m/s$ e $\omega = 1rad/s$, o erro de regime não foi anulado, devido às restrições dos atuadores, isto é, dos motores.

3.3.2 Compensação de zona morta

Nesse trabalho, considera-se principalmente o efeito do atrito estático. Com isso, a resposta em malha aberta possui zona morta. O controlador que foi projetado inclui um termo que compensa o atrito estático, portanto a saída da malha de controle de velocidades deve minimizar a zona morta. Para validar essa suposição foram feitas simulações com o intuito de comparar o desempenho em malha aberta com o desempenho do MPC para cada estado. Para essa série de simulações, foram utilizados os seguintes parâmetros:

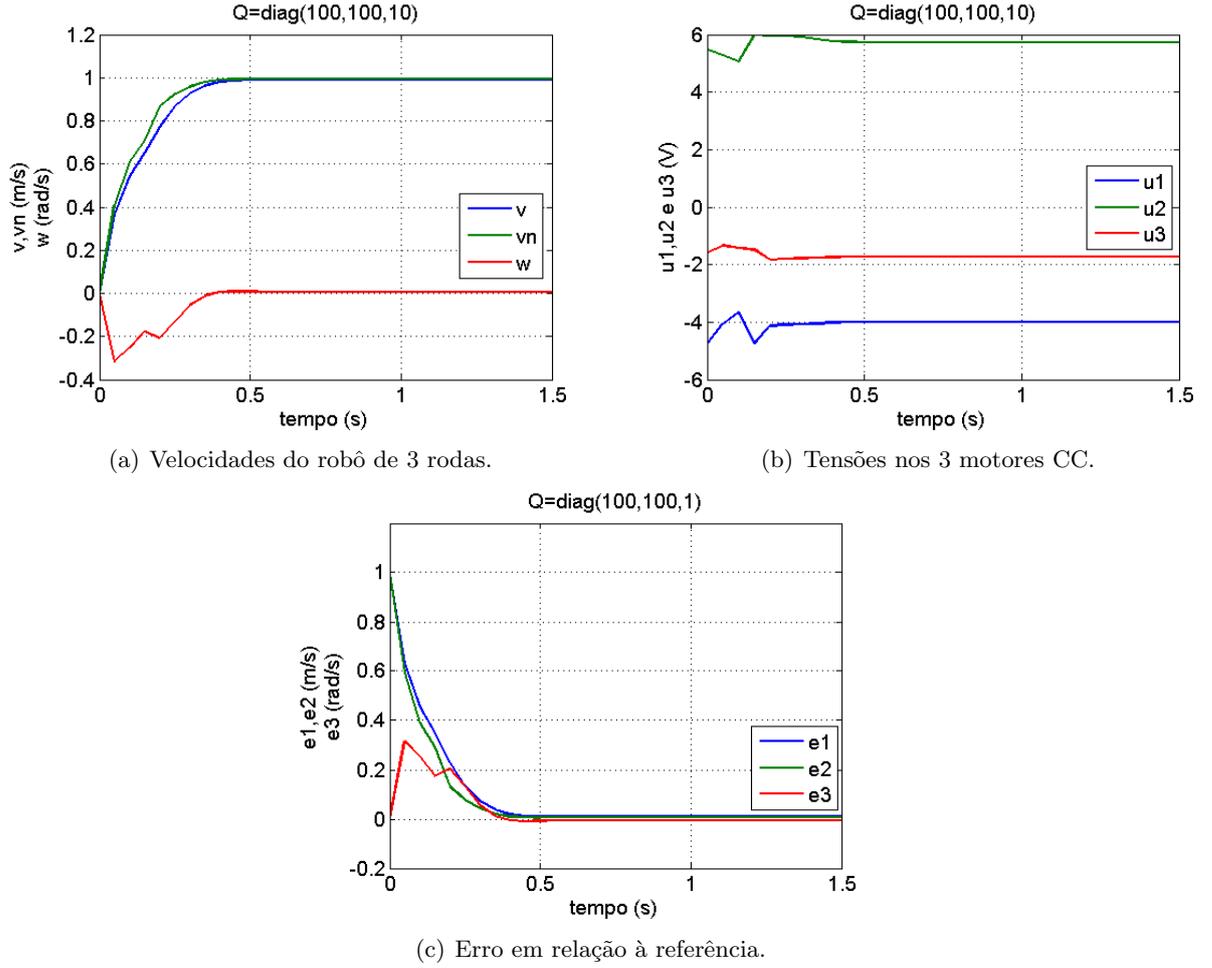


Figura 3.5: Simulação do controle de velocidades, referência $v = 1m/s$, $vn = 1m/s$ e $\omega = 0rad/s$.

$$R = \begin{pmatrix} 10^{-2} & 0 & 0 \\ 0 & 10^{-2} & 0 \\ 0 & 0 & 10^{-2} \end{pmatrix},$$

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$e \quad H_p = H_u = 3.$$

Primeiramente o controlador foi testado para uma referência em rampa, variando v de $-0,6m/s$ até $0,6m/s$ com passos de $0,01m/s$ a cada período de amostragem. Depois, foi feita uma simulação em malha aberta do modelo, com tensões nos motores 2 e 3 entre $-2,4V$ e $2,4V$ e no motor 1 tensão igual a $0V$. Com esses valores de tensão gera-se uma velocidade v semelhante à aplicada ao controlador. Os valores de tensão variaram em passos de $0,04V$ para que se mantivesse uma proporção equivalente. Os resultados das duas simulações podem ser comparados na figura 3.8. Do gráfico da saída, conclui-se que enquanto no caso sem compensação (malha aberta) a zona morta está presente, no caso com compensação (malha fechada) o atrito estático é vencido e o problema de zona morta é resolvido. Para o caso com a compensação da zona morta a tensão varia de uma forma brusca para valores pequenos em módulo, para que o efeito do atrito estático seja compensado.

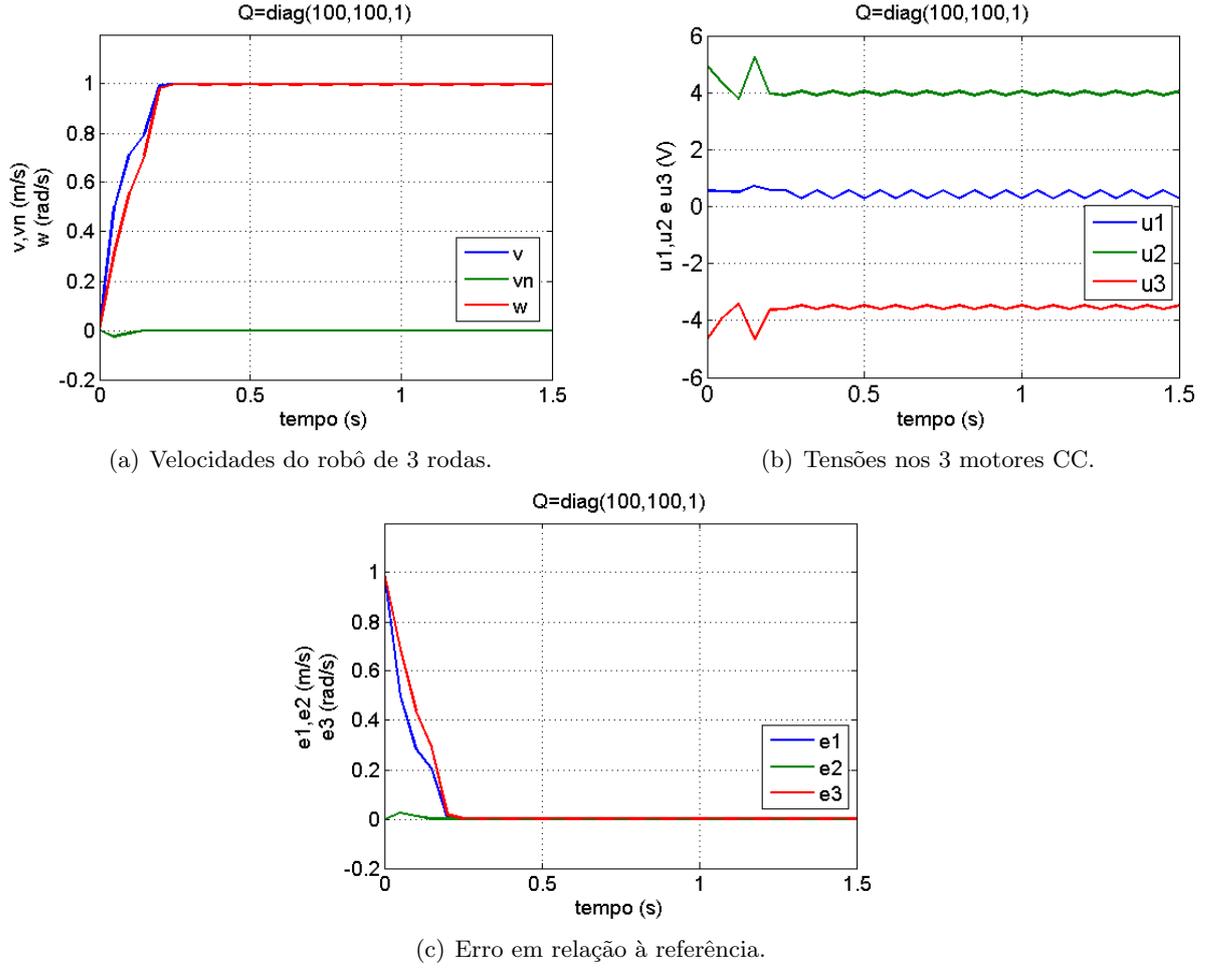


Figura 3.6: Simulação do controle de velocidades, referência $v = 1m/s$, $vn = 0m/s$ e $\omega = 1rad/s$.

Para o caso da velocidade vn repetiu-se um procedimento análogo para realizar as simulações, considerando outras entradas de tensão para que o vn na saída possa variar de $-0,6m/s$ até $0,6m/s$. Também considerou-se uma proporção equivalente entre o passo de velocidade e o de tensão. Os resultados das duas simulações podem ser comparados nas figuras 3.9(a) e 3.9(b). Observações análogas ao parágrafo anterior podem ser feitas para os gráficos em questão.

Para ω o procedimento foi repetido novamente. Os resultados das duas simulações podem ser comparados nas figuras 3.10(a) e 3.10(b). Um efeito semelhante ao ocorrido para os testes com v e vn pode ser observado no gráfico das ações de controle. A zona morta não pôde ser observada no gráfico de ω por um problema de escala. Para limites de tensão de $-0,01V$ até $0,01V$ com passo de $0,0001V$ por período de amostragem, pode-se obter o gráfico da figura 3.10(c).

3.3.3 Seguimento de trajetórias

Para simular o controle escolheu-se primeiramente a trajetória em degrau já que ela tem mudanças bruscas de orientação e direção, que permitem testar o controlador em diferentes condições de operação. A trajetória pode ser observada na figura 3.11. Na primeira e na terceira parte da trajetória a orientação deve ser mantida em $\theta = 0rad$ e na segunda deve ser mantido $\theta = \frac{\pi}{2}rad$. As simulações foram feitas com um tempo de amostragem de $50ms$ com uma velocidade de navegação (v_{nav}) de $0,6m/s$.

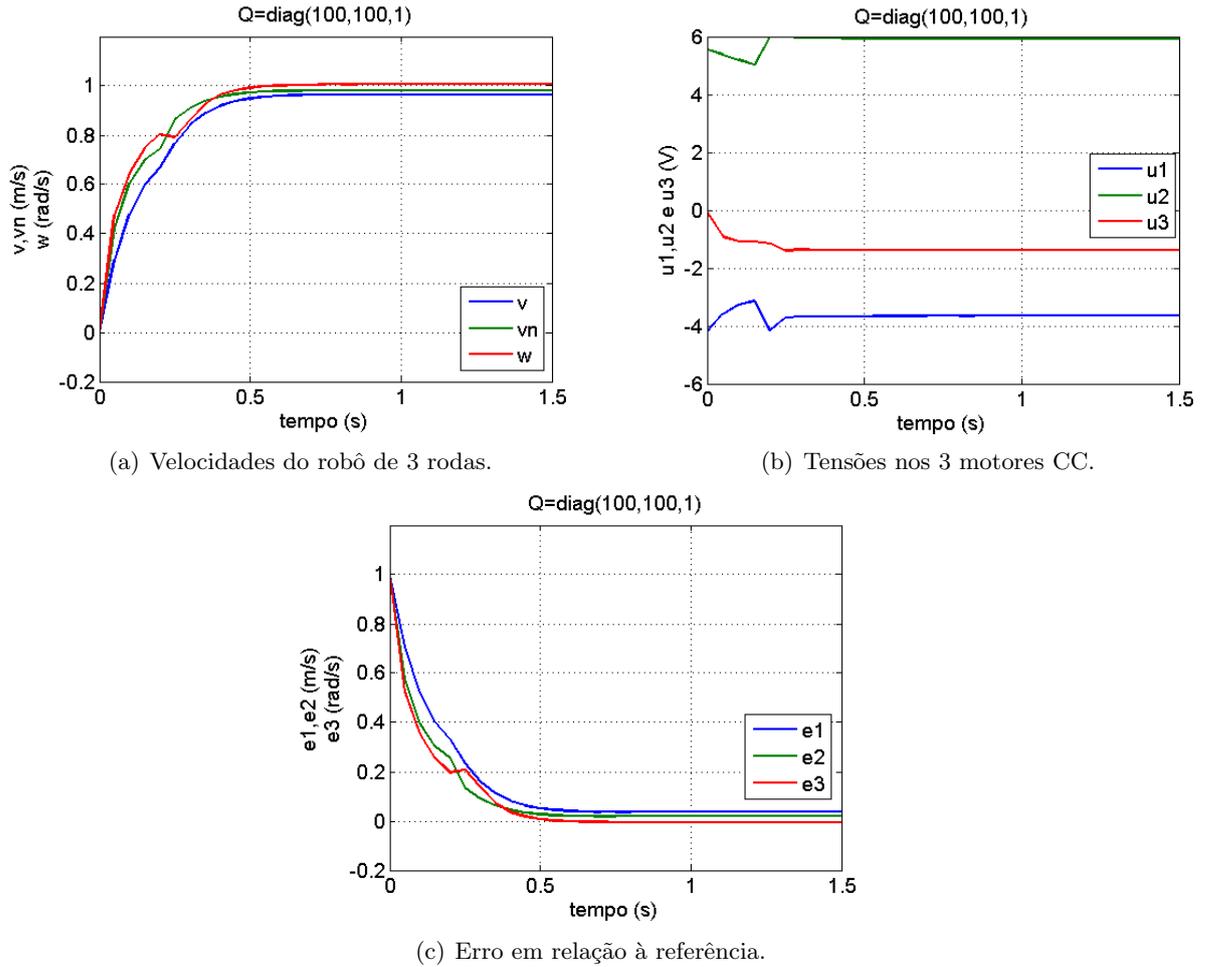


Figura 3.7: Simulação do controle de velocidades, referência $v = 1\text{m/s}$, $vn = 1\text{m/s}$ e $\omega = 1\text{rad/s}$.

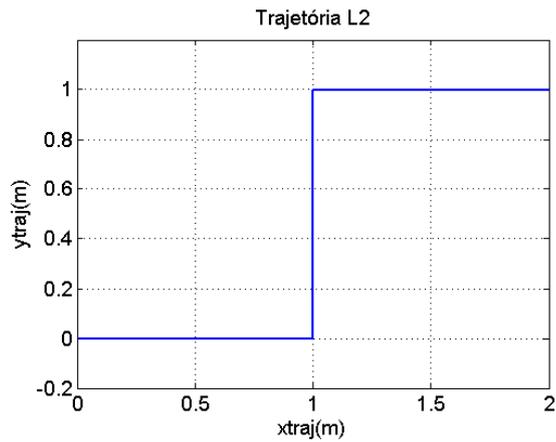
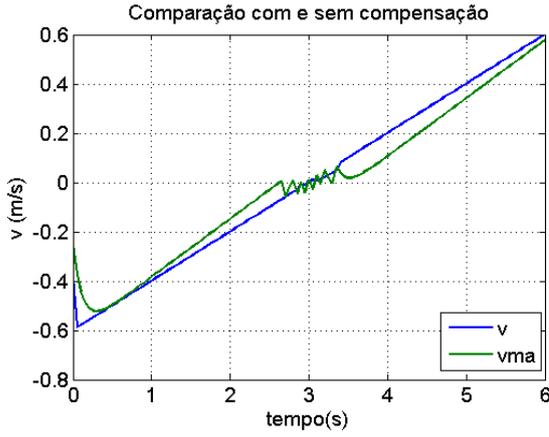
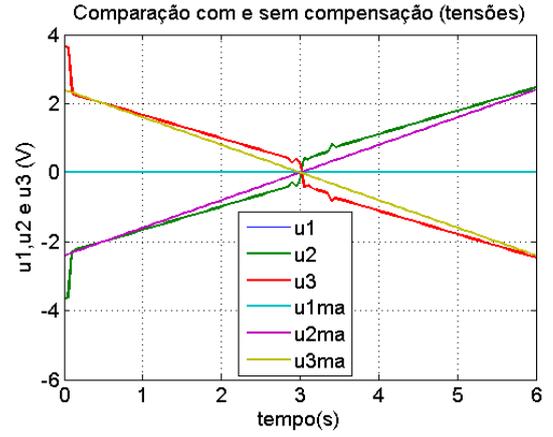


Figura 3.11: Trajetória a ser seguida.

Para todos os testes o valor de Q foi mantido em $10I$, onde I é a matriz identidade 3×3 . Foram realizadas simulações mantendo um determinado valor fixo para R e variando os horizontes de predição e controle, e simulações mantendo H_p e H_u e variando a matriz R . Essas simulações permitem analisar a influência dos parâmetros de sintonia no desempenho do algoritmo MPC e no esforço de controle nos atuadores.

