UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA
E DE COMPUTAÇÃO

Identificação em Tempo Real de Modelo Dinâmico de Robô Móvel com Acionamento Diferencial e Zona Morta

Ellon Paiva Mendes

Orientador: Prof. Dr. Adelardo Adelino Dantas de Medeiros

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, Janeiro de 2012

Seção de Informação e Referência

Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Mendes, Ellon Paiva
Identificação em tempo real de modelo dinâmico de robô móvel com acionamento diferencial e zona morta / Ellon Paiva Mendes. – Natal, RN, 2012.
60 f. : il.

Orientador: Adelardo Adelino Dantas de Medeiros.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. Robôs móveis – Dissertação. 2. Identificação de sistemas – Dissertação. 3. Sistemas de Hammerstein – Dissertação. 4. Separação do termo chave – Dissertação. 5. Identificação de zona morta – Dissertação. 6. Mínimos quadrados recursivo – Dissertação. I. Medeiros, Adelardo Adelino Dantas de. II. Universidade Federal do Rio Grande do Norte. III. Título.

RN/UF/BCZM CDU 004.869

Identificação em Tempo Real de Modelo Dinâmico de Robô Móvel com Acionamento Diferencial e Zona Morta

Ellon Paiva Mendes

Dissertação de Mestrado aprovada em 27 de janeiro de 2012 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Adelardo Adelino Dantas de Medeiros (orientador) .... DCA/UFRN

Prof. Dr. Pablo Javier Alsina ..... DCA/UFRN

Prof. Dr. Andre Gustavo Scolari Conceição ..... DEE/UFBA

Prof. Dr. Carlos Eduardo Trabuco Dorea ..... DCA/UFRN

A minha mãe, *por todo seu amor,*
confiança e dedicação.

Agradecimentos

À Deus, por permitir receber mais esta benção em minha vida.

Aos professores Adelardo Medeiros e Pablo Alsina pela orientação, pela paciência e pela amizade nesses anos de vida acadêmica.

Aos meus companheiros de pesquisa do Laboratório de Robótica, Filipe Lins, João Paulo Guimarães, Márcio Valério Araújo, Gutemberg Santiago, Luiz Henrique Silva, Marcial Medeiros, Eduardo Rodrigues, Tania Laura, Diogo Pedrosa, Armando Sanca, Alessandro Schildt, Guilherme Santos, Vitor Meneghetti, e todos os demais que trabalharam em diversos momentos conosco, especialmente a Lennedy Soares pelos conselhos que me levaram a terminar este trabalho.

Aos meus amigos que sempre confiaram na minha capacidade, em especial ao pessoal do *Guardian*, por toda a amizade durante todos esses anos e pelos ótimos momentos que passamos juntos.

À Rafaella Monique, que apesar de tudo sempre esteve próxima para me apoiar quando foi preciso.

À minha querida mãe, Raimunda Paiva, por todo o amor, por tudo que ela fez e faz por mim, e por ser a excelente mãe que me colocou no caminho dos estudos e deu início a essa minha caminhada.

Ao CNPq, pelo apoio financeiro.

Resumo

Vários robôs móveis apresentam comportamentos não-lineares, principalmente ocasionados por fenômenos de atrito entre as partes mecânicas do robô ou entre o robô e o solo. Modelagens puramente lineares apresentam-se eficientes em alguns casos, mas é preciso levar em consideração as não-linearidades do robô quando se deseja movimentos ou posicionamentos precisos. Este trabalho propõe um procedimento de identificação paramétrica do modelo de um robô móvel com acionamento diferencial, no qual são consideradas as não-linearidades do tipo zona-morta presentes nos atuadores do robô. A proposta baseia-se no modelo de Hammerstein para dividir o sistema em blocos lineares e não-lineares. O princípio da separação do termo chave é utilizado para demonstrar a relação entre as entradas e saídas do sistema com os parâmetros tanto da parcela linear quanto da não-linear. Os parâmetros de ambas as parcelas são identificados simultaneamente, através de um algoritmo de mínimos quadrados recursivo. Os resultados mostram que é possível identificar o valor os limites da zona-morta assim como os parâmetros da parcela linear do modelo do sistema.

Palavras-chave: Robôs Móveis, Identificação de Sistemas, Sistemas de Hammerstein, Separação do Termo Chave, Zona Morta, Mínimos Quadrados Recursivo.

Abstract

Several mobile robots show non-linear behavior, mainly due friction phenomena between the mechanical parts of the robot or between the robot and the ground. Linear models are efficient in some cases, but it is necessary take the robot non-linearity in consideration when precise displacement and positioning are desired. In this work a parametric model identification procedure for a mobile robot with differential drive that considers the dead-zone in the robot actuators is proposed. The method consists in dividing the system into Hammerstein systems and then uses the key-term separation principle to present the input-output relations which shows the parameters from both linear and non-linear blocks. The parameters are then simultaneously estimated through a recursive least squares algorithm. The results shows that is possible to identify the dead-zone thresholds together with the linear parameters.

Keywords: Mobile Robots, System Identification, Hammerstein Systems, Key-term Separation Principle, Dead-zone, Recursive Least Squares.

Sumário

Sumário i

Lista de Figuras iii

Lista de Tabelas v

Lista de Símbolos e Abreviaturas vii

1 Introdução 1

1.1 Objetivo 3

1.2 Justificativa 3

1.3 Revisão da literatura 4

1.4 Organização do trabalho 5

2 Modelagem do sistema 7

2.1 Modelo cinemático 7

2.2 Modelo linear equivalente 9

2.3 Modelo dinâmico 10

2.4 Modelo com zona-morta 14

3 Estimação paramétrica 17

3.1 Parametrização do modelo linear 17

3.2 Parametrização do modelo com zona-morta 18

3.3 Simulador do sistema 19

3.4 Estimação do deslocamento linear 22

3.4.1 Cálculo do módulo de Δl 23

3.4.2 Cálculo do sinal de Δl 25

3.4.3 Validação do cálculo de $\hat{\Delta l}$ 26

3.5 Identificação dos parâmetros 27

3.5.1 Identificação do modelo linear 27

3.5.2 Identificação do modelo com zona-morta 29

3.6 Resultados simulados 30

4 Resultados experimentais **41**

4.1 Descrição do sistema 41

4.1.1 Sistema de visão e localização 42

4.1.2 Transmissão e tempo de atraso 45

4.1.3 Limites do espaço de trabalho 45

4.2 Estimação dos parâmetros 46

5 Conclusões e trabalhos futuros **57**

Referências bibliográficas **59**

Lista de Figuras

2.1 Estrutura do robô 8
2.2 Diagrama de forças 11
2.3 Modelo da zona-morta 14
2.4 Modelo de Hammerstein SISO 16
2.5 Robô descrito em modelos de Hammerstein 16

3.1 Possíveis aproximações para Δl 23
3.2 Pontos de controle da curva de Bezier 24
3.3 Cálculo do sinal de Δl 25
3.4 Comparação entre $\hat{\Delta l}$ e Δl 26
3.5 Sinais PRBS utilizados para excitar o robô simulado 31
3.6 Saída do robô simulado à entradas PRBS 32
3.7 Validação do modelo sem zona-morta para o robô simulado 33
3.8 Sinais Gaussianos utilizados para excitar o robô simulado 34
3.9 Saída do robô simulado à entradas Gaussianas 35
3.10 Evolução da estimação dos parâmetros a_1^L e a_2^L 36
3.11 Evolução da estimação dos parâmetros b_1^L e b_2^L 36
3.12 Evolução da estimação dos parâmetros b_3^L e b_4^L 36
3.13 Evolução da estimação dos parâmetros a_1^Θ e a_2^Θ 37
3.14 Evolução da estimação dos parâmetros b_1^Θ e b_2^Θ 37
3.15 Evolução da estimação dos parâmetros b_3^Θ e b_4^Θ 37
3.16 Evolução da estimação dos parâmetros η_1 e η_2 38
3.17 Validação do modelo com zona-morta para o robô simulado 39
3.18 Predição usando modelo sem zona-morta e entradas de baixa amplitude . 40

4.1 Robô real 42
4.2 Estrutura utilizada no sistema real 42
4.3 Localização do robô na imagem 43
4.4 Análise da precisão do sistema de visão 44
4.5 Tempo de atraso do sistema 45

4.6 Sinais PRBS utilizados para excitar o robô real 46
4.7 Saída do robô à entradas PRBS 47
4.8 Validação do modelo sem zona-morta para o robô real 48
4.9 Sinais Gaussianos utilizados para excitar o robô real 49
4.10 Saída do robô real à entradas gaussianas 50
4.11 Evolução parâmetros a_1^L e a_1^L para o robô real 51
4.12 Evolução parâmetros b_1^L e b_1^L para o robô real 51
4.13 Evolução parâmetros b_3^L e b_4^L para o robô real 51
4.14 Evolução parâmetros a_1^Θ e a_1^Θ para o robô real 52
4.15 Evolução parâmetros b_1^Θ e b_1^Θ para o robô real 52
4.16 Evolução parâmetros b_3^Θ e b_4^Θ para o robô real 52
4.17 Evolução parâmetros η_1 e η_2 para o robô real 53
4.18 Validação do modelo com zona-morta para o robô real 54
4.19 Saídas obtidas do modelo sem zona-morta do robô real com entradas de baixa amplitude 55