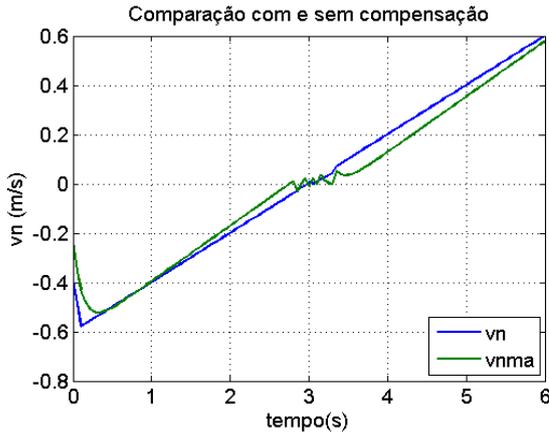


(a) Velocidade v .

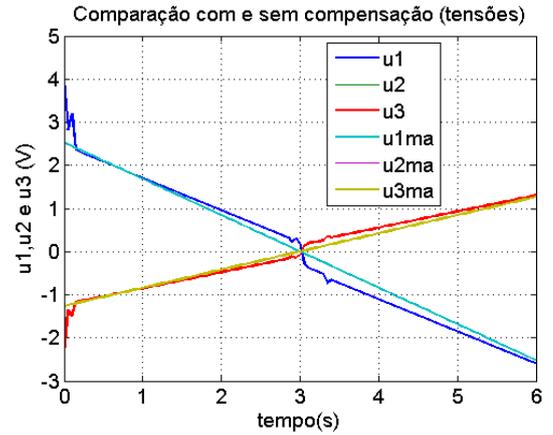


(b) Tensões nos 3 motores CC.

Figura 3.8: Simulação da compensação de zona morta para v (ma - malha aberta).



(a) Velocidade vn .



(b) Tensões nos 3 motores CC.

Figura 3.9: Simulação da compensação de zona morta para vn .

Analisemos primeiramente os gráficos das figuras 3.12 e 3.13, que comparam o desempenho do algoritmo utilizando valores diferentes de H_p e H_u e mantendo $Q = 10I$ e $R = I$ fixos. Do seguimento da trajetória (figura 3.12(a)), percebe-se que quanto maiores os horizontes, maior a capacidade do controlador prever uma mudança brusca de direção e orientação, mas essa diferença não é tão significativa para esses valores de Q e R , portanto o melhor valor é $H_p = H_u = 1$, pois um aumento dos horizontes não compensaria, já que isso aumentaria a ordem das matrizes utilizadas no algoritmo sem uma melhora significativa no desempenho. Dos gráficos das coordenadas e da orientação (figuras 3.12(b), 3.12(c) e 3.12(d)) percebe-se que para $H_p = H_u = 1$ a trajetória é seguida com um tempo de navegação (tempo de execução da trajetória) menor, isso se deve ao fato de que nessa simulação os valores das tensões dos motores são maiores do que para os outros valores de horizontes, portanto o controle é mais agressivo, isso pode ser visto nas figuras 3.12(e), 3.12(f) e 3.13.

Os gráficos das figuras 3.14 e 3.15, mostram os resultados de simulações feitas analogamente às do parágrafo anterior, mas dessa vez mantendo $R = 10^{-2}I$, o que significa um controle mais “agressivo”, no sentido de permitir que as ações de controle sejam maiores. Por esse motivo é que nesse caso a resposta é menos oscilatória do que a das figuras 3.12 e 3.13. Da figura 3.14(a), observa-se que, analogamente ao caso anterior a influência do aumento dos horizontes

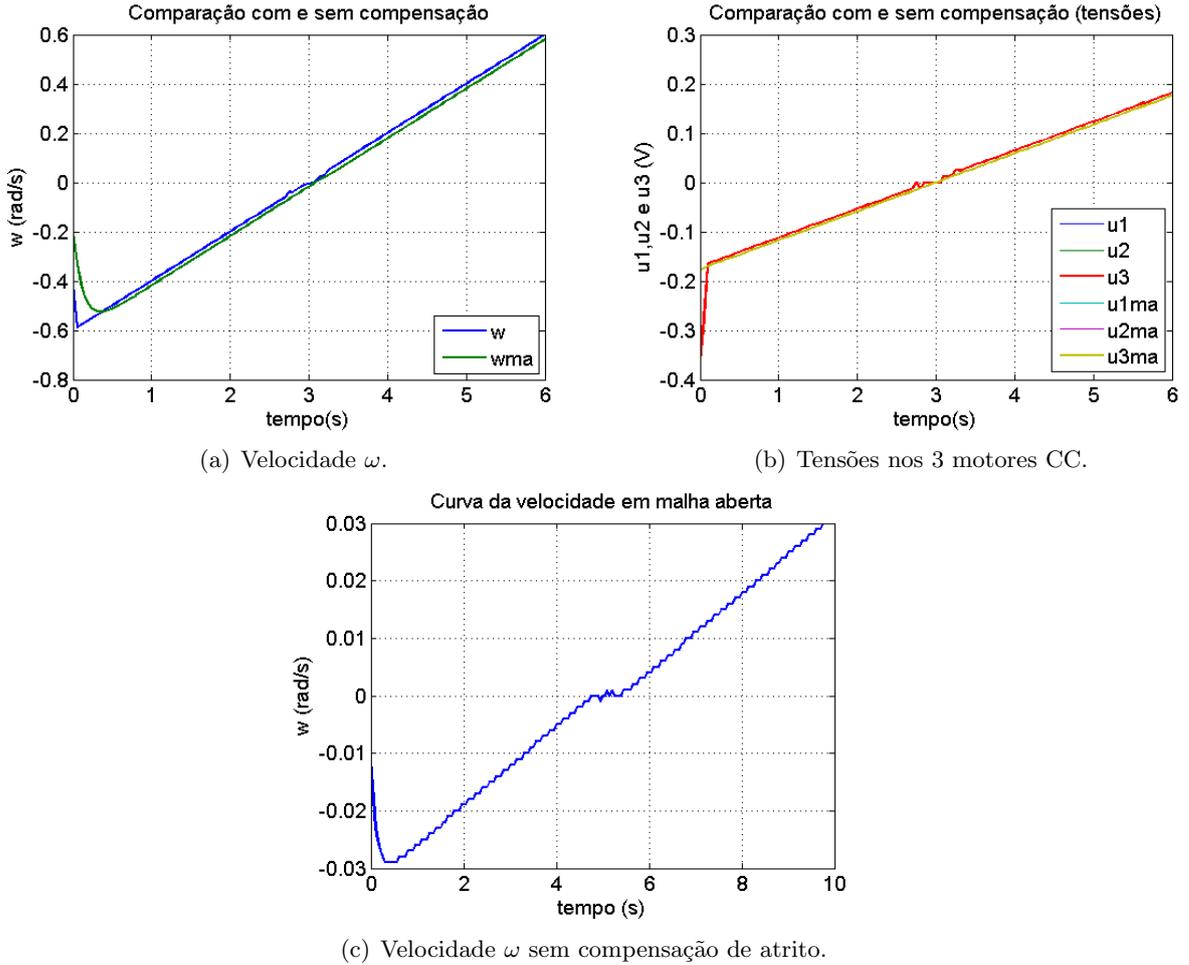


Figura 3.10: Simulação da compensação de zona morta para ω .

no gráfico do seguimento da trajetória não é tão significativa. Dos gráficos das coordenadas e da orientação (figuras 3.14(b), 3.14(c), 3.14(d)), pode-se concluir que para os horizontes de 5 o tempo de navegação foi muito menor do que para os outros dois valores devido à predição, as ações de controle têm uma maior capacidade de prever as ações de controle futuras, conseqüentemente as tensões dos motores, observadas na figura 3.15 são maiores para esse caso, principalmente a partir da segunda curva de 90° .

Nos gráficos das figuras 3.16 e 3.17, foi analisado o desempenho do controlador utilizando como parâmetros $H_p = H_u = 5$, $Q = 10I$ e variando a matriz R . Na figura 3.16(a) observa-se que o seguimento da trajetória foi melhor para o valor de $R = 0,001I$. Do comportamento das coordenadas e da orientação, pode-se notar que para o valor de $R = I$, o tempo de navegação foi menor do que para os outros valores, isso se deve ao fato de que para esse valor as tensões nos motores são maiores no começo da simulação, o que faz com que ele cumpra o objetivo mais rapidamente, como pode ser observado nas figuras 3.17(c), 3.17(d) e 3.17(e). Nota-se também que para valores menores da ponderação, $R = 0,01, 0,001$ por exemplo, os valores de tensão alcançam até $5V$, enquanto que para valores maiores, as tensões alcançam valores de no máximo $3V$.

A figura 3.18 contém os gráficos do seguimento das referências de velocidade utilizando $R = 10^{-2}I$ e $R = 10^{-1}I$, respectivamente, e com $H_p = H_u = 5$. Observa-se que durante toda a simulação o erro de rastreamento é menor para $R = 10^{-2}I$, e o valor de v atinge $0,6m/s$ pouco antes da primeira mudança brusca de direção e orientação. Isso explica o fato do tempo de navegação ser menor para $R = 10^{-2}I$.

Os tempos de navegação para cada conjunto de parâmetros utilizados para a trajetória em degrau, podem ser observados na tabela 3.1.

Tabela 3.1: Parâmetros da simulação e tempo total de navegação.

Parâmetros	Tempo (seg)
$H_p = 1, H_u = 1, Q = 10I, R = 10e^{-3}I$	11,8
$H_p = 3, H_u = 3, Q = 10I, R = 10e^{-3}I$	12,05
$H_p = 5, H_u = 5, Q = 10I, R = 10e^{-3}I$	12,05
$H_p = 1, H_u = 1, Q = 10I, R = 10e^{-2}I$	11,2
$H_p = 3, H_u = 3, Q = 10I, R = 10e^{-2}I$	12,2
$H_p = 5, H_u = 5, Q = 10I, R = 10e^{-2}I$	9
$H_p = 1, H_u = 1, Q = 10I, R = 10e^{-1}I$	9,7
$H_p = 3, H_u = 3, Q = 10I, R = 10e^{-1}I$	11,85
$H_p = 5, H_u = 5, Q = 10I, R = 10e^{-1}I$	10,75
$H_p = 1, H_u = 1, Q = 10I, R = I$	4,95
$H_p = 3, H_u = 3, Q = 10I, R = I$	5,45
$H_p = 5, H_u = 5, Q = 10I, R = I$	5,6

Utilizando como parâmetros $H_p = H_u = 3, R = 0,01I, Q = 100I$ e $v_{nav} = 0,6m/s$, obtém-se, para a trajetória em oito os resultados da figura 3.19. O estado inicial do robô é $x = 1m, y = 0m$ e $\theta = 0rad$, portanto a simulação começa em um ponto que não pertence à trajetória. Das figuras 3.19(a) e 3.19(b) observa-se que o robô alcançou a trajetória rapidamente, e a partir desse ponto o robô segue a trajetória sem erro de posição perceptível. Na figura 3.19(c), observa-se que o limite de $6V$ não foi atingido, portanto o sistema de controle não teve dificuldade para seguir a trajetória em oito. Da figura 3.19(d), conclui-se que as velocidades foram seguidas sem erro perceptível.

Dos testes, pode-se concluir que o desempenho foi satisfatório e que é possível ajustar facilmente os parâmetros de sintonia para a aplicação desejada. Foi possível também analisar a influência dos parâmetros de sintonia no desempenho do algoritmo MPC.

3.4 Controle Preditivo do Robô Omnidirecional de 3 rodas considerado como um sistema afim por partes

O controle de sistemas PWA (Piecewise Afine) inspira cada vez mais interesse da comunidade de pesquisa, já que eles podem aproximar sistemas não-lineares. O controle ótimo para a classe de sistemas em pauta pode ser feito tanto resolvendo um problema de otimização *on-line* ou resolvendo um certo número de problemas multi-paramétricos *off-line*. Por programação multi-paramétrica pode-se resolver tanto o problema linear (mp-LP) quanto o quadrático (mp-QP). A solução tem a forma de uma lei de controle afim por partes pela qual o espaço de estados é particionado em um dado número de regiões e para cada uma delas a lei de controle ótima é uma função afim do estado [Grieder et al., 2004].

A implementação *on-line* se trata de simplesmente avaliar a que região um determinado ponto pertence.

O modelo não-linear em estudo é dado pela seguinte equação de estados:

$$x(k+1) = Ax(k) + Bu(k) + K_d \text{signal}(x(k)), \quad (3.23)$$

Onde a função *signal* retorna 1 se o argumento for positivo, -1 se for negativo e 0 se o argumento também for zero.

Pode-se dividir o espaço de estados da equação (3.23) em 8 regiões nas quais o sistema pode ser tratado como linear, já que o vetor de estados tem 3 elementos. Pode-se então escrever o problema da seguinte forma:

$$x(k+1) = Ax(k) + Bu(k) + K \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \quad (3.24)$$

onde $a_i = 1$ se $x_i > 0$, $a_i = -1$ se $x_i < 0$. e $a_i = 0$ se $x_i = 0$

Pode-se controlar sistemas PWA da seguinte forma:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i, \quad (3.25)$$

$$A_i x(k) + B_i u(k) \leq W_i, i \in I \quad (3.26)$$

se $x(k) \in D_i$

onde D_i é uma região do espaço de estados. O conjunto $I \subset N$ é definido como $I = [1, \dots, I_f]$, onde I_f é o número de dinâmicas.

A solução foi implementada utilizando a ferramenta MPT (*Multi Parametric Toolbox*), com o *software Matlab/Simulink* [Kvasnica et al., 2004]. O objetivo do estudo dessa seção é testar a viabilidade de se usar a solução do pacote MPT para o controle preditivo do robô, para ao final do capítulo comparar com a viabilidade do Controle Preditivo com compensação de atrito.

O problema de controle preditivo pode ser formulado da seguinte maneira [Grieder et al., 2004]:

$$J_N^*(x(0)) = \min_{u_0, \dots, u_{N-1}} \sum_{k=0}^{N-1} (u_k' R u_k + x_k' Q x_k) + x_N' Q_f x_N \quad (3.27)$$

s.a.: $x_N \in X_{set}$,

$L_i x_k + E_i u_k \leq W_i$, se $x_k \in D_i$,

$x_{k+1} = A_i x_k + B_i u_k + f_i$, $x_0 = x(0)$,

onde Q e Q_f são matrizes simétricas semi-definidas positivas e R é uma matriz simétrica definida positiva.

Para o caso em estudo escolheu-se trabalhar com a norma quadrática, já que o resultado deve ser comparado com o obtido para um algoritmo de controle preditivo adaptado para considerar as não-linearidades do robô descrito na seção 2.2. Na ferramenta MPT a função objetivo a ser otimizada é dada por:

$$\min_{u(0), \dots, u(N-1)} x(N)^T P_N x(N) + \sum_{k=0}^{N-1} (u(k)^T R u(k) + x(k)^T Q x(k)) \quad (3.28)$$

s.a.: $x(k+1) = f_{dyn}(x(k), u(k), w(k))$

$u_{min} \leq u(k) \leq u_{max}$

$\Delta u_{min} \leq \Delta u(k) \leq \Delta u_{max}$

$y_{min} \leq g_{dyn}(x(k), u(k)) \leq y_{max}$

$x(N) \in T_{set}$ [Grieder et al., 2004]

A ferramenta utilizada pode implementar controladores *on-line* e *off-line*, dependendo do objetivo da aplicação, no caso em estudo optou-se pela abordagem *off-line* [Kvasnica et al., 2004].

A ferramenta MPT é relativamente fácil de ser utilizada, porém não se conseguiu trabalhar com horizontes maiores do que 1, já que isso exigiu um esforço computacional proibitivo. Nesse trabalho, a lei de controle calculada para o sistema PWA em questão foi aplicada na mesma estrutura em cascata da seção 3.2, para que se pudesse realizar uma comparação das duas técnicas utilizadas para o controle do robô omnidirecional. As simulações foram realizadas no software Matlab com um período de amostragem de 40 ms.

Primeiramente foram feitas simulações do controle de velocidade. Foram utilizados os seguintes parâmetros de sintonia:

$$R = \begin{pmatrix} 10^{-3} & 0 & 0 \\ 0 & 10^{-3} & 0 \\ 0 & 0 & 10^{-3} \end{pmatrix},$$

$$Q = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

$$\text{e } H_p = H_u = 1.$$

Foi dada uma referência de $v = 1m/s$, $vn = 0m/s$ e $\omega = 0rad/s$. A figura 3.20 mostra como as regiões do controlador se distribuem no espaço de estados. A figura 3.21 mostra os gráficos das velocidades na saída e das 3 tensões na entrada do modelo tratado na forma PWA. O seguimento da referência constante em $v = 1m/s$ apresenta um offset de 1% no regime permanente, um pequeno tempo de resposta e não apresenta sobre-sinal. Percebe-se do gráfico das tensões dos motores que o controle foi agressivo já que as tensões no segundo e no terceiro motor atingem 6V em módulo no início da simulação, esbarrando na restrição sobre as ações de controle.

Depois foi feita uma simulação para a referência de $v = 0m/s$, $vn = 1m/s$ e $\omega = 0rad/s$. Da figura 3.22 percebe-se que o seguimento da referência constante em $vn = 1m/s$ também apresenta um offset de 1% no regime permanente, um pequeno tempo de resposta e não apresenta sobre-sinal. Percebe-se do gráfico das tensões dos motores que a tensão no primeiro motor atinge 6V em módulo no início da simulação.

Finalmente foi dada uma referência de $v = 0m/s$, $vn = 0m/s$ e $\omega = 1rad/s$. Da figura 3.23 percebe-se que o seguimento da referência constante em $w = 1rad/s$ não apresenta erro de regime permanente, possui um pequeno tempo de resposta e não apresenta sobre-sinal. A análise das ações de controle permite afirmar que o seguimento dessa referência não exige muito do controlador, esse é o motivo do erro de regime zerar e das primeiras ações de controle não esbarrarem na restrição.

Mantendo-se os parâmetros, foi feita uma simulação para avaliar o desempenho do controle de posição e comparar com o desempenho do algoritmo MPC. A simulação foi realizada para uma trajetória em formato de retângulo, já que esta apresenta mudanças bruscas de direção e orientação.