Lista de Tabelas

3.1 Parâmetros do modelo sem zona-morta estimados com dados simulados . 31
3.2 Parâmetros do modelo com zona-morta estimados com dados simulados . 38

4.1 Parâmetros do modelo sem zona-morta estimados com dados reais 48
4.2 Parâmetros do modelo com zona-morta estimados com dados do robô real 53

Lista de Símbolos e Abreviaturas

(x,y)	posição do robô
F_i	força aplicada pelo giro da roda i
F_m	força motor
F_{at}	força de atrito
G	centro de massa
J	momento de inércia
$K^L_{a_i}, K^L_{b_i}, K^\Theta_{a_i}, K^\Theta_{b_i}$	constantes formadas durante a derivação do modelo discreto
N	numero de amostras
T_m	torque motor
T_{at}	torque de atrito
Δl	distância percorrida em um período de amostragem
Δl_M	módulo da distância percorrida em um período de amostragem
Ψ	matriz de regressores
Ψ^L	matriz de regressores usado na estimação do modelo de l
Ψ^Θ	matriz de regressores usado na estimação do modelo de θ
β	coeficiente de atrito angular
$\boldsymbol{\theta}$	vetor de parâmetros
$\boldsymbol{\theta}^L$	vetor com parâmetros do modelo de l
$\boldsymbol{\theta}^\Theta$	vetor com parâmetros do modelo de θ

$\boldsymbol{\xi}$	vetor de erros de estimação do modelo
η	limite da zona morta simétrica
η^+	limite da zona morta para entradas positivas
η^-	limite da zona morta para entradas negativas
η_i^L	limites virtuais da zona-morta na estimação recursiva de l
η_i^Θ	limites virtuais da zona-morta na estimação recursiva de θ
$\eta_i^{\max}$	valor máximo de η_i durante a estimação recursiva
γ	parâmetro da curva de Bezier
$\mathbf{B}(t)$	curva de Bezier em função do parâmetro γ
$\mathbf{P}_i$	i-ésimo ponto de uma curva de Bezier
μ	ganho da zona morta simétrica
μ^+	ganho da zona morta para entradas positivas
μ^-	ganho da zona morta para entradas negativas
ω	velocidade angular do robô
ω_0	velocidade angular inicial
ω_i	velocidade angular do eixo acoplado a roda i
ϕ	deslocamento angular do robô
ψ	vetor de regressores
ψ^L	vetor de regressores usado na estimação do modelo de l
ψ^Θ	vetor de regressores usado na estimação do modelo de θ
σ	desvio padrão
θ	orientação do robô
θ_0	orientação inicial

ξ	erro de estimação do modelo
a_i^L, b_i^L, e_i^L	parâmetros a serem identificados relativos a saída l
$a_i^\Theta, b_i^\Theta, e_i^\Theta$	parâmetros a serem identificados relativos a saída θ
b	coeficiente de atrito linear
d	atraso discreto de tempo
d_r	comprimento lateral
$d_{P_0P_1}$	distância entre o ponto $\mathbf{P}_0$ e $\mathbf{P}_1$
$d_{P_2P_3}$	distância entre o ponto $\mathbf{P}_2$ e $\mathbf{P}_3$
e_x,e_y,e_θ,e_l	ruídos de medição do simulador
l	deslocamento linear do robô
l_0	deslocamento linear inicial
m	massa
r_i	raio da roda i
$s(\cdot)$	função degrau
u_i	entrada do sistema que representa o *duty cycle* do PWM aplicado no terminal do atuador acoplado a roda i
$u_i^\star$	saída da zona-morta cuja entrada é u_i
$u_i^{\max}$	valor máximo do sinal de entrada
v	velocidade linear do robô
v_0	velocidade linear inicial
v_i	velocidade linear na roda i
v_x	componente da velocidade linear do robô no eixo X
v_y	componente da velocidade linear do robô no eixo Y
x_{proj}	projeção do ponto (x_k, y_k) no eixo fixado em (x_{k-1}, y_{k-1}) no sentido de θ_{k-1}

Capítulo 1

Introdução

Os robôs há muito tempo deixaram de ser objetos presentes somente na ficção científica e passaram a ser parte de nossa realidade. Vemos exemplos em todos os lugares, desde os tradicionais robôs de linhas de montagem em indústrias automotivas a robôs de auxílio a tarefas domésticas, ou mais além, como aqueles enviados para explorar outros planetas. Por definição, um robô é um dispositivo que atua no ambiente de forma automática, modificando-o de certa forma a cumprir um objetivo ou tarefadiversos tipos existentes, aqueles que conseguem alterar a própria posição no espaço são chamados de robôs móveis. Dentro deste grupo, eles ainda podem ser divididos de acordo com o meio de locomoção: terrestres, aéreos ou marítimos. Dentre estes, os robôs móveis terrestres, também conhecidos como AGV's (*autonomous ground vehicles*) vêm sendo bastante estudados durante os últimos anos.