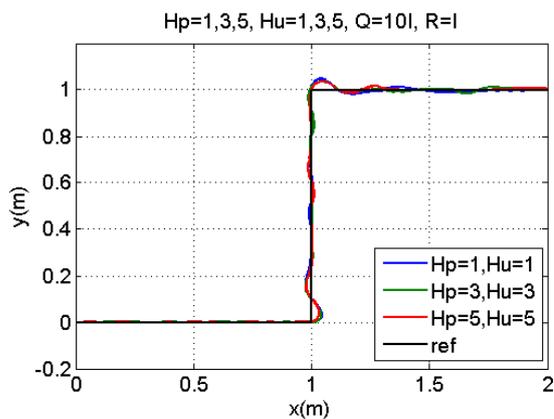
A figura 3.24(a) compara o seguimento da trajetória para as duas técnicas utilizadas, e pode-se perceber que o desempenho é bem parecido. As figuras 3.24(b), 3.24(b) e 3.24(b) mostram as tensões nos 3 motores CC, pode-se verificar que os valores das ações de controle utilizando as duas técnicas são bem parecidos e que o tempo de resposta é um pouco maior para o controle preditivo do modelo PWA. Nas figuras 3.25(a), 3.25(b) e 3.25(c) os valores das coordenadas x e y e o valor da orientação θ foram separados e pode-se perceber que o controle projetado utilizando a ferramenta MPT atuou atrasado em relação ao MPC.

Optou-se por aplicar no robô, o algoritmo MPC, já que para a outra técnica não se conseguiu trabalhar com horizontes maiores do que 1, pois para esse modelo, é exigido um esforço computa-

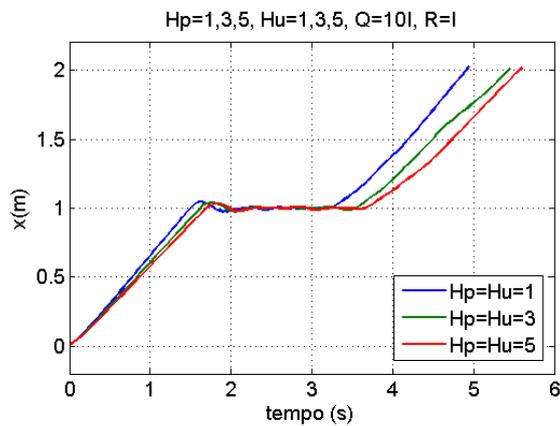
cional proibitivo para calcular leis de controle com horizontes maiores utilizando a ferramenta MPT. Uma outra razão é o número excessivo de regiões (figura 3.20), na prática seria difícil saber a qual região um determinado ponto pertence e esse número de regiões aumenta, com o aumento dos horizontes. O problema de otimização do MPC foi modificado para compensar o atrito, como descrito na seção 3.2.

3.5 Conclusões

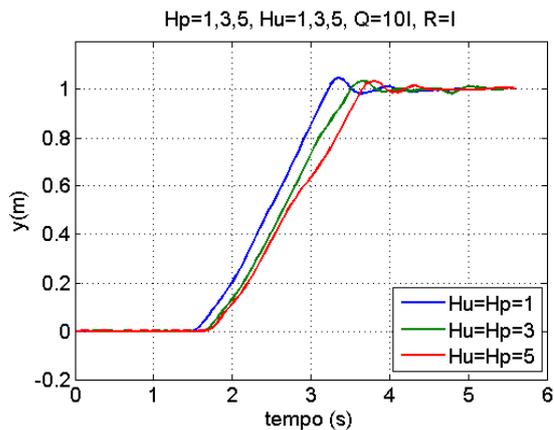
Nesse capítulo a teoria de controle preditivo foi apresentada e foram analisadas simulações do algoritmo MPC e da lei de controle projetada com o auxílio da ferramenta MPT. Para o algoritmo MPC os resultados obtidos foram satisfatórios para as simulações de controle de velocidades, compensação de zona morta e seguimento de trajetória. Para esse último conjunto de simulações foi feito um estudo da influência dos parâmetros de sintonia no desempenho do controlador. O desempenho da ferramenta MPT também foi satisfatório, porém as simulações para horizontes maiores do que 1 exigem um grande esforço computacional para essa aplicação, o que impediu uma análise da influência dos parâmetros de sintonia para a lei de controle afim por partes que foi obtida.



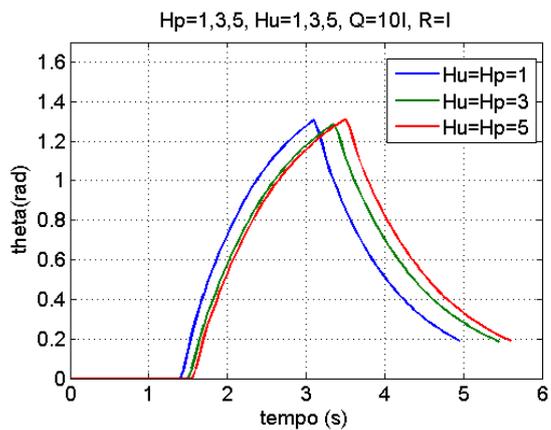
(a) Seguimento da trajetória degrau.



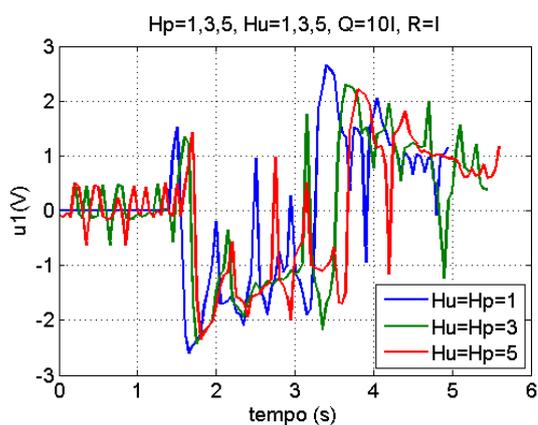
(b) Coordenada x .



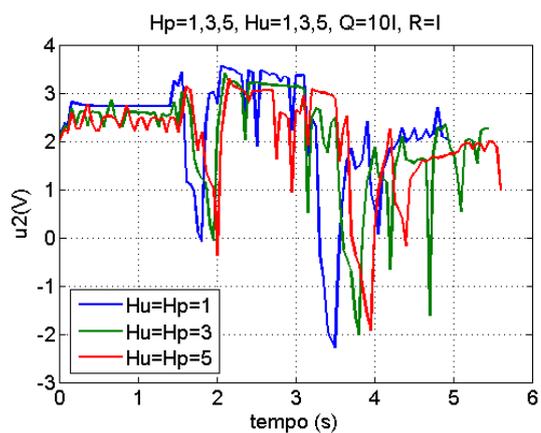
(c) Coordenada y .



(d) Orientação θ .



(e) Tensão no motor 1.



(f) Tensão no motor 2.

Figura 3.12: Simulação do seguimento da trajetória degrau, $R = I$.

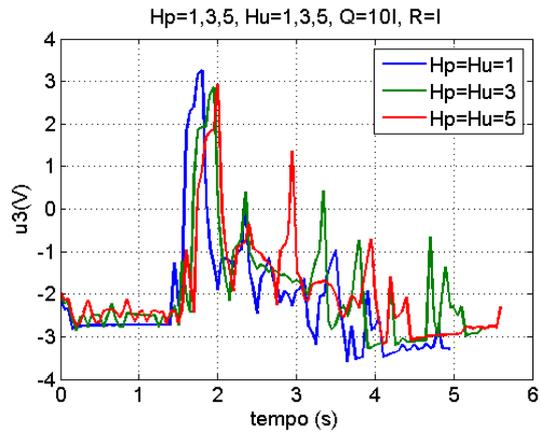


Figura 3.13: Tensão no motor 3. Continuação da figura 3.12.

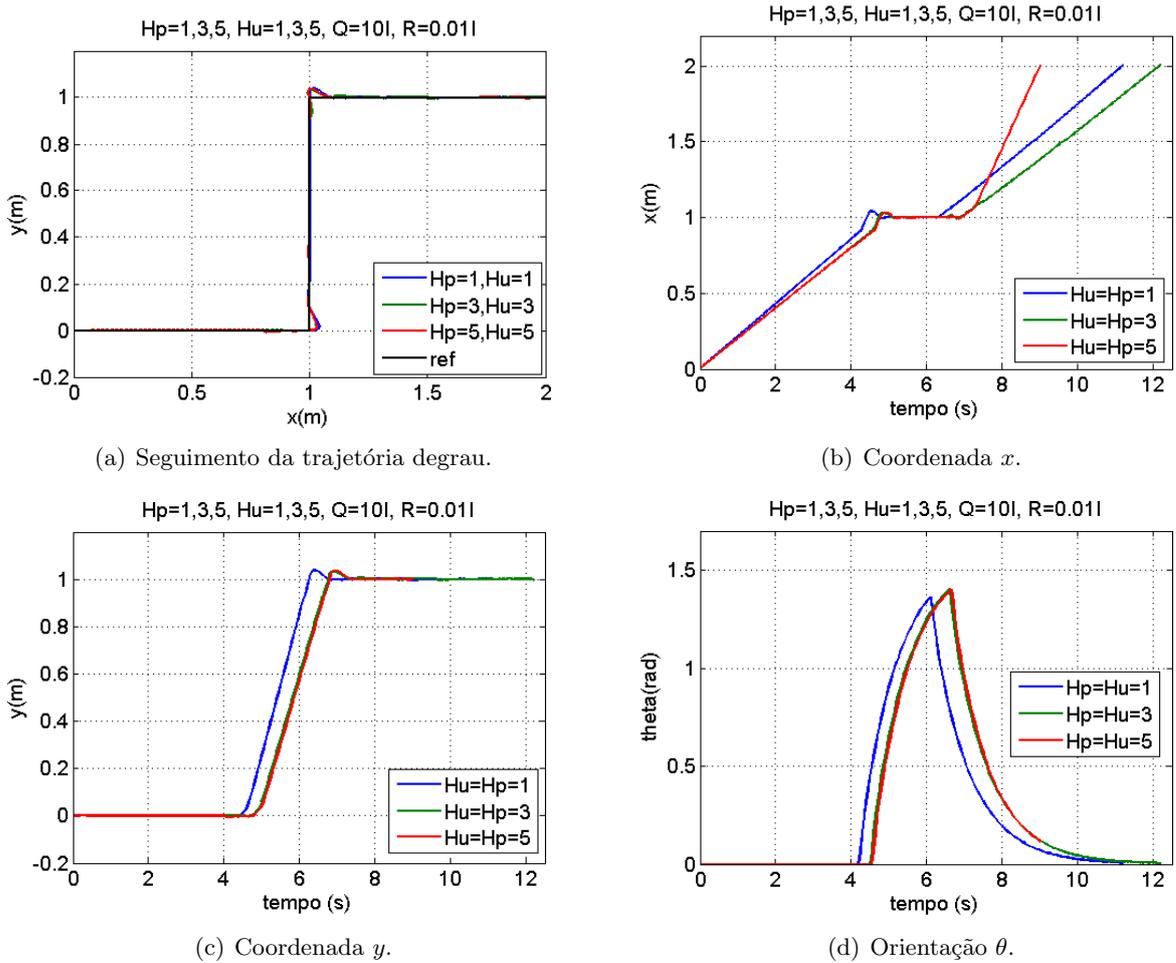
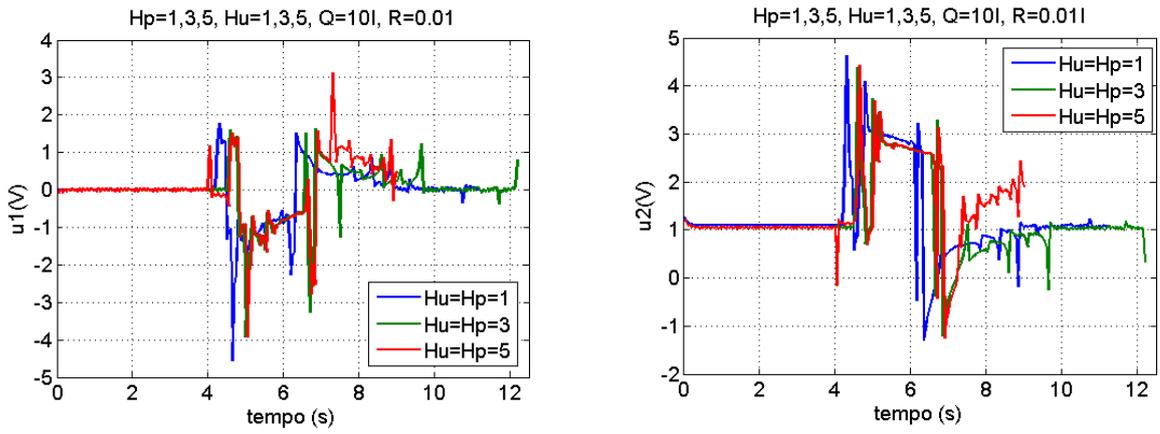
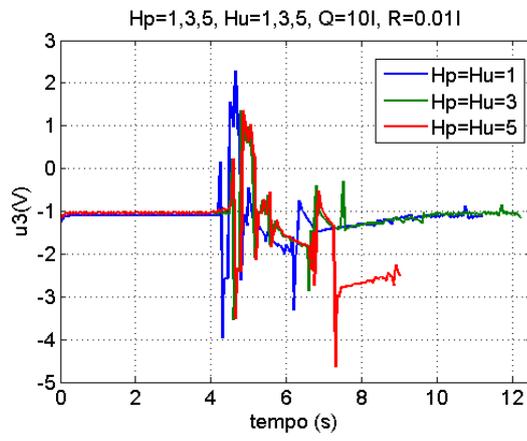


Figura 3.14: Simulação do seguimento da trajetória degrau, $R = 10^{-2}I$.



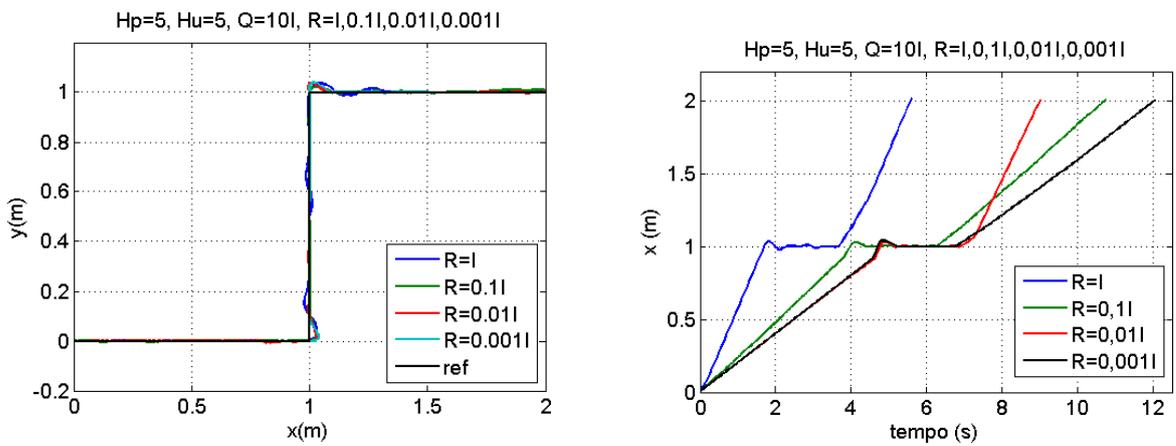
(a) Tensão no motor 1.

(b) Tensão no motor 2.



(c) Tensão no motor 3.

Figura 3.15: Simulação do seguimento da trajetória degrau, $R = 10^{-2}I$, tensões nos motores CC.



(a) Seguimento da trajetória degrau.

(b) Coordenada x .

Figura 3.16: Simulação do seguimento da trajetória degrau, $H_p = H_u = 5$.

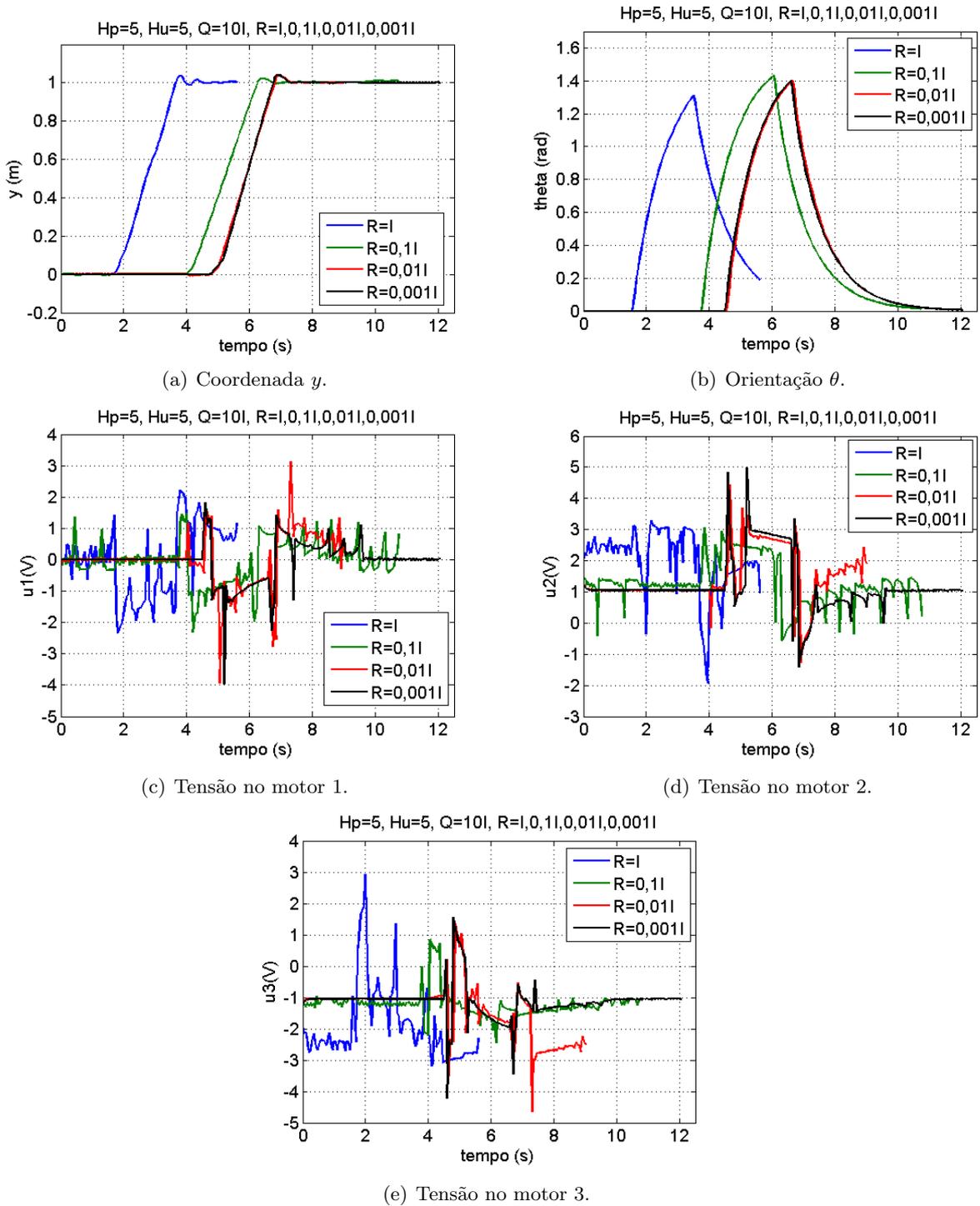


Figura 3.17: Simulação do seguimento da trajetória degrau, $H_p = H_u = 5$, continuação.