Os robôs móveis terrestres podem se locomover de várias formas, como por exemplo utilizando pernas, patas, esteiras ou rodas. Além disso, eles geralmente possuem uma gama de sensores que são utilizados para sensoriamento do ambiente, e assim realizar a localização de obstáculos no ambiente de trabalho assim como a própria localização. Os sensores também são diversos: câmeras, sonares, sensores a laser, sensores de odometria, etc. Dentro destes, as câmeras têm apresentado um grande potencial. Com a grande capacidade de processamento dos computadores disponíveis nos dias de hoje, é possível utilizar imagens obtidas de câmeras para extrair diversas características do ambiente em tempo real. Uma das possíveis utilizações de câmeras como sensores é a auto-localização do robô através da busca por marcos no ambiente de trabalho.

Às vezes não é possível embarcar o sensoriamento por câmeras nos robôs, principalmente quando existem restrições de tamanho. Nesses casos pode-se aplicar a lógica inversa: utilizar câmeras em um sistema de visão global, que tem a função de localizar os robôs através de marcos anexados aos mesmos. Um exemplo de aplicação dessa técnica é o sistema de visão utilizado pelas equipes de futebol de robôs das categorias *Mirosot*

e *IEEE Very Small*. Nestas categorias, os robôs são pequenos cubos de 7.5cm de aresta com pouco espaço para sistemas de localização embarcados. Dessa forma, as equipes recorrem a um sistema de visão global, que localiza o robô através de marcas posicionadas na sua parte superior. Geralmente tais robôs possuem somente motores para se deslocar no ambiente, baterias e uma eletrônica embarcada que se comunica com um computador através de radiofrequência e aciona os motores. Os robôs em si são considerados "sem inteligência", sendo um algoritmo em um computador o responsável pelo comportamento inteligente do time. Para tal, o algoritmo faz uso do sistema de visão para localização e de alguma técnica de controle para movimentar os robôs corretamente de acordo com a estratégia da equipe. Na literatura existem diversos algoritmos de controle disponíveis para os diversos tipos de sistemas, sendo o controlador PID o mais simples e mais utilizado na prática. A maioria deles faz uso de algum modelo da planta a ser controlada para ajustar as constantes do controlador.

Existem algumas abordagens conhecidas para se obter o modelo de plantas que se deseja controlar. Uma delas é a modelagem fenomenológica do sistema, também conhecida como modelagem *caixa-branca*. Nessa modelagem são utilizados os conhecimentos sobre a física e dinâmica do sistema para se derivar um modelo adequado. Na prática porém é comum encontrar plantas onde sabe-se pouco sobre as relações entre as suas entradas e saídas, tornando-se difícil obter o modelo do sistema através da modelagem *caixa-branca*. Uma alternativa à modelagem fenomenológica é a identificação de sistemas, onde são utilizados dados de entrada e saída de uma planta para inferir qual o modelo que melhor descreve esses dados. Quando não se tem nenhuma informação disponível sobre o sistema além desses dados, a identificação é chamada de identificação *caixa-preta*. Apesar de não se saberem as relações causa-efeito do sistema, o modelador tem que fazer suposições de como as entradas e saídas se relacionam, geralmente através de algum modelo paramétrico, e fazer isso somente a partir dos dados não é trivial. Porém, na maioria dos casos, o modelador tem alguma informação sobre o comportamento do sistema, mas não o suficiente para se obter o modelo fenomenológico. Nesses casos ele pode utilizar a identificação *caixa-cinza*, onde se utiliza essa informação para obter o modelo paramétrico que mais de aproxime do sistema, e então utiliza-se alguma técnica de estimação para obter os valores dos parâmetros do modelo.

Na teoria, pode-se tentar obter um modelo que represente todas as dinâmicas possíveis do sistema. Na prática os sistemas são modelados para representar as dinâmicas relativas ao problema, dentro de uma margem aceitável de erro. Os modelos lineares são mais utilizados, devido a sua simplicidade. Porém, a maioria dos sistemas reais são não-lineares, e não são bem aproximados completamente por modelos lineares. Uma abordagem utili-

zada nesses casos é obter modelos lineares para certos pontos de operação dos sistemas, o que é aceitável na maioria dos casos, dependendo da exatidão que se deseje obter do modelo.