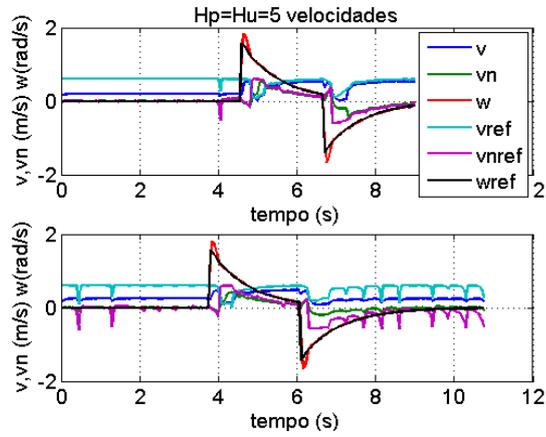
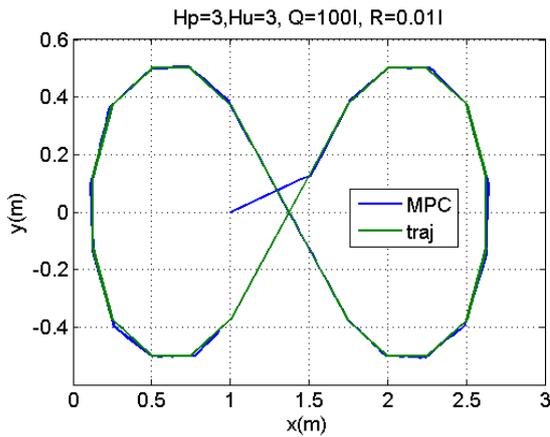
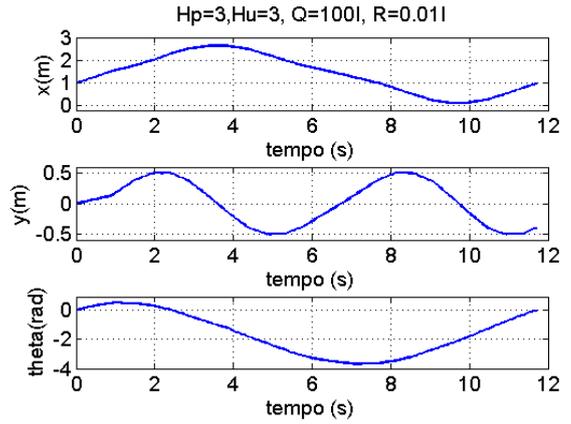


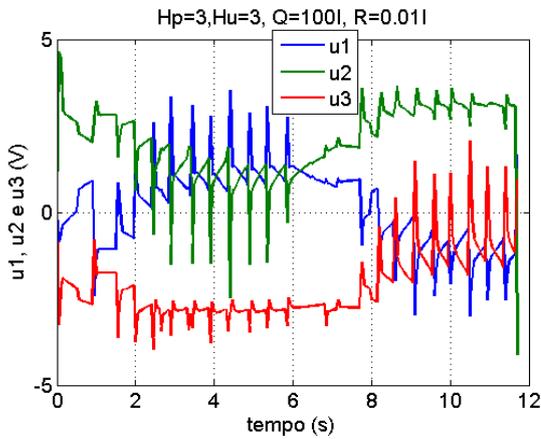
Figura 3.18: Seguimento das referências de velocidade, $H_p = H_u = 5$, para $R = 10^{-2}I$ (gráfico de cima) e $R = 10^{-1}I$ (gráfico de baixo).



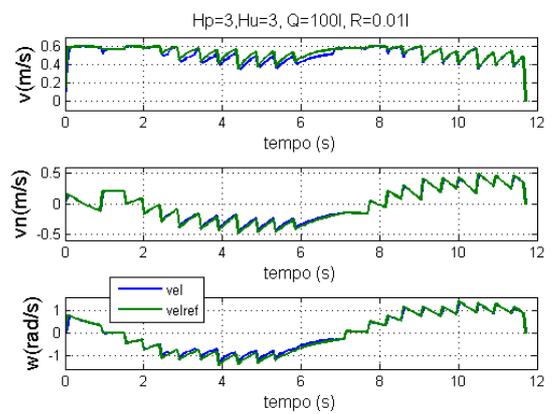
(a) Seguimento da trajetória em oito.



(b) Coordenadas x e y e a orientação θ .



(c) Tensões nos 3 motores CC.



(d) Seguimento das referências de velocidade.

Figura 3.19: Simulação do seguimento da trajetória em oito.

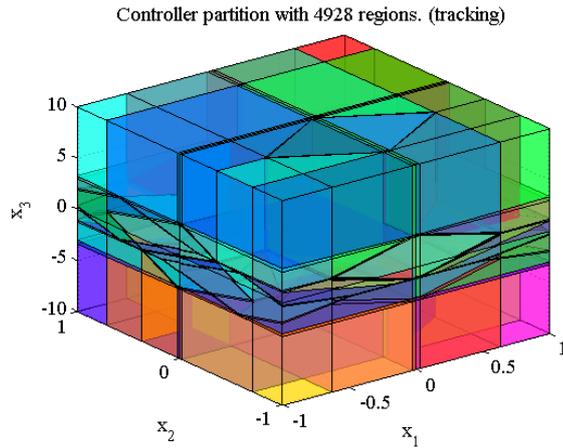


Figura 3.20: Representação geométrica do controlador no espaço de estados.

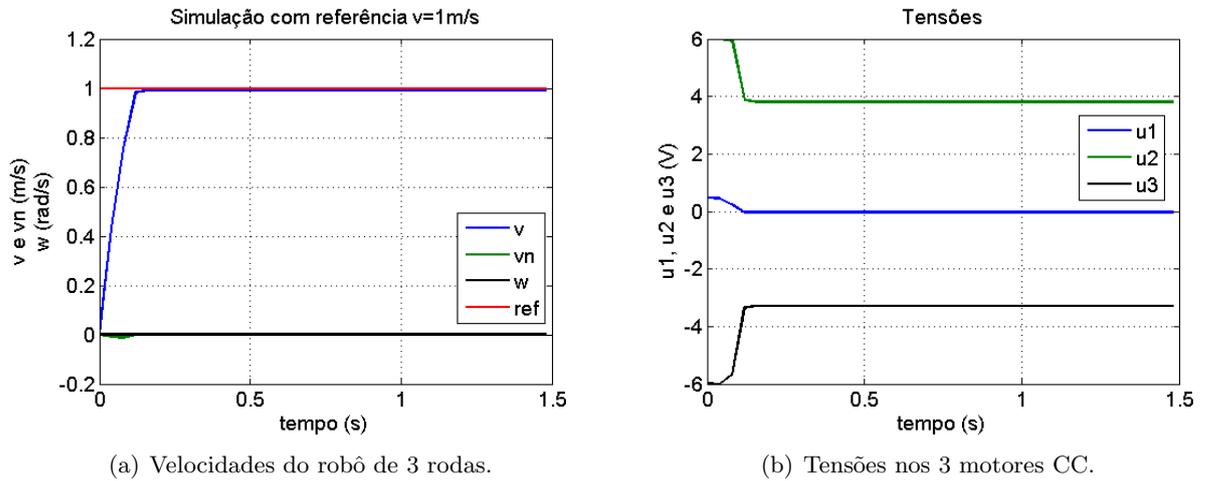


Figura 3.21: Simulação do controle do robô visto como um sistema afim por partes (PWA-*piecewise affine*), referência $v = 1\text{m/s}$.

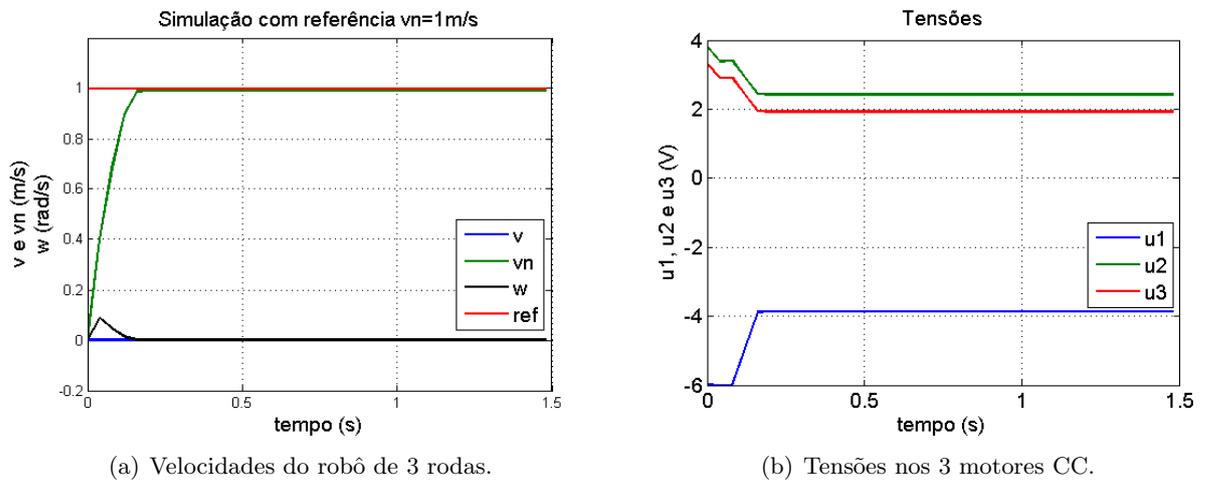
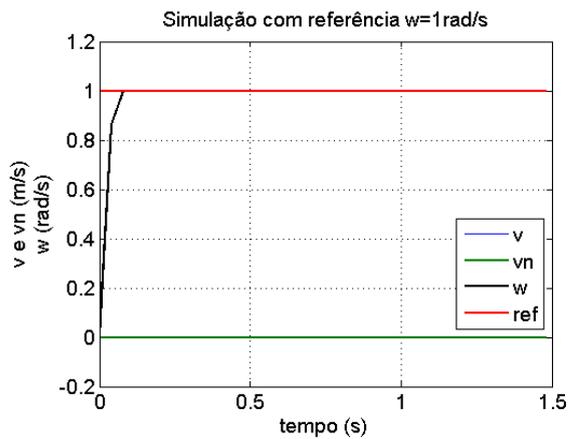
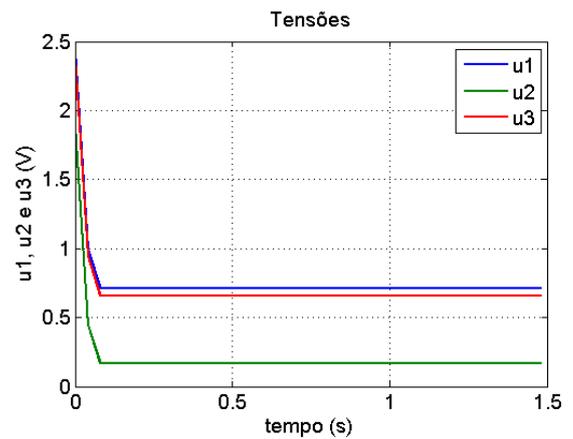


Figura 3.22: Simulação do controle do robô visto como um sistema PWA, referência $vn = 1\text{m/s}$.

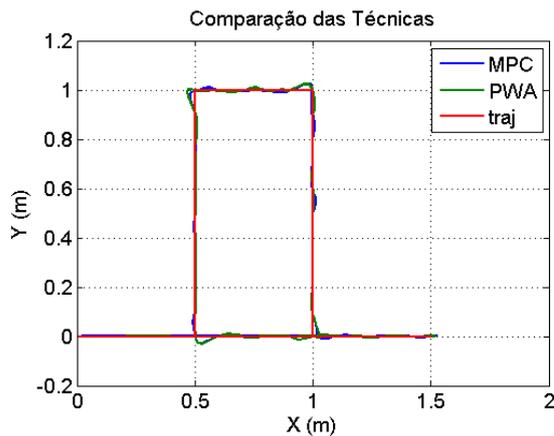


(a) Velocidades do robô de 3 rodas.

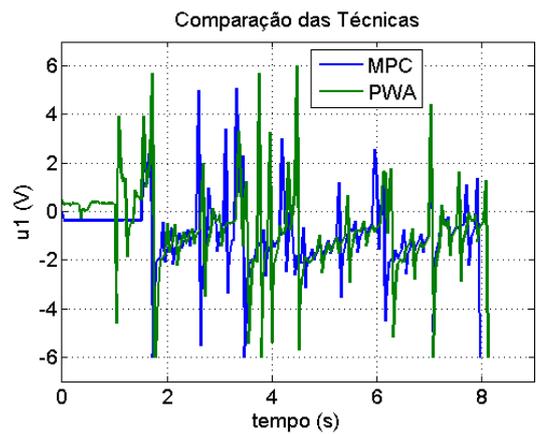


(b) Tensões nos 3 motores CC.

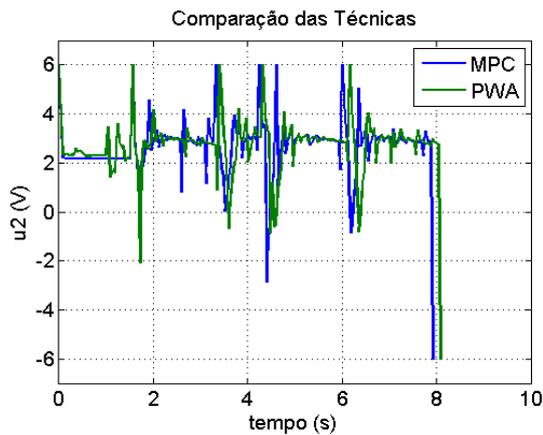
Figura 3.23: Simulação do controle do robô visto como um sistema PWA, referência $\omega = 1rad/s$.



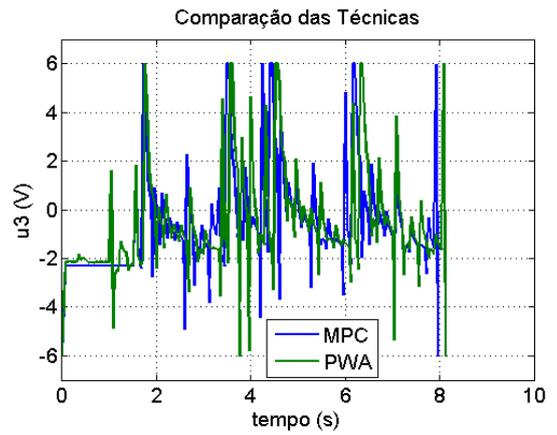
(a) Seguimento da trajetória degrau.



(b) Tensão no motor 1.

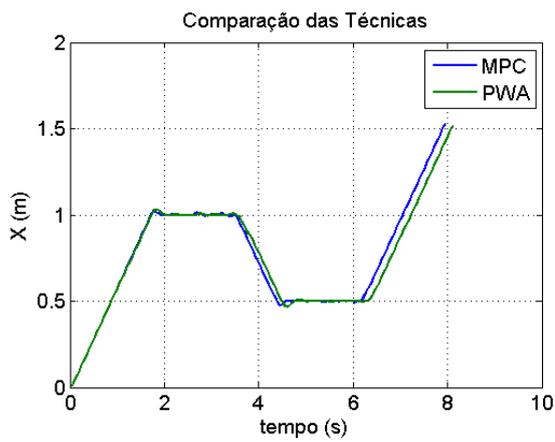


(c) Tensão no motor 2.

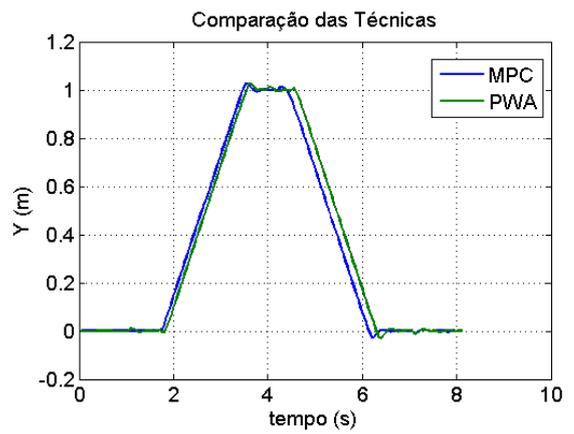


(d) Tensão no motor 3..