Algumas não-linearidades podem ser consideradas como sendo *de entrada*, por agirem no sinal de entrada antes de serem aplicadas no sistema, ou *de saída*, por agir no sinal de saída do sistema. Sistemas com tais características são conhecidos respectivamente como *sistemas de Hammerstein* e *sistemas de Wiener* [Aguirre 2007]. Considerando os sistemas de Hammerstein, as não-linearidades de entrada podem ser dos tipos: saturação, zona-morta, folga, e histerese. Dentre essas, a zona-morta é considerada como sendo a faixa de entrada que não produz resposta dinâmica na saída do sistema, podendo ser variante no tempo e é comum em todos os tipos de atuadores [Tao & Kokotović 1994]. Em um robô móvel acionado por atuadores elétricos, a zona-morta é em geral decorrente de forças de reação causadas pelos componentes do atrito nos eixos de movimento [Sanca 2006]. Considerar tais efeitos no modelo de um robô móvel é importante quando se deseja obter uma melhor precisão do modelo para entradas de baixa amplitude.

1.1 Objetivo

O objetivo deste trabalho é realizar a modelagem e identificação de um robô móvel com acionamento diferencial utilizando a abordagem *caixa-cinza*. O sistema consiste em duas entradas, representando o sinal aplicado em cada um dos dois motores utilizados pelo robô para se deslocar, e tem como saída a sua posição e orientação. Essa saída será reescrita em termos de deslocamento linear e angular, evitando assim ter de modelar os efeitos da restrição não-holonômica. Considerando o robô como um sistema de Hammerstein MIMO, são obtidos modelos paramétricos para cada saída do sistema onde aparecem os parâmetros tanto das parcelas não-lineares quanto da parcela linear e então todos estes parâmetros são identificados simultaneamente utilizando técnicas recursivas de estimação.

1.2 Justificativa

Em um trabalho passado realizado no Laboratório de Robótica da UFRN, foi realizada a identificação de um modelo para uma versão antiga do robô da Equipe POTI de Futebol de Robôs. O modelo foi obtido em funções de transferência discreta e esperava-se poder encontrar o equivalente em funções de transferência contínuas e assim estimar as gran-

dezas desconhecidas do robô. Porém, tal procedimento não foi possível pois o modelo estimado não tinha equivalente contínuo. A suposição é que isso ocorreu devido a zona-morta do sistema não ter sido considerada. Por isso a necessidade de estimar também os parâmetros da zona-morta junto com o modelo do sistema.

1.3 Revisão da literatura

A identificação de robôs móveis utilizados em competições de futebol de robôs foi realizada em Pereira (2000) e Guerra (2005). No primeiro o sistema foi desacoplado e foram identificados modelos MISO para cada saída do sistema. Foi utilizado um modelo não-linear devido às restrições não holonômicas do robô, e um controle foi desenvolvido baseado no modelo obtido. No segundo trabalho foi realizada uma modelagem fenomenológica do robô e o sistema foi identificado sem desacoplamento das variáveis de saída. Porém, a não-linearidade devida à restrição não-holonômica foi evitada através da mudança das variáveis de saída do modelo. Nenhum deles considera não-linearidades de entrada no sistema.

Uma identificação de um manipulador robótico com banda-morta e zona-morta foi realizada em Malik et al. (2009). A identificação do manipulador é realizada em dois estágios, primeiro utilizando a informação de regime permanente do sistema para identificar alguns parâmetros e os restantes são identificados posteriormente utilizando a resposta transitória não afetada pela zona-morta em um algoritmo de mínimos quadrados não-linear.

Em Er-Wei & Bai (2002) é feito um estudo da identificação de várias não-linearidades de entrada quando a sua estrutura é conhecida. É proposta uma técnica determinística para os casos em que as não-linearidades são parametrizadas com um parâmetro e um algoritmo de identificação baseado na análise de correlação é apresentado para o caso genérico.

Uma modelagem não-linear e estimação de parâmetros para robôs móveis omnidirecionais foi proposto em Conceição et al. (2009), utilizando-se de três procedimentos experimentais. Em cada experimento o robô é colocado em situações que permitem simplificar o seu modelo dinâmico, evidenciando a característica que se deseja estimar, usando para isso o método de mínimos quadrados. Dessa forma, é possível obter os coeficientes de atrito viscoso, os coeficientes de atrito de *coulomb* e o momento de inércia do robô.

A identificação de sistemas lineares em conjunto com suas não-linearidades de entrada foi abordada em [Vörös 2004, Vörös 2003, Vörös 2006] para sistemas de Hammerstein com zonas-mortas descontínuas, e em Vörös (2001) para sistemas de Wiener. Todos

os trabalhos utilizados consideraram sistemas SISO e utilizam uma técnica chamada de separação do termo chave para escrever os parâmetros da zona-morta e da parte linear do sistema em uma só equação e utiliza algum algoritmo de estimação recursiva para estimar simultaneamente todos os parâmetros do modelo.