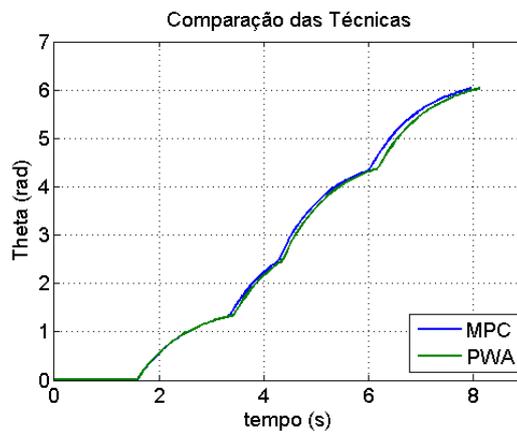
Figura 3.24: Simulação do seguimento da trajetória degrau, comparação do controle preditivo com compensação de atrito com o controle do modelo visto como um sistema PWA.



(a) Coordenada x .



(b) Coordenada y .



(c) Orientação θ .

Figura 3.25: Simulação do seguimento da trajetória degrau, comparação do controle preditivo com compensação de atrito com o controle do modelo visto como um sistema PWA, continuação.

Capítulo 4

Implementação do controlador no robô

Esse capítulo tratará da aplicação do controlador proposto no capítulo 3 ao robô móvel omnidirecional com 3 rodas. Será feita primeiramente uma descrição do ambiente Lazarus. Depois, os resultados obtidos para os testes de controle de velocidade, compensação de zona morta e seguimento de trajetória serão analisados mostrando o desempenho do controlador aplicado ao robô real. O algoritmo de otimização utilizado para minimizar a função custo está descrito em detalhes no apêndice A.

4.1 Lazarus/Comunicação

Nesse trabalho o ambiente Lazarus foi utilizado para implementar a lei de controle e comunicar com o sistema computacional do robô. O Lazarus é uma ferramenta de desenvolvimento livre e de código aberto para o compilador Free Pascal que também é livre e de código aberto. O Lazarus IDE (*Integrated Development Environment* - Ambiente de Desenvolvimento Integrado) é um ambiente de programação estável e rico em recursos para criar aplicações *standalone* gráficas e para console. É um compilador de pascal orientado a objeto compatível com os sistemas operacionais Linux, Mac OS X e Win32 e oferece um editor de código personalizado e ambiente de criação visual junto com um gerenciador de pacotes, depurador e integração completa GUI com o compilador Free Pascal [Baeseman et al., 1999b].

No ambiente Lazarus foi criado o *software Decision*, utilizado para controlar o robô. O *software* recebe do sistema computacional os valores de velocidades, tensões e correntes, e utilizando essas informações o estado do robô (malha externa) é calculado a partir da cinemática inversa e é utilizado pelo MPC, depois disso, na malha interna, os sinais de controle, que são as tensões nos 3 motores de corrente contínua, são passados para o robô. A cada período de amostragem de 50 ms, o algoritmo MPC calcula as tensões a serem aplicadas nos motores a partir dos valores da referência e do estado do robô, as tensões são então aplicadas ao robô e um novo vetor de estados é obtido e realimentado, fechando a malha interna. A cada iteração as coordenadas x , y e θ serão realimentadas, fechando a malha externa, e comparadas com a referência de posição e uma nova referência de velocidade será passada para o MPC. Os dados são passados do *Decision* para o robô através da porta serial. A figura 4.1 ilustra o *software Decision*.

4.2 Resultados Experimentais

Foram feitos experimentos para verificar o desempenho do controlador aplicado no robô real. Os experimentos serão analisados na mesma ordem das simulações, primeiramente os experimentos relativos ao controle de seguimento de referências constantes de velocidade, depois será analisada

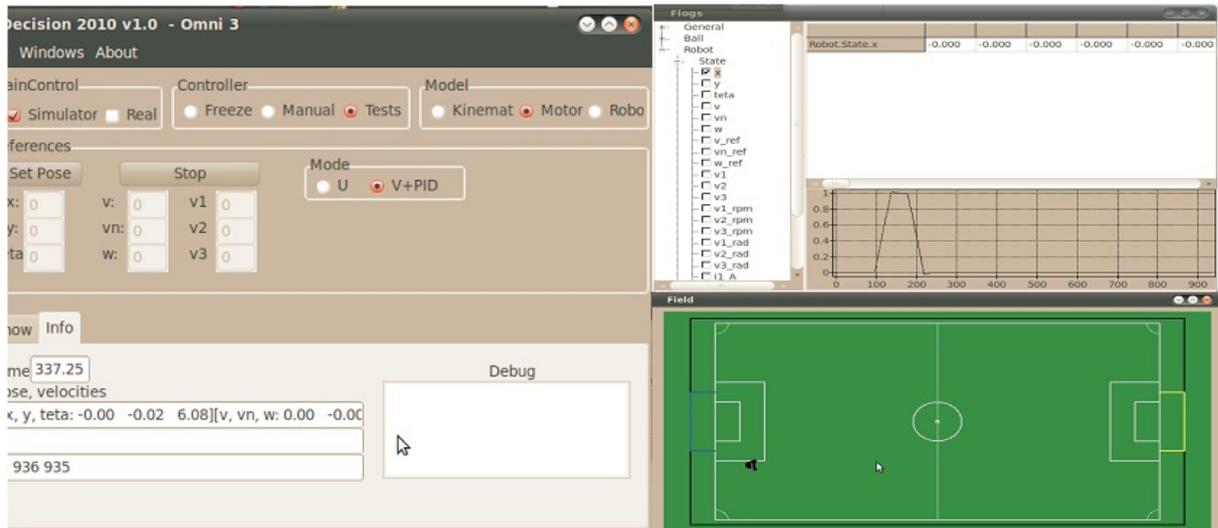


Figura 4.1: *Software Decision.*

a compensação de zona morta e por fim a análise do teste com o esquema em cascata fazendo com que o robô siga uma determinada trajetória no espaço.

Os experimentos que foram realizados sofrem a influência de muitas perturbações, que não estão presentes nas simulações. O atrito do solo não é constante e a matriz K_d , que leva em consideração os coeficientes de atrito, é considerada constante no modelo, o que pode prejudicar o desempenho do controlador. O desempenho dos testes também sofre a influência das condições ambientais, do ruído e do erro de quantização. Outro fator importante é que a distribuição de massa no robô não é exatamente uniforme ao contrário do que se considera no modelo.

Nos experimentos as informações do robô são fornecidas exclusivamente pela odometria. Portanto, o robô calcula o erro baseado nessa informação que possui um erro em relação ao valor real da posição do robô.

4.2.1 Controle de Velocidades

Para os testes de controle de velocidades, foram utilizados os mesmos parâmetros da simulação, mas com:

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Para a referência de $v = 1m/s$, a curva de velocidade, o gráfico das ações de controle, o gráfico do erro e o gráfico da função custo podem ser observados na figura 4.2. Pode-se observar que a referência foi seguida com um tempo de resposta de 0,6s, já que o valor de v ficou oscilando em torno de $1m/s$ e os erros ficaram oscilando em torno de zero depois que a velocidade atingiu o regime permanente. As tensões nos motores 2 e 3 convergem para uma faixa em torno de 3,76V e $-3,76V$, respectivamente, enquanto u_1 fica oscilando em torno de 0V, coerente com o movimento para frente com velocidade constante, observa-se no teste que as tensões de entrada dos motores nunca saturam. Da figura 4.2(d) observa-se que o custo se aproximou de zero.

Para a referência de $vn = 1m/s$, a curva de velocidade e o gráfico das ações de controle podem ser observados na figura 4.3. Pode-se observar que a referência foi seguida, já que o valor de vn ficou oscilando em torno de $1m/s$, o que atesta que o controlador buscou o objetivo, e o tempo de resposta foi de 0,55s. As tensões nos motores 2 e 3 convergem para uma faixa em

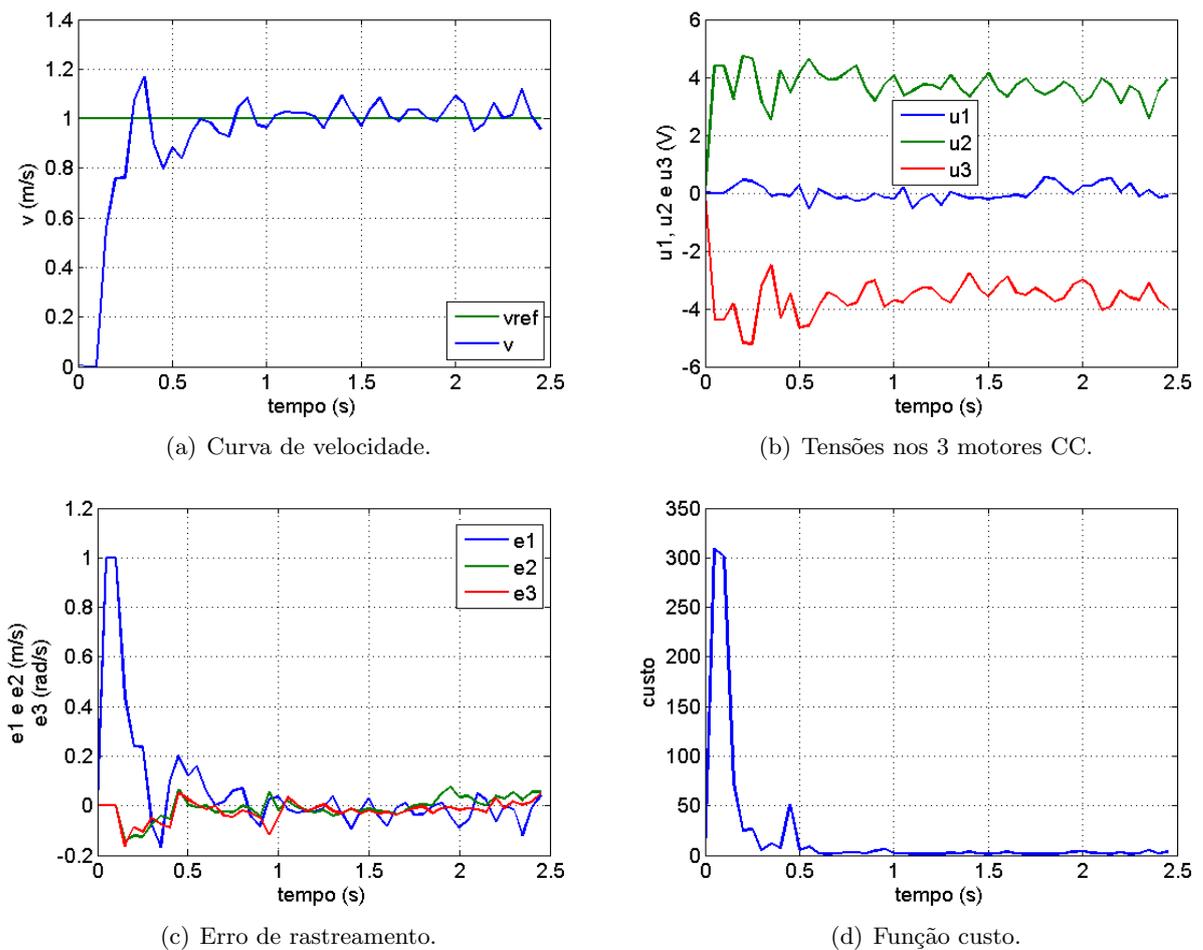


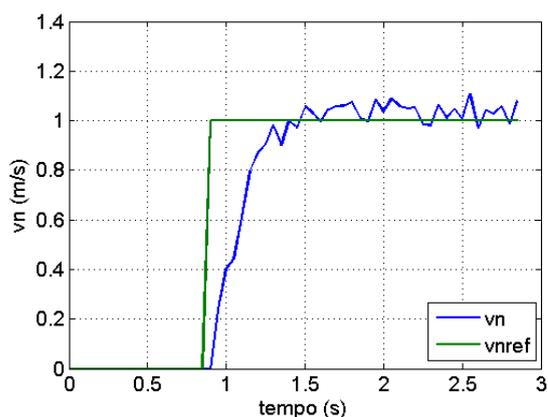
Figura 4.2: Experimento com a referência $v = 1m/s$.

torno de aproximadamente $2V$ enquanto u_1 fica oscilando em torno de $-4V$, coerente com o movimento na direção normal com velocidade constante.

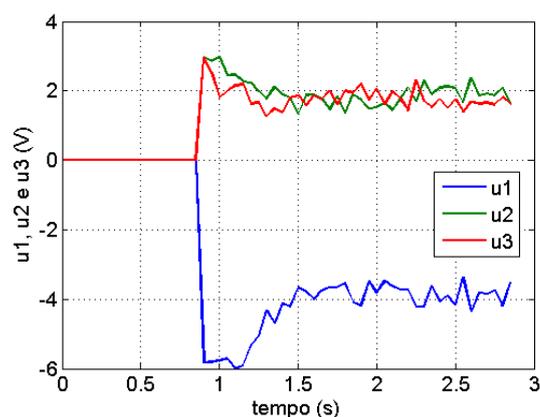
Para a referência de $\omega = 1rad/s$, a curva de velocidade e o gráfico das ações de controle podem ser observados na figura 4.4. Pode-se observar que a referência foi seguida, já que o valor de ω ficou oscilando em torno de $1rad/s$ com um tempo de resposta de $0,5s$, o que atesta que o controlador buscou o objetivo. As tensões nos motores oscilam em torno de aproximadamente $0,3V$, sendo que a diferença entre elas e as oscilações são frutos das incertezas devidas ao ambiente, que não foram consideradas em simulação.

Para a referência de $v = 0,5m/s$ e $\omega = 1rad/s$, a curva das velocidades, o gráfico das ações de controle, o gráfico do erro e o gráfico da função custo podem ser observados na figura 4.5. Pode-se observar que a referência foi seguida, já que os valores de v e ω oscilam em torno do objetivo e conseqüentemente o erro de rastreamento cai para um valor em torno de zero. Da figura 4.5(b) observa-se que no regime permanente, os valores de u_1 , u_2 e u_3 oscilam com uma baixa amplitude em torno dos valores de $0,4V$, $2,5V$ e $-1,8V$ respectivamente. Já a função custo caiu para um valor pequeno, o que significa que o controlador se portou bem, pelo fato de ter buscado o objetivo.

Para a referência de $v = 0,6m/s$ e $vn = -0,6m/s$, a curva de velocidade e o gráfico das ações de controle podem ser observados na figura 4.6. Do gráfico das velocidades v e vn , percebe-se que passado o regime transitório, os valores oscilaram em torno do objetivo com uma amplitude relativamente pequena, o que quer dizer que o controlador MPC buscou a referência. Nota-se um tempo de resposta de $0,7s$. As tensões u_1 , u_2 e u_3 convergiram para uma faixa aproximadamente

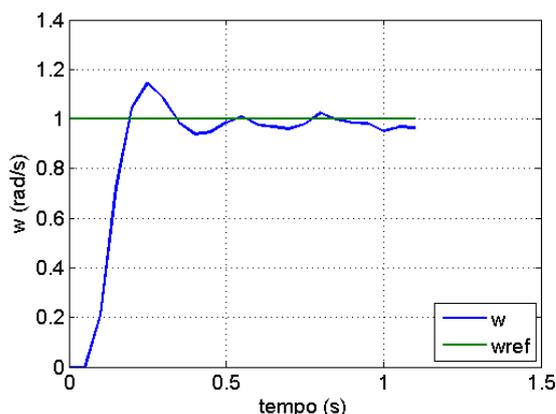


(a) Curva de velocidade.

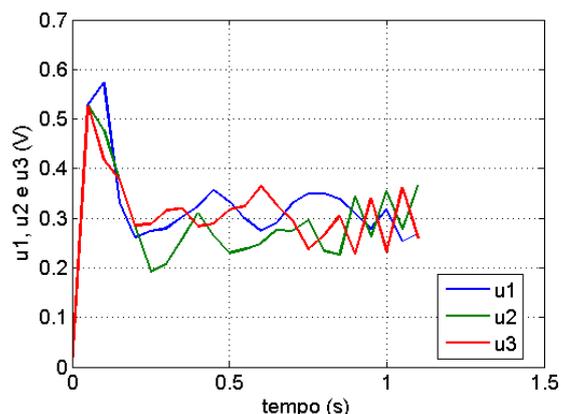


(b) Tensões nos 3 motores CC.

Figura 4.3: Experimento com a referência $vn = 1m/s$.



(a) Curva de velocidade.



(b) Tensões nos 3 motores CC.

Figura 4.4: Experimento com a referência $\omega = 1rad/s$.

em torno de $2,4V$, $0,95V$ e $-3,35V$, sendo que as ações de controle também oscilam com uma amplitude pequena, o que quer dizer que o desempenho foi bem próximo do ideal.

Concluiu-se que o desempenho do controle de velocidades do robô foi satisfatório, já que mesmo com as incertezas impostas pelo ambiente, os resultados dos experimentos conseguiram se aproximar bem dos resultados das simulações. Observou-se também para os experimentos que o algoritmo proposto é bastante eficiente, pois as referências foram seguidas sem erro de regime e com tempo de resposta de no máximo $0,7s$.

4.2.2 Compensação de zona morta

Os testes da Compensação de zona morta foram feitos mantendo os mesmos parâmetros do controle de velocidades. Para realizar os testes utilizou-se a mesma sequência e um procedimento análogo ao da simulação para que se possa analisar como o robô se comporta quando v , vn ou ω passam por zero.