1.4 Organização do trabalho

O restante da dissertação está dividida como descrito a seguir. O capítulo 2 trata de mostrar a modelagem do robô e das zonas-mortas utilizadas nesse trabalho. O capítulo 3 descreve o processo de parametrização e estimação do modelo sem e com zona-morta, assim como o a descrição do robô simulado utilizado para validar alguns procedimentos do método descrito. Além disso, também são mostrados resultados obtidos com dados do robô simulado. No capítulo 4 é feita uma descrição do sistema utilizado e são apresentados o resultado da identificação para o sistema real. Finalmente, no capítulo 5 são feitas considerações sobre as conclusões do trabalho e são mostradas sugestões para possíveis trabalhos futuros.

Capítulo 2

Modelagem do sistema

Para realizar a identificação de um sistema usando métodos paramétricos é necessário primeiro ter um modelo matemático que espera-se que represente bem a dinâmica do sistema. Neste capítulo iremos obter tal modelo, que será utilizado posteriormente para estimação dos parâmetros. Inicialmente será modelado um robô considerando-se entradas sem não-linearidades. Logo após será descrito o modelo da zona-morta utilizada neste trabalho, sendo ela incorporada ao modelo linear. O modelo derivado no presente capítulo é baseado nos modelos utilizados em Pereira (2000) e Guerra (2005).

2.1 Modelo cinemático

O robô utilizado neste trabalho consiste em um veículo com duas rodas fixas controladas por motores independentes como mostrado na figura 2.1, com seu centro de massa localizado no centro do eixo entre as duas rodas. O robô é um sistema com dois graus de liberdade cujas saídas são representadas pela sua posição (x,y) e a orientação θ. Considera-se que:

$$v_1 = \omega_1 r_1 \qquad\qquad v_2 = \omega_2 r_2 \tag{2.1}$$

onde v_i é a velocidade linear da roda i, ω_i é a velocidade angular do eixo acoplado a roda i e r_i é o raio da roda i, onde $i = 1$ indica a roda direita e $i = 2$ indica a roda esquerda. Temos ainda que:

$$v = \frac{v_1 + v_2}{2} \qquad\qquad \omega = \frac{v_1 - v_2}{d_r} \tag{2.2}$$

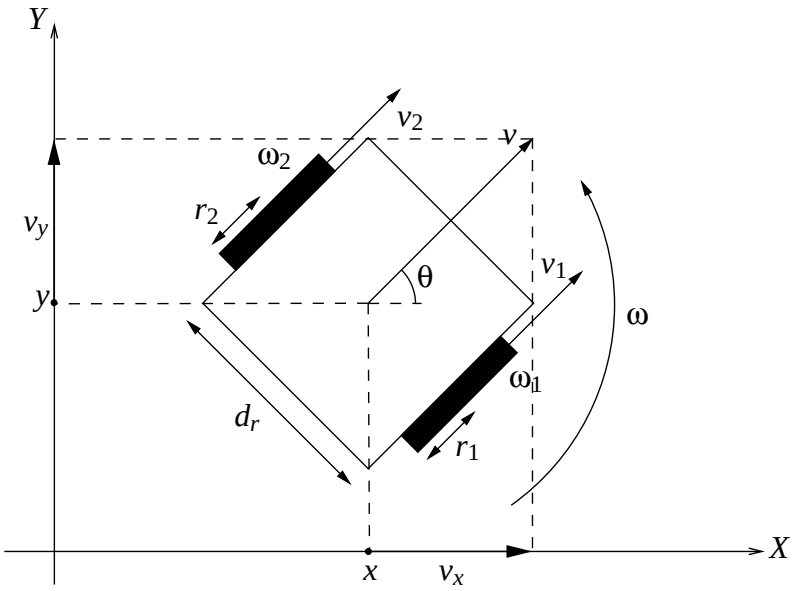

Figura 2.1: A estrutura do robô utilizado, onde: v e ω são as velocidades linear e angular do robô; v_i é a velocidade linear da roda i; v_x é a componente de velocidade linear no eixo X; v_y é a componente de velocidade linear no eixo Y; ω_i e r_i são respectivamente a velocidade angular e o raio da roda i; d_r é o comprimento da lateral do robô; (x,y) representa a pose do robô e θ a sua orientação.

onde v e ω são as velocidades linear e angular no centro do robô e d_r o comprimento da lateral do robô. Decompondo a velocidade nos eixos X e Y, temos:

$$\begin{aligned} v_x &= v\cos(\theta) \\ v_y &= v\text{sen}(\theta) \end{aligned} \tag{2.3}$$

onde θ é a orientação do robô. Logo, temos o seguinte modelo cinemático:

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{\theta} &= \omega \end{aligned} \tag{2.4}$$

e a restrição não-holonômica do sistema pode ser representada por:

$$\dot{x}\text{sen}(\theta) - \dot{y}\cos(\theta) = 0 \tag{2.5}$$

2.2 Modelo linear equivalente

O modelo (2.4) representa a cinemática do robô, porém esse modelo é não-linear, devido à restrição não-holonômica representada pela dependência de θ nas equações de x e y, através de v_x e v_y. Com o objetivo de utilizar algoritmos de controle mais simples e os resultados matemáticos existentes para sistemas lineares, podemos reescrever esse modelo não-linear como uma representação linear do robô. Tal mudança foi feita através de uma substituição das variáveis de saída: a configuração do robô não vai ser mais descrita pela sua posição (x,y) e sua orientação θ, sendo agora expressa pelo deslocamento linear l e pelo deslocamento angular ϕ. O modelo cinemático com as novas saídas é obtido reescrevendo (2.4) como:

$$\begin{aligned} \dot{l} &= v \\ \dot{\phi} &= \omega \end{aligned} \tag{2.6}$$

As novas saídas l e ϕ representam o quanto o robô se deslocou em função das velocidades v e ω, respectivamente. Por exemplo, para um robô parado girando sobre o próprio eixo, temos l constante, pois o robô não alterou sua posição, e ϕ crescente caso o robô gire no sentido horário, ou decrescente caso ele gire no sentido anti-horário. Se o robô estivesse andando em uma linha reta, teríamos ϕ constante, pois o robô não gira sobre seu eixo, e l crescente caso o robô se desloque para frente, ou decrescente caso ele se desloque para trás.

Observe que a saída do sistema ainda é (x,y,θ) e somente a saída do modelo foi modificada. A saída ϕ pode ser obtida diretamente de θ, considerando a condição inicial $\phi_0 = \theta_0$, e eliminando a descontinuidade em $\pm\pi$ que existe em θ. Com isso, temos que $\phi = \theta$ e podemos reescrever o modelo cinemático (2.6) como:

$$\begin{aligned} \dot{l} &= v \\ \dot{\theta} &= \omega \end{aligned} \tag{2.7}$$

Um problema inerente a esta nova representação é o fato da grandeza l ser não-mensurável. Caso seja utilizado um sistema de visão global (como o utilizado neste trabalho), a configuração do robô está disponível somente em instantes de tempo determinados, não sendo possível descrever qual a trajetória foi percorrida pelo robô entre duas amostras do sistema (um exemplo desse problema pode ser visto na figura 3.1). Torna-se necessário então o uso de um algoritmo de estimação de l_k. Tal algoritmo será descrito mais adiante,

na seção 3.4. É importante frisar que o modelo (2.7) é linear e não foi realizada nenhuma simplificação para obtê-lo.

Uma última consideração sobre a mudança de variáveis de saída: ao realizar um procedimento desse tipo, normalmente espera-se que em algum momento seja feito o procedimento inverso, ou seja, obter a posição (x,y) do robô baseado na distância percorrida l do modelo, já que $\phi = \theta$. Porém, tal procedimento não será realizado e usaremos de fato a saída modificada, sendo isso possível devido ao sistema de controle utilizado. Esse sistema garante que a pose do robô converge para a pose de destino ao escrever a pose de destino em coordenadas polares em relação a posição atual do robô e utilizando dois controladores, sendo um para minimizar o erro entre a orientação e a orientação do destino, e o outro para minimizar a distância entre a posição atual e uma projeção do destino sobre uma reta que tem a mesma orientação que o robô e passa pela posição atual [Vieira 2006]. Ou seja, é possível levar o robô para a pose desejada utilizando as medidas de distância percorrida e a orientação do robô, não sendo necessário converter a saída l do modelo de volta para x e y.

2.3 Modelo dinâmico

A figura 2.2 mostra as forças que atuam no robô, sejam elas geradas pelo acionamento dos motores ou pelo atrito. Pelo diagrama, temos que:

$$\begin{aligned} F_m - F_{at} &= m\dot{v} \\ T_m - T_{at} &= J\dot{\omega} \end{aligned} \tag{2.8}$$

onde m é a massa do robô, J é o momento de inércia, F_m é a força motor, T_m é o torque motor, e F_{at} e T_{at} são a força e torque de atrito dinâmico, sendo estas últimas dadas por:

$$F_{at} = bv \qquad\qquad T_{at} = \beta\omega \tag{2.9}$$

sendo b e β respectivamente os coeficientes de atrito linear e angular do robô. Juntando (2.8) e (2.9) obtemos

$$\begin{aligned} m\dot{v} + bv &= F_m \\ J\dot{\omega} + \beta\omega &= T_m \end{aligned} \tag{2.10}$$

As equações em (2.10) são equações diferenciais ordinárias (EDO) de primeiro grau,