Os gráficos das figuras 4.7(a) e 4.7(b), correspondem respectivamente, ao comportamento da velocidade v e das tensões correspondentes, que podem ser utilizados para comparar o desempenho do robô em malha aberta e em malha fechada. Pode-se observar que para o caso em malha fechada a zona morta é muito menor do que para a resposta em malha aberta, graças ao efeito

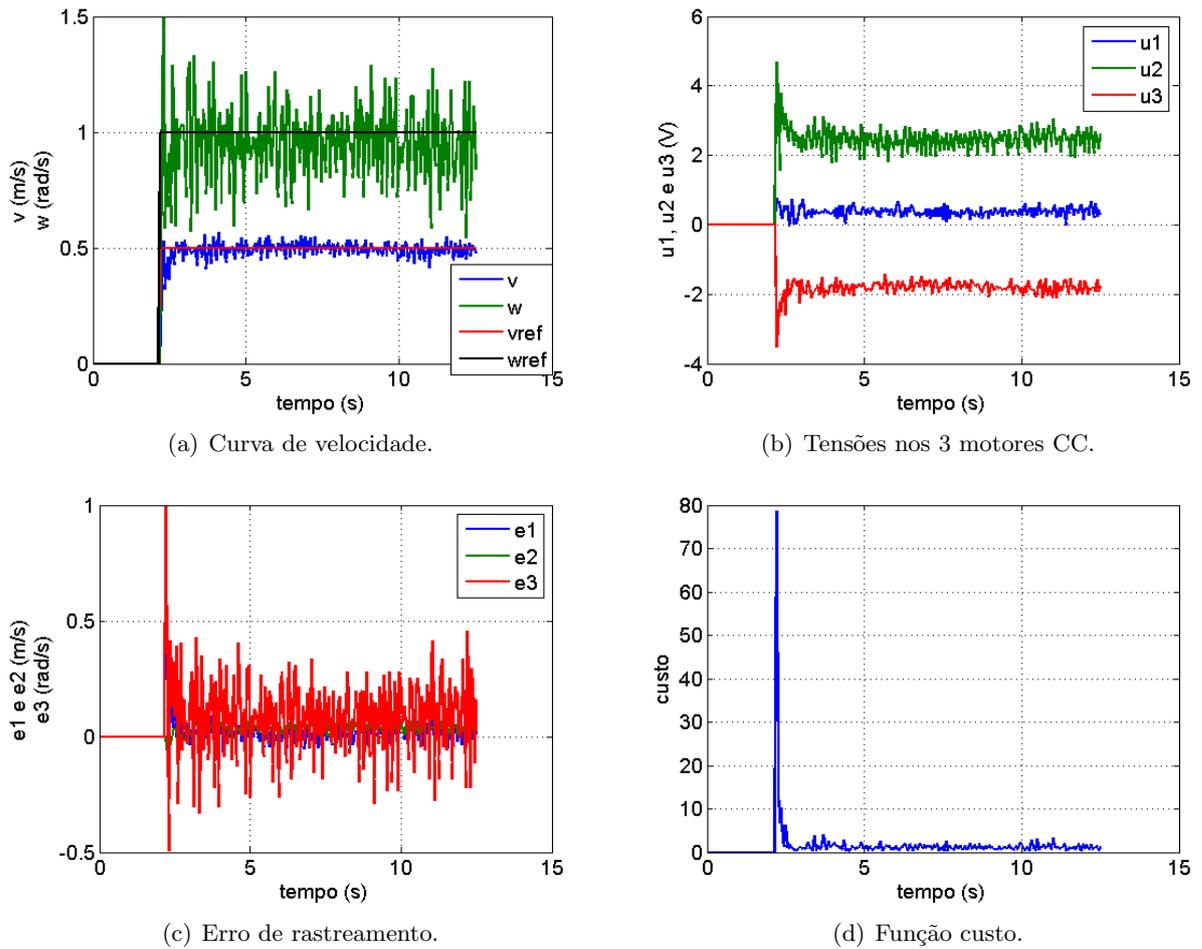


Figura 4.5: Experimento com a referência $v = 0,5m/s$ e $\omega = 1rad/s$.

da compensação do atrito estático (figura 4.7(a)). Da comparação com a reta, percebe-se que boa parte da zona morta foi compensada. Do gráfico das ações de controle (4.7(b)) percebe-se claramente a compensação, enquanto para o caso em malha aberta as tensões variam linearmente com o tempo, para o caso em malha fechada o comportamento das tensões varia bruscamente quando elas estão próximas de zero.

Na figura 4.8(a) pode-se perceber que a zona morta foi menor para o caso em malha fechada, e a compensação do atrito foi um pouco mais eficiente do que para o teste de v . Do gráfico das tensões nos motores (figura 4.8(b)) nota-se também uma variação brusca no comportamento das tensões para o caso com compensação de zona morta, para a tensão u_1 a variação foi maior do que para u_2 e u_3 já que em módulo essa tensão é maior do que as outras duas.

Da figura 4.9(a) conclui-se que a zona morta é bem menor para o caso com a compensação de atrito. No gráfico das tensões (figura 4.9(b)) nota-se que a variação nas ações de controle também ocorreu. Pela comparação da curva do ω com a reta em vermelho, pode-se perceber que a compensação de atrito foi mais eficiente para esse caso.

Nas simulações do capítulo 3, a compensação do atrito estático foi mais eficiente já que não houve zona morta na resposta em malha fechada. Por outro lado, nos experimentos analisados acima, ainda restou uma pequena parte da zona morta, e isso se deve às variações do coeficiente de atrito do solo e à zona morta dos motores, que não são consideradas no modelo de simulação utilizado. Observou-se também que o experimento com melhor desempenho foi o do ω o que permite concluir que esse é o estado que sofre menos influência do atrito.

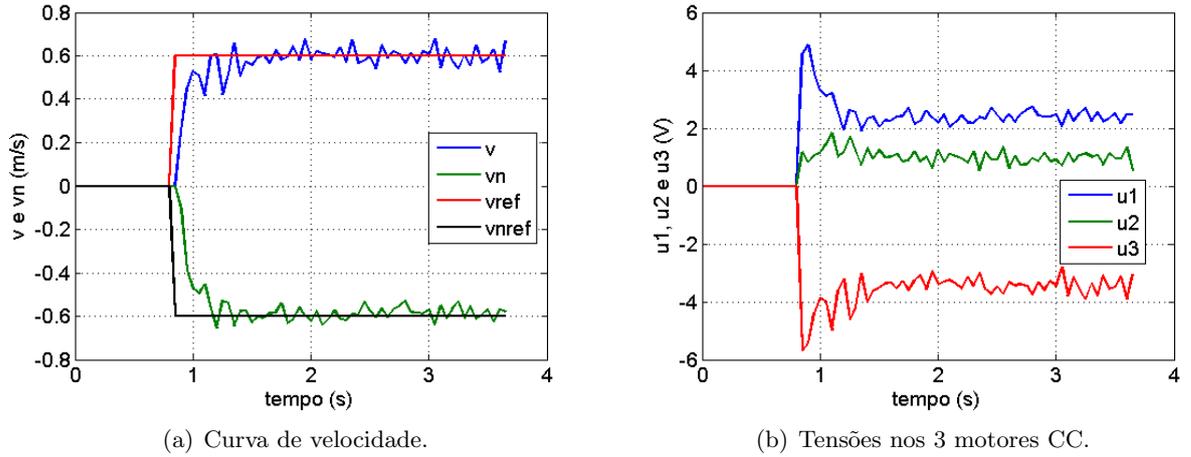


Figura 4.6: Experimento com a referência de $v = 0,6\text{m/s}$ e $vn = -0,6\text{m/s}$.

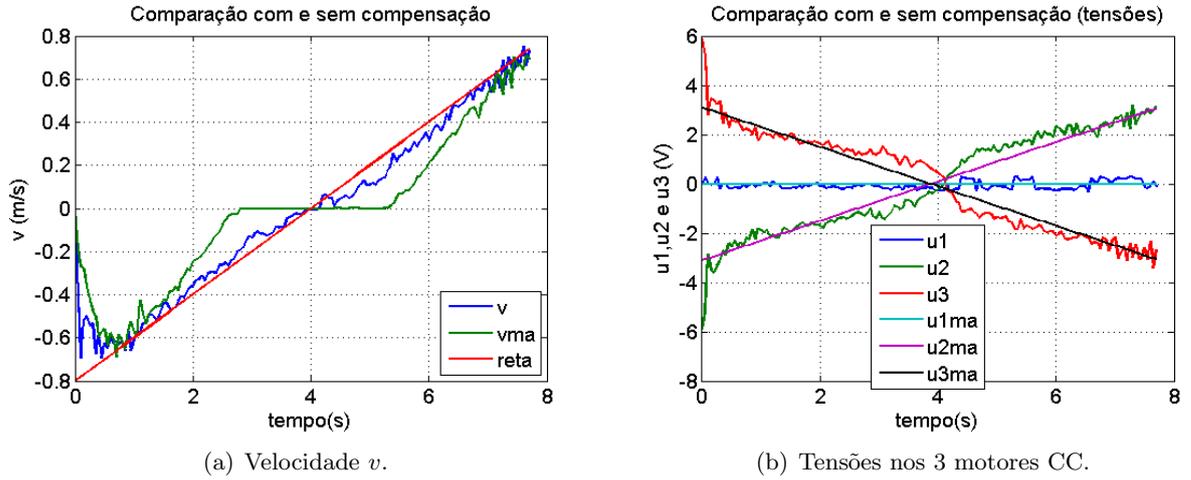


Figura 4.7: Compensação de zona morta para v .

4.2.3 Seguimento de trajetórias

Nesta etapa, foram utilizados os seguintes parâmetros de sintonia:

$$R = \begin{pmatrix} 10^{-2} & 0 & 0 \\ 0 & 10^{-2} & 0 \\ 0 & 0 & 10^{-2} \end{pmatrix},$$

$$Q = \begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}$$

$$\text{e } H_p = H_u = 3.$$

Para a trajetória em degrau, com $v_{nav} = 0,3\text{m/s}$, foram obtidos os gráficos das figuras 4.10 e 4.11. Observa-se na figura do rastreamento da trajetória (figura 4.10(a)) que o desempenho do seguimento foi satisfatório já que o erro de seguimento foi pequeno até nas transições. As figuras 4.10(b) e 4.10(c) mostram o comportamento das coordenadas e da orientação, respectivamente, e

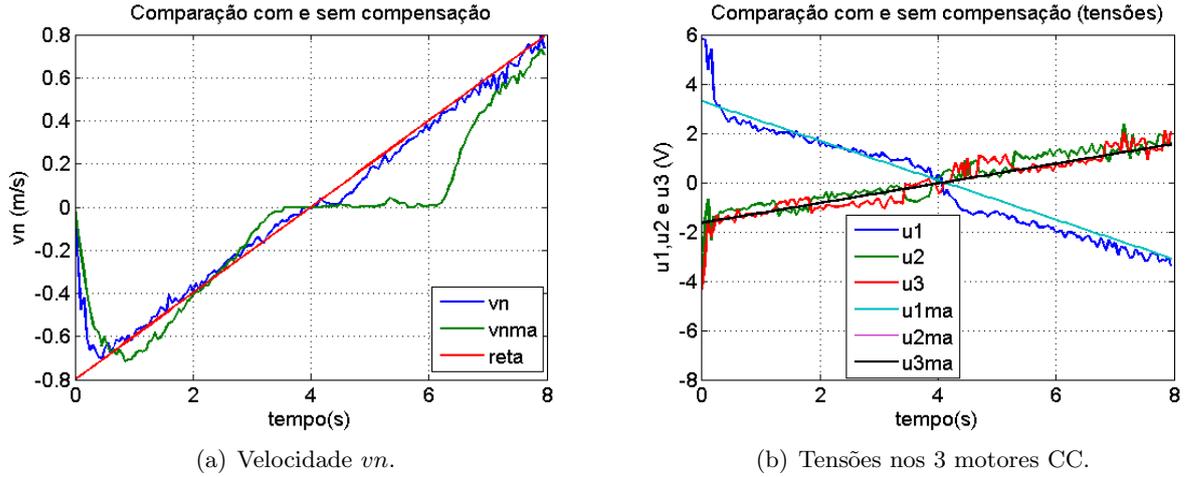


Figura 4.8: Compensação de zona morta para v_n .

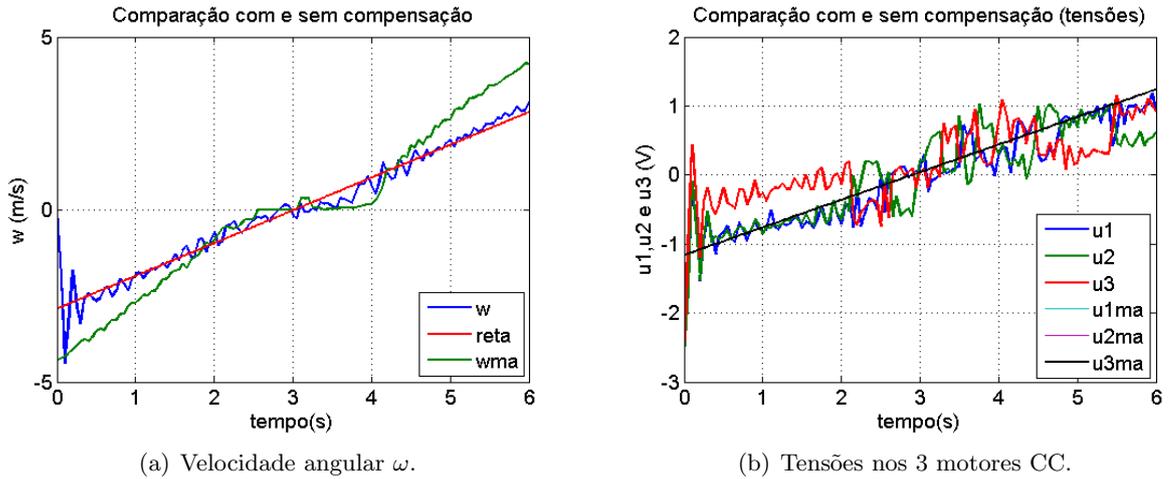


Figura 4.9: Compensação de zona morta para ω .

observa-se que a trajetória foi seguida com um tempo de navegação de 12,3s. Da figura 4.10(d), nota-se que os valores de tensão dos motores alcançaram no máximo valores na faixa de 3,5V em módulo, portanto não ocorreu saturação dos motores, percebe-se também que o controle foi mais agressivo nas transições, denotando que a predição foi eficiente. Dos gráficos do rastreamento de velocidades conclui-se que a referência de velocidade foi seguida, e que o erro de rastreamento é maior quando ocorrem as mudanças de direção e orientação, percebe-se também, em especial para ω , que existem oscilações, que se devem às incertezas do aparato experimental, que foram descritas anteriormente (figuras 4.10(e), 4.10(f) e 4.11).

Também foi feito um teste para a trajetória diagonal na qual $y = x$ e $\theta = 0rad$. Com $v_{nav} = 0,6m/s$, tomando como posição inicial $x = 0m$, $y = 0,5m$ e $\theta = 1,57rad$ ao invés de tomar a origem da trajetória, obtém-se as figuras 4.12 e 4.13. Do seguimento da trajetória e do desempenho separado de cada coordenada e da orientação, nas figuras 4.12(a), 4.12(b) e 4.12(c), respectivamente, observa-se que a trajetória foi rastreada, de modo que, partindo da sua posição inicial o robô alcançou a trajetória na posição $x = y = 1m$ em dois segundos. Do gráfico da figura 4.12(d) percebe-se que as tensões dos motores alcançaram valores da ordem de 4,5V, o que quer dizer que os atuadores não foram tão sacrificados, pois os valores das tensões nem chegaram perto de 6V. Do rastreamento das referências de velocidade (figuras 4.13(a), 4.13(b), 4.13(c)),

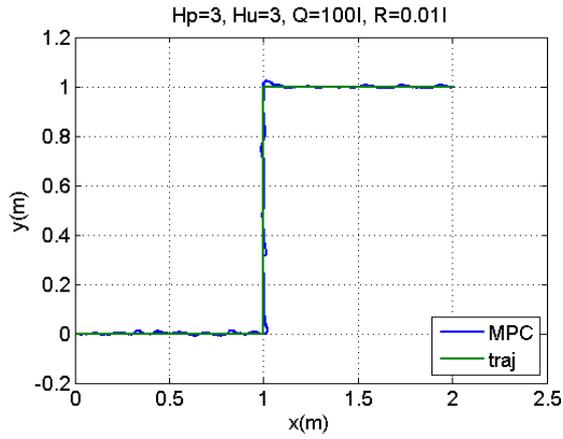
conclui-se que as velocidades foram seguidas com um tempo de resposta da ordem de $0,5s$.

Para a trajetória em oito, com $v_{nav} = 0,5m/s$, tomando como posição inicial $x = 1m$, $y = 0m$ e $\theta = 0rad$ que não pertence à trajetória, obtém-se as figuras 4.14 e 4.15. Do seguimento da trajetória e do desempenho separado de cada coordenada e da orientação, nas figuras 4.14(a), 4.14(b) e 4.15(a), respectivamente, observa-se que, partindo da sua posição inicial o robô alcançou a trajetória rapidamente. Do gráfico da figura 4.15(b) percebe-se que as tensões dos motores não saturaram durante o experimento, o que quer dizer que o controlador não teve dificuldades para rastrear a trajetória. Do rastreamento das referências de velocidade (figuras 4.15(c), 4.15(d) e 4.15(e)), conclui-se que as velocidades foram seguidas com um tempo de resposta da ordem de $0,45s$.

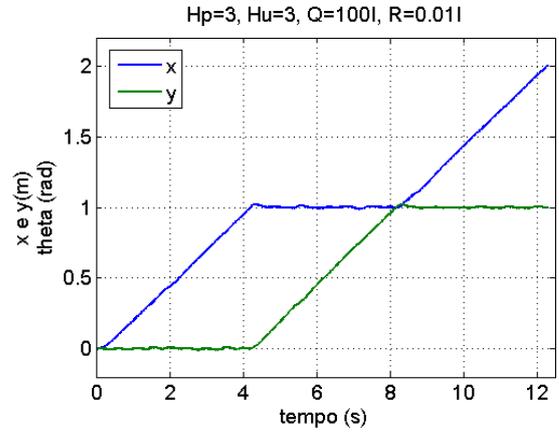
4.3 Conclusões

Dos testes realizados nesse capítulo, verifica-se que os resultados apresentados seguiram a mesma tendência que os da simulação, porém com o desempenho limitado pelos fatores não incluídos no modelo do robô. Foram observadas oscilações nos estados, e para os testes da zona morta, observou-se que a zona morta quase se extinguiu. A justificativa está em todas as incertezas do ambiente dos testes que não foram incluídas no modelo de simulação, especialmente as variações no coeficiente de atrito do solo.