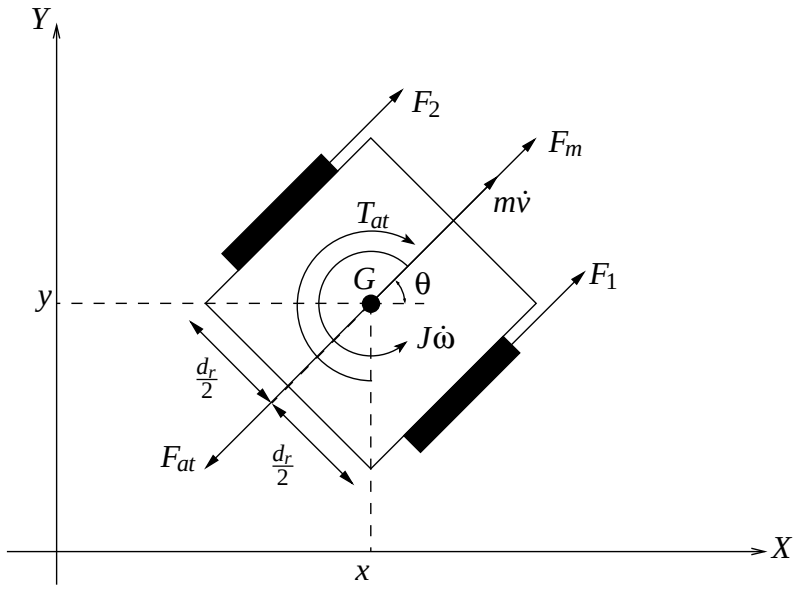

Figura 2.2: Diagrama de forças aplicadas ao robô, onde: F_i é a força gerada pelo giro da roda i; F_m é a força motor; F_{at} é a força de atrito; T_{at} é o torque de atrito; $m\dot{v}$ é a força resultante; $J\dot{\omega}$ é o torque resultante; G é o centro de massa; d_r é o comprimento lateral; (x,y) representa a pose do robô e θ a sua orientação.

e apresentam as seguintes soluções:

$$\begin{aligned} v(t) &= \frac{F_m}{b} + (v_0 - \frac{F_m}{b})e^{-\frac{b}{m}t} \\ \omega(t) &= \frac{T_m}{\beta} + (\omega_0 - \frac{T_m}{\beta})e^{-\frac{\beta}{J}t} \end{aligned} \tag{2.11}$$

onde v_0 e ω_0 são as velocidades linear e angular iniciais. As equações em (2.11) podem ser integradas para obtermos as equações de $l(t)$ e $\theta(t)$. Dessa forma, temos:

$$\begin{aligned} l(t) &= l_0 + \frac{m}{b}\left(1 - e^{-\frac{b}{m}t}\right)v_0 + \frac{1}{b}\left[t - \frac{m}{b}\left(1 - e^{-\frac{b}{m}t}\right)\right]F_m \\ \theta(t) &= \theta_0 + \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}t}\right)\omega_0 + \frac{1}{\beta}\left[t - \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}t}\right)\right]T_m \end{aligned} \tag{2.12}$$

onde l_0 e θ_0 são respectivamente o deslocamento linear inicial e a orientação inicial do robô.

As equações (2.11) e (2.12) descrevem a evolução contínua de l, θ e respectivas velocidades v e ω. Porém, apesar do sistema ser contínuo, a leitura da posição e orientação dos robôs ocorre em instantes discretos de tempo, sendo portanto interessante obter um modelo discreto do sistema. As equações (2.11) e (2.12) podem ser reescritas na forma

discreta como:

$$\begin{aligned} v_k &= \left(e^{-\frac{b}{m}T}\right) v_{k-1} + \frac{1}{b}\left(1 - e^{-\frac{b}{m}T}\right) F_{m,k-1} \\ \omega_k &= \left(e^{-\frac{\beta}{J}T}\right) \omega_{k-1} + \frac{1}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right) T_{m,k-1} \end{aligned} \tag{2.13}$$

$$\begin{aligned} l_k &= l_{k-1} + \frac{m}{b}\left(1 - e^{-\frac{b}{m}T}\right) v_{k-1} + \frac{1}{b}\left[T - \frac{m}{b}\left(1 - e^{-\frac{b}{m}T}\right)\right] F_{m,k-1} \\ \theta_k &= \theta_{k-1} + \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right) \omega_{k-1} + \frac{1}{\beta}\left[T - \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right)\right] T_{m,k-1} \end{aligned} \tag{2.14}$$

onde assume-se a causalidade no sistema ao utilizar $F_{m,k-1}$ e $T_{m,k-1}$. Ao escrever as equações (2.13) e (2.14) em variáveis de estado, temos o seguinte:

$$\begin{aligned} \begin