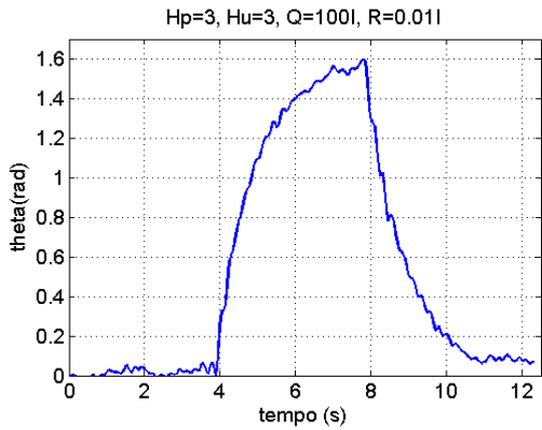
Os resultados obtidos foram satisfatórios já que nos experimentos de controle de velocidades usando somente a malha interna as referências foram seguidas com uma boa aproximação, obteve-se sucesso na compensação de zona morta e as trajetórias foram rastreadas sem erro aparente. As referências de velocidade passadas para o MPC pelo bloco da cinemática inversa foram seguidas e as tensões dos motores ficaram sempre abaixo de $6V$.



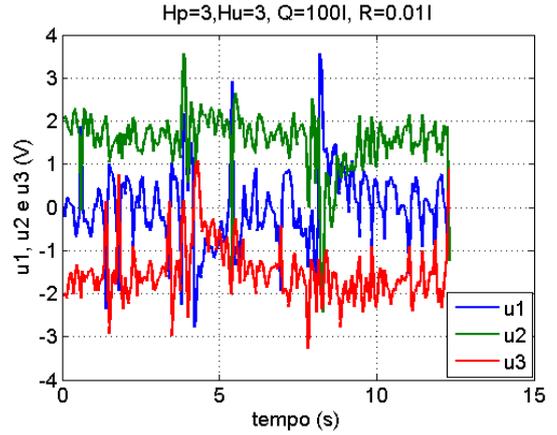
(a) Seguimento da trajetória.



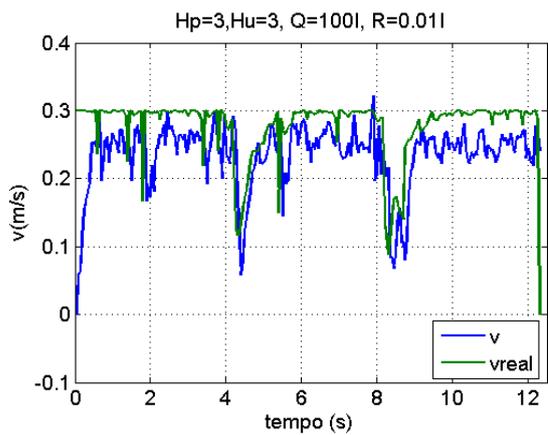
(b) Coordenadas x, y .



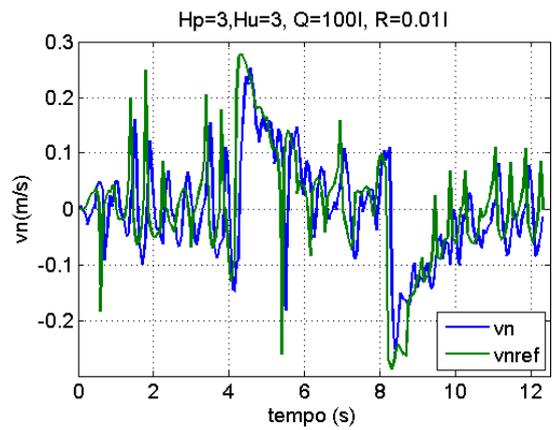
(c) Orientação θ .



(d) Tensões nos motores.



(e) Componente v da velocidade.



(f) Componente vn da velocidade.

Figura 4.10: Seguimento da trajetória em degrau, $v_{nav} = 0,3m/s$.

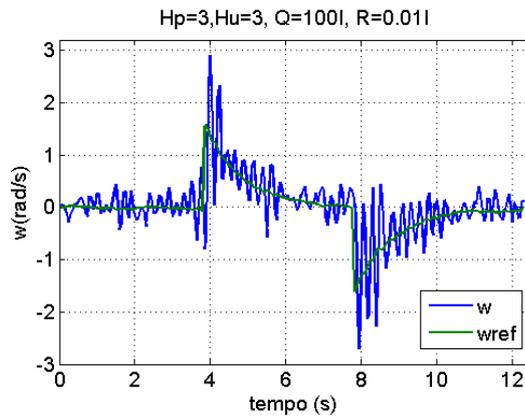
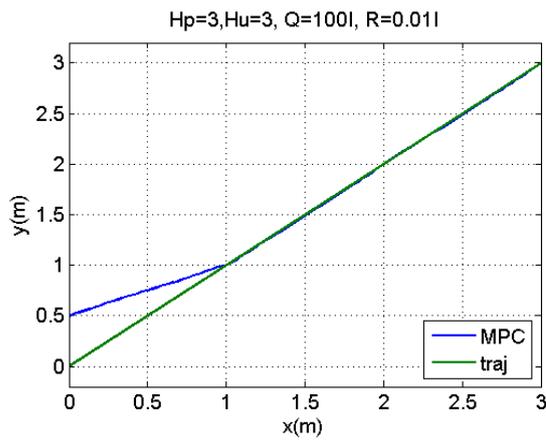
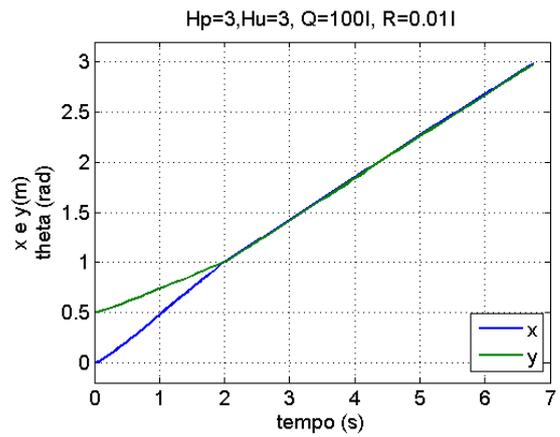


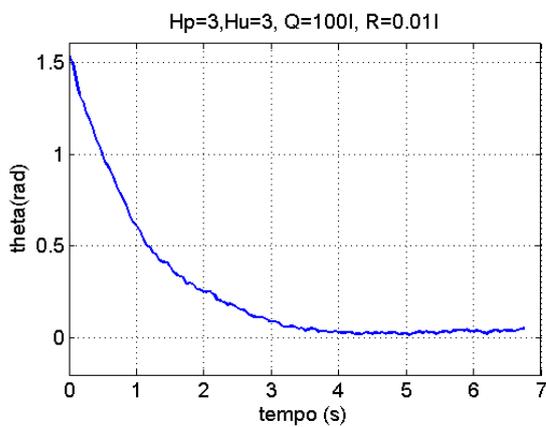
Figura 4.11: Componente ω da velocidade. Continuação da figura 4.10.



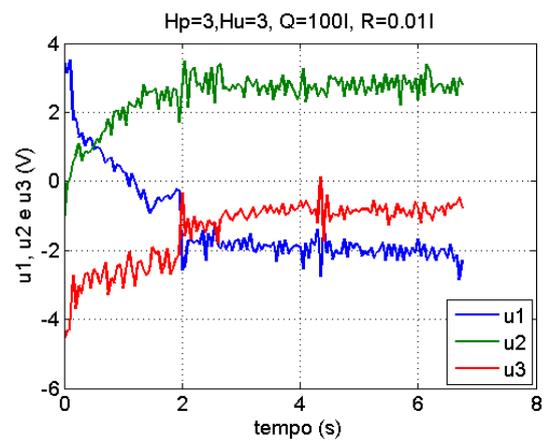
(a) Seguimento da trajetória.



(b) Coordenadas x e y e θ .

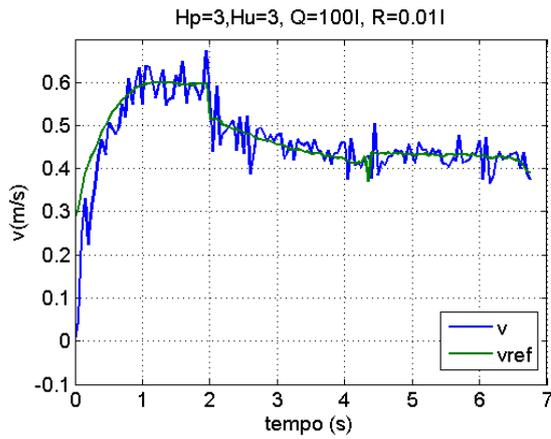


(c) Orientação θ .

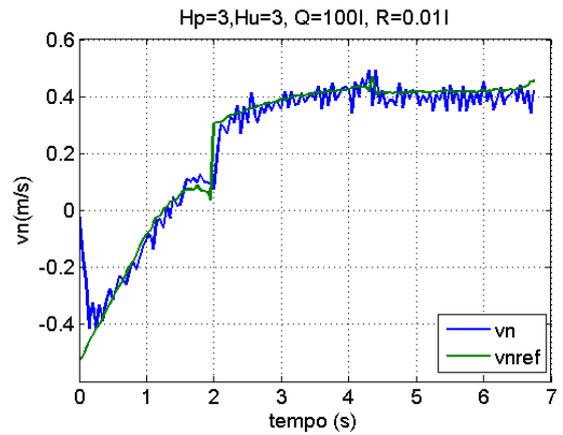


(d) Tensões nos motores.

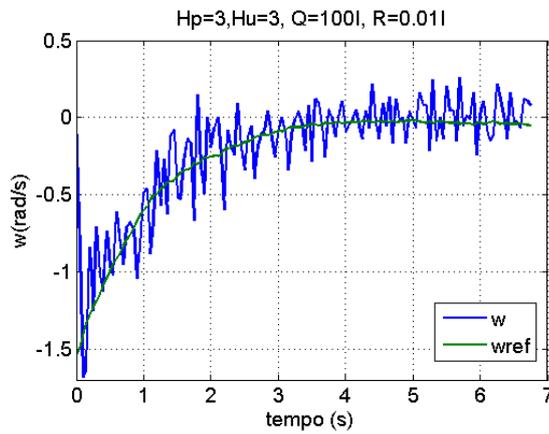
Figura 4.12: Seguimento da trajetória diagonal, $v_{nav} = 0,6m/s$.



(a) Componente v da velocidade.

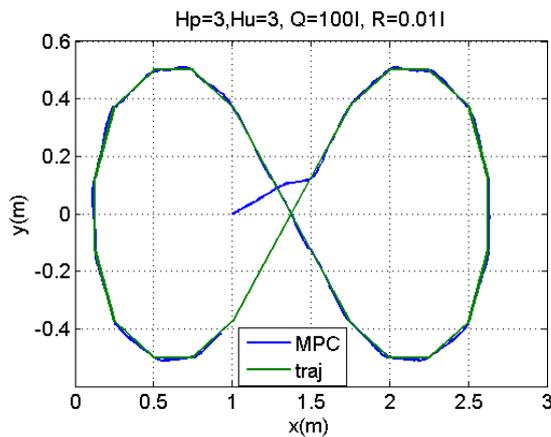


(b) Componente vn da velocidade.

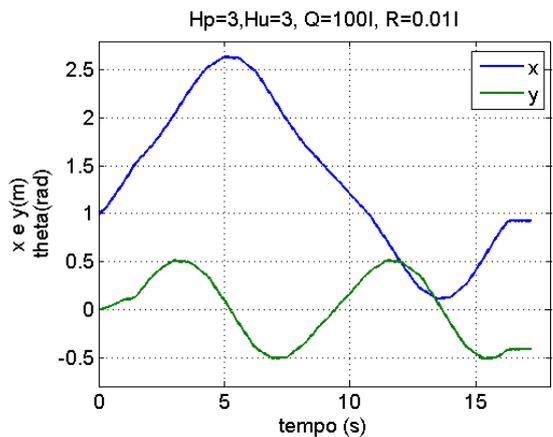


(c) Componente ω da velocidade.

Figura 4.13: Componentes de velocidade. Continuação da figura 4.12.

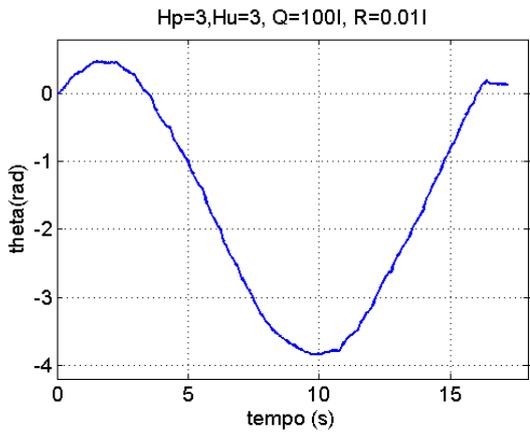


(a) Seguimento da trajetória.

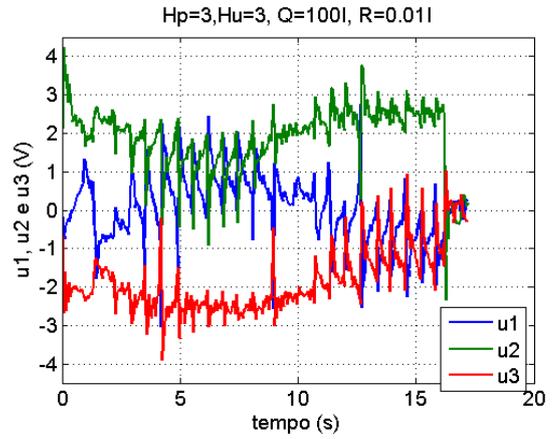


(b) Coordenadas x e y .

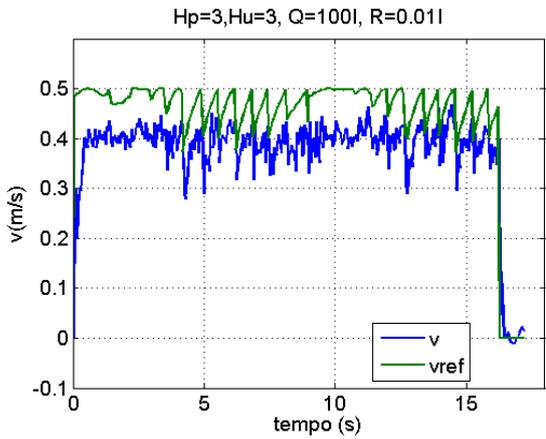
Figura 4.14: Seguimento da trajetória em oito, $v_{nav} = 0,5m/s$.



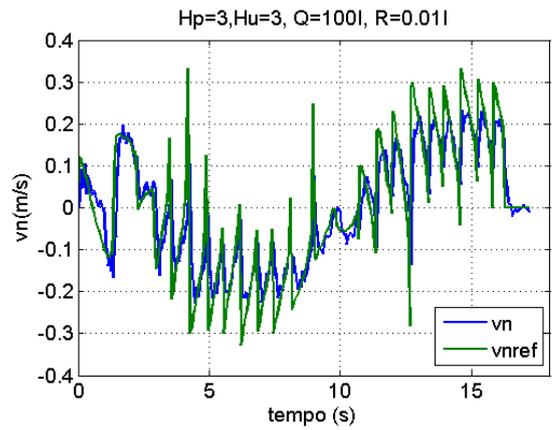
(a) Orientação θ .



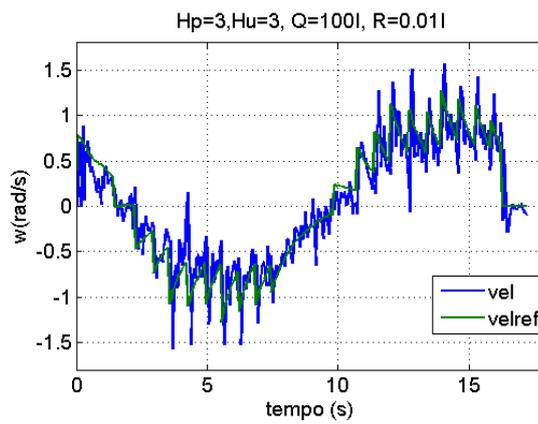
(b) Tensões nos motores.



(c) Componente v da velocidade.



(d) Componente vn da velocidade.



(e) Componente ω da velocidade.

Figura 4.15: Seguimento da trajetória em oito, $v_{nav} = 0,5m/s$, continuação.

Capítulo 5

Conclusão Geral

Nesse trabalho, foi proposto um algoritmo MPC com compensação de atrito com o objetivo de resolver o problema de seguimento de trajetórias para um robô móvel omnidirecional de 3 rodas. No sistema de controle em cascata, o bloco da cinemática inversa é utilizado para obter as referências de velocidade que são passadas para o MPC a cada iteração, a partir da posição atual do robô. Portanto o algoritmo desenvolvido realiza o controle de velocidades, realizado na malha interna da cascata. Foi utilizado um modelo simplificado para o robô, e as matrizes do espaço de estados foram utilizadas para projetar a lei de controle preditivo. Aplicou-se também, ao mesmo modelo, uma lei de controle projetada com o auxílio da ferramenta MPT do *software Matlab/Simulink*, que utiliza o modelo na forma afim por partes.

Para permitir o uso do MPC linear, foi usado um modelo simplificado. O valor do momento de inércia é aproximado, não são consideradas deformações nas rodas nem irregularidades no terreno e a distribuição de massa foi considerada uniforme. Apesar disso a técnica obteve desempenho satisfatório para o controle de trajetória. Os parâmetros também foram validados, comparando cada saída do modelo com as do robô para um dado conjunto de entradas. Observou-se que o modelo reproduz bem o robô já que suas saídas tiveram valores em torno das do robô.

O algoritmo MPC foi desenvolvido e foram realizadas simulações para analisar o desempenho da técnica utilizada. Primeiramente foram feitos testes do controle de velocidades em que as referências foram seguidas com um bom desempenho, já que os tempos de resposta foram pequenos e não houve erro de regime, exceto para valores onde a entrada de equilíbrio viola as restrições. Para essa referência, o robô se moveria em um círculo com raio igual a $1m$, girando em torno do próprio eixo com $\omega = 1rad/s$. Observou-se também que o controlador foi capaz de compensar a zona morta. Para as simulações do seguimento das trajetórias de referência foi feito um estudo da influência dos parâmetros de sintonia. Também nessa etapa foi obtido um desempenho satisfatório. Para os testes com a lei de controle projetada a partir da ferramenta MPT, foram obtidos resultados semelhantes ao algoritmo MPC para $H_p = H_u = 1$. Já para esse valor dos horizontes, a lei de controle obtida tem 4928 regiões, e para valores maiores de H_p e H_u esse número aumenta muito, o que torna impraticável a realização das simulações. Por esse motivo, optou-se por implementar somente o algoritmo MPC na plataforma experimental.

Foi necessário implementar um algoritmo de otimização para resolver o problema de controle preditivo. Utilizou-se para isso o algoritmo dos Pontos Interiores. A implementação se mostrou viável, pois todos os cálculos puderam ser realizados dentro do período de amostragem. Dos resultados do controle de velocidades, observou-se que elas atingem a referência. Dos testes da compensação de atrito, nota-se que o esquema de controle proposto não extinguiu, mas reduziu bastante o efeito de zona morta. As trajetórias foram seguidas de forma satisfatória. As razões pela qual se escolheu o controle preditivo foram justificadas na prática, já que as restrições foram sempre respeitadas, a técnica não apresenta dificuldades em controlar um sistema multivariável e foi possível realizar a compensação de atrito.

Como trabalho futuro, o modelo poderia ser melhorado para ficar mais próximo à realidade, incorporando ao modelo perturbações decorrentes das irregularidades do terreno, dos erros de

medição e da distribuição de massa, por exemplo. Poderia-se também, avaliar a estratégia de controle ao incorporar realimentação visual, ao invés de considerar apenas informações da odometria. O controle também poderia ser melhorado através do uso do controle preditivo robusto que considera erros de modelagem, utilizando diretamente o controle preditivo não-linear, ou até o linear por partes *on-line*.

Referências Bibliográficas

- [Anderson and Stone, 2007] Anderson, P. and Stone, H. (2007). Predictive guidance and control for a tail-sitting unmanned aerial vehicle. In *Information, Decision and Control, 2007. IDC '07*, pages 148–153.
- [Baeseman et al., 1999a] Baeseman, C., Miller, S., and Hess, M. A. (1999a). Lazarus project.
- [Baeseman et al., 1999b] Baeseman, C., Miller, S., and Hess, M. A. (1999b). Lazarus wiki.
- [Camacho and Bordons, 2004] Camacho, E. and Bordons, C. (2004). *Model Predictive Control*. Springer-Verlag Ltd.
- [Canudas de Wit et al., 1995] Canudas de Wit, C., Olsson, H., Astrom, K., and Lischinsky, P. (1995). A new model for control of systems with friction. *Automatic Control, IEEE Transactions on*, 40(3):419–425.
- [Conceição et al., 2009] Conceição, A., Moreira, A., and Costa, P. (2009). Practical approach of modeling and parameters estimation for omnidirectional mobile robots. *Mechatronics, IEEE/ASME Transactions on*, 14(3):377–381.
- [Conceição et al., 2007] Conceição, A., Oliveira, H., Sousa e Silva, A., Oliveira, D., and Moreira, A. (2007). A nonlinear model predictive control of an omni-directional mobile robot. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, pages 2161–2166.
- [Cutler and Ramaker, 1979] Cutler, C. R. and Ramaker, B. L. (1979). Dynamic matrix control - a computer control algorithm. *AIChE 86th National Meeting*.
- [Gang et al., 2005] Gang, D. S., Ming, F. C., and Dong, L. H. (2005). Orientation control of a differential mobile robot through wheel synchronization. *Mechatronics, IEEE/ASME Transactions on*, 10(3):345 – 351.
- [Grieder et al., 2004] Grieder, P., Kvasnica, M., Baotic, M., and Morari, M. (2004). Low complexity control of piecewise affine systems with stability guarantee. *Proceeding of the 2004 American Control Conference*, 2(1):1196–1201.
- [Hwang and Chang, 2007] Hwang, C.-L. and Chang, L.-J. (2007). Trajectory tracking and obstacle avoidance of car-like mobile robots in an intelligent space using mixed h2/hinf decentralized control. *Mechatronics, IEEE/ASME Transactions on*, 12(3):345 – 352.
- [Ji et al., 2003] Ji, M., Zhang, Z., Biswas, G., and Sarkar, N. (2003). Hybrid fault adaptive control of a wheeled mobile robot. *Mechatronics, IEEE/ASME Transactions on*, 8(2):226 – 233.
- [Kang and Hedrick, 2009] Kang, Y. and Hedrick, J. (2009). Linear tracking for a fixed-wing uav using nonlinear model predictive control. *Control Systems Technology, IEEE Transactions on*, 17(5):1202–1210.

- [Kanjawanishkul and Zell, 2008] Kanjanawanishkul, K. and Zell, A. (2008). Distributed model predictive control for coordinated path following control of omnidirectional mobile robots. In *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*, pages 3120–3125.
- [Kanjawanishkul and Zell, 2009] Kanjanawanishkul, K. and Zell, A. (2009). Path following for an omnidirectional mobile robot based on model predictive control. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, page 3341.
- [Kvasnica et al., 2004] Kvasnica, M., Grieder, P., and Baotić, M. (2004). Multi-Parametric Toolbox (MPT).
- [Li and Zell, 2007] Li, X. and Zell, A. (2007). Motion control of an omnidirectional mobile robot. In *ICINCO-RA (1)*, pages 125–132.
- [Lim et al., 2008] Lim, H., Kang, Y., Kim, C., Kim, J., and You, B.-J. (2008). Nonlinear model predictive controller design with obstacle avoidance for a mobile robot. In *Mechatronic and Embedded Systems and Applications, 2008. MESA 2008. IEEE/ASME International Conference on*, pages 494–499.
- [Liu et al., 2008] Liu, Y., Zhu, J., Williams II, R., and Wu, J. (2008). Omni-directional mobile robot controller based on trajectory linearization. *Robotics and Autonomous Systems*, 56(5):461 – 479.
- [Low and Wang, 2008] Low, C. B. and Wang, D. (2008). Gps-based tracking control for a car-like wheeled mobile robot with skidding and slipping. *Mechatronics, IEEE/ASME Transactions on*, 13(4):480 – 484.
- [Maciejowski, 2002] Maciejowski, J. (2002). *Predictive control with constraints*. Pearson Education Ltd.
- [Maxon, 2002] Maxon (2002). Folha de dados a-max 22 r179, precious metal brushes cll, 5 watt. 1a ed. maxon motors.
- [Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2nd edition.
- [Raffo et al., 2009] Raffo, G., Gomes, G., Normey-Rico, J., Kelber, C., and Becker, L. (2009). A predictive controller for autonomous vehicle path tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 10(1):92–102.
- [Ray, 2009] Ray, L. (2009). Estimation of terrain forces and parameters for rigid-wheeled vehicles. *Robotics, IEEE Transactions on*, 25(3):717–726.
- [Richalet et al., 1978] Richalet, J., Rault, A., Testud, J., and Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *automatica*, 14(5):413–428.
- [Shim et al., 2006] Shim, D., Chung, H., and Sastry, S. (2006). Conflict-free navigation in unknown urban environments. *Robotics & Automation Magazine, IEEE*, 13(3):27–33.
- [Zhao et al., 2009] Zhao, D., Deng, X., and Yi, J. (2009). Motion and internal force control for omnidirectional wheeled mobile robots. *Mechatronics, IEEE/ASME Transactions on*, 14(3):382 – 387.

Apêndice A

Método dos Pontos Interiores

A maior dificuldade na implementação do controle preditivo é o método de otimização. Trata-se de um problema quadrático cuja função custo é dada pela equação 3.14. A obtenção das matrizes que serão utilizadas pelo algoritmo de otimização é feita a partir das matrizes do espaço de estados e da etapa da predição, que consiste na grande vantagem da ferramenta de controle preditivo. Existem vários tipos de algoritmo de programação quadrática, o escolhido foi o Algoritmo dos Pontos Interiores, que é um método de barreira.

O problema de otimização em questão pode ser escrito como:

$$\min_{x,s} f(x) \tag{A.1}$$

$$\begin{aligned} \text{s.a. } c_E(x) &= 0 \\ c_I(x) - s &= 0 \\ s &\geq 0, \end{aligned}$$

onde as restrições de desigualdade $c_I \geq 0$ foram transformadas em restrições de igualdade introduzindo um vetor s de variáveis de folga [Nocedal and Wright, 2006].

As condições de Kuhn-Tucker para a programação não-linear podem ser dadas por:

$$\nabla f - A_E^T y - A_I^T z = 0, \tag{A.2}$$

$$Sz - \mu e = 0, \tag{A.3}$$

$$c_E(x) = 0, \tag{A.4}$$

$$c_I(x) - s = 0, \tag{A.5}$$

onde μ é o parâmetro de barreira, $A_E(x)$ e $A_I(x)$ são as Jacobianas das funções c_E e c_I , respectivamente, e y e z são os multiplicadores de Lagrange. Na implementação, a cada iteração o parâmetro μ é multiplicado por uma constante σ . S e Z são matrizes diagonais, cujos elementos são dados pelos vetores s e z , respectivamente, e faz-se $e = (1, 1, \dots, 1)^T$ [Nocedal and Wright, 2006].

Aplicando o método de Newton ao sistema não-linear definido acima, nas variáveis x , s , y e z , pode-se obter [Nocedal and Wright, 2006]:

$$\begin{pmatrix} \nabla_{xx}^2 & 0 & -A_E^T(x) & -A_I^T(x) \\ 0 & Z & 0 & S \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} px \\ ps \\ py \\ pz \end{pmatrix} = - \begin{pmatrix} \nabla f - A_E^T(x) - A_I^T(x) \\ Sz - \mu e \\ c_E(x) \\ c_I(x) - s \end{pmatrix}. \quad (\text{A.6})$$

Para o modelo do robô ∇_{xx}^2 tem o mesmo valor da Hessiana, chamada de H , $A_E(x)$ corresponde à matriz da restrições de igualdade, e a matriz $A_I(x)$ é a matriz das restrições de desigualdade definida no capítulo 3 multiplicada por -1, já que as restrições definidas anteriormente tiveram que ser invertidas, já que elas foram inicialmente definidas como $c_I \leq 0$.

Resolve-se então o sistema e os valores de x , z , y e s são atualizados a cada iteração, utilizando o valor do vetor p , até a obtenção da solução ótima. Para o problema em questão a variável que se deseja estimar, o x do problema, é a ação de controle u aumentado definido como $u_{aum} = [u_2(k) \ \dots \ u_2(k + H_u - 1) \ u_1(k)]^T$. Segue atualização:

$$\begin{aligned} u_{aum}^+ &= u_{aum} + \alpha_s^{max} p u_{aum}, \\ s^+ &= s + \alpha_s^{max} p s, \\ y^+ &= y + \alpha_y^{max} p y, \\ z^+ &= z + \alpha_z^{max} p z, \text{ onde} \\ \alpha_y^{max} &= \max \alpha \in (0, 1] : y + \alpha p y \geq (1 - \tau) y, \\ \alpha_s^{max} &= \max \alpha \in (0, 1] : s + \alpha p s \geq (1 - \tau) s, \\ \alpha_z^{max} &= \max \alpha \in (0, 1] : z + \alpha p z \geq (1 - \tau) z. \end{aligned}$$

O parâmetro α é o comprimento do passo (em cada direção), que é determinado de maneira a não violar as restrições de não negatividade das variáveis z e s (o que é feito através das desigualdades utilizando-se o τ).

Ao contrário do *software Matlab*, o Lazarus não possui uma função pronta que realize a programação quadrática, portanto elaborou-se uma função no *Matlab* que implementa o algoritmo dos pontos interiores para realizar a otimização. Essa função foi validada utilizando o diagrama de blocos da figura 3.2. Os resultados dos testes tiveram um comportamento idêntico ao controlador MPC do capítulo 3. A função criada em *Matlab* foi então implementada em linguagem Pascal no *software Decision* criado no ambiente Lazarus.

Para o problema em questão, o problema de otimização pode ser escrito como segue:

$$\begin{aligned} \min u_{aum}^T H u_{aum} - g_-^T u_{aum} \\ \text{s.a. } D u_{aum} \leq d \\ A_{eq} u_{aum} = b_{eq}. \end{aligned} \quad (\text{A.7})$$

Expandindo as equações das restrições e substituindo os valores numéricos, tem-se que:

$$\begin{bmatrix} I & 0 & \dots & I \\ -I & 0 & \dots & -I \end{bmatrix} \begin{bmatrix} u_2(k) \\ \dots \\ u_2(k + H_u - 1) \\ u_1(k) \end{bmatrix} \leq \begin{bmatrix} 6 \\ 6 \end{bmatrix}, \quad (\text{A.8})$$

$$\begin{bmatrix} 0 & 0 & \dots & B \end{bmatrix} \begin{bmatrix} u_2(k) \\ \dots \\ u_2(k + H_u - 1) \\ u_1(k) \end{bmatrix} = -K \text{signal}(x(k)). \quad (\text{A.9})$$

Para as restrições de igualdade e desigualdade, respectivamente. I representa a matriz identidade de dimensão 3×3 , cada 6 na verdade é um vetor de dimensão 3 e cada zero é uma matriz 3×3 . Lembrando-se que cada termo do vetor u_{aum} e o estado x tem 3 elementos. Por simplicidade, considera-se, a partir de agora, que u_i , com $i = 1, \dots, 3H_u + 3$ corresponde a cada elemento do vetor, já que u_{aum} é composto de $H_u + 1$ vetores de 3 elementos. Analogamente H tem dimensões $(3H_u + 3) \times 3H_u + 3$ e g_- tem dimensões $(3H_u + 3) \times 1$.

O gradiente da função custo da equação A.7 é dado por:

$$\nabla f = \begin{bmatrix} 2h_{1,1}u_1 + 2h_{1,2}u_2 + \dots + 2h_{1,3H_u+3}u_{3H_u+3} - g_{-1} \\ 2h_{2,1}u_1 + 2h_{2,2}u_2 + \dots + 2h_{2,3H_u+3}u_{3H_u+3} - g_{-2} \\ \dots \\ 2h_{3H_u+3,1}u_1 + 2h_{3H_u+3,2}u_2 + \dots + 2h_{3H_u+3,3H_u+3}u_{3H_u+3} - g_{-3H_u+3} \end{bmatrix}. \quad (\text{A.10})$$

Efetuada a multiplicação matricial da equação A.8, multiplicando as 6 restrições de desigualdade por -1 para mudar o sentido do sinal de desigualdade e inserindo o vetor s de variáveis de folga para colocar as restrições de desigualdade no formato da equação A.5, tem-se:

$$\begin{aligned} -u_1 - u_{3H_u+1} + 6 \geq 0 &\Rightarrow -u_1 - u_{3H_u+1} + 6 - s_1 = 0, \\ -u_2 - u_{3H_u+2} + 6 \geq 0 &\Rightarrow -u_2 - u_{3H_u+2} + 6 - s_2 = 0, \\ -u_3 - u_{3H_u+3} + 6 \geq 0 &\Rightarrow -u_3 - u_{3H_u+3} + 6 - s_3 = 0, \\ u_1 + u_{3H_u+1} + 6 \geq 0 &\Rightarrow u_1 + u_{3H_u+1} + 6 - s_4 = 0, \\ u_2 + u_{3H_u+2} + 6 \geq 0 &\Rightarrow u_2 + u_{3H_u+2} + 6 - s_5 = 0, \\ u_3 + u_{3H_u+3} + 6 \geq 0 &\Rightarrow u_3 + u_{3H_u+3} + 6 - s_6 = 0, \end{aligned}$$

onde s_i , com $i = 1, \dots, 6$ representa os elementos do vetor s .

Colocando as 3 restrições de igualdade na forma da equação A.4, tem-se:

$$\begin{aligned} A_{eq(1,3H_u+1)}u_{3H_u+1} + A_{eq(1,3H_u+2)}u_{3H_u+2} + A_{eq(1,3H_u+3)}u_{3H_u+3} - b_{eq1} &= 0, \\ A_{eq(2,3H_u+1)}u_{3H_u+1} + A_{eq(2,3H_u+2)}u_{3H_u+2} + A_{eq(2,3H_u+3)}u_{3H_u+3} - b_{eq2} &= 0, \\ A_{eq(3,3H_u+1)}u_{3H_u+1} + A_{eq(3,3H_u+2)}u_{3H_u+2} + A_{eq(3,3H_u+3)}u_{3H_u+3} - b_{eq3} &= 0. \end{aligned}$$

Efetuada a multiplicação matricial da equação A.3, obtém-se:

$$s_1z_1 - \mu = 0, \quad s_2z_2 - \mu = 0, \quad s_3z_3 - \mu = 0, \quad s_4z_4 - \mu = 0, \quad s_5z_5 - \mu = 0 \quad \text{e} \quad s_6z_6 - \mu = 0.$$

Concatenando os vetores obtidos, o vetor independente da equação A.6 pode ser montado. Tem-se que $D = -A_I$, portanto a equação A.6 pode ser re-escrita como segue:

$$\begin{pmatrix} H & 0 & -A_E^T & D^T \\ 0 & Z & 0 & S \\ A_E & 0 & 0 & 0 \\ -D & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} pu_{aum} \\ ps \\ py \\ pz \end{pmatrix} = - \begin{pmatrix} \nabla f - A_E^T y + D^T z \\ Sz - \mu e \\ c_E u_{aum} \\ c_I u_{aum} - s \end{pmatrix}. \quad (\text{A.11})$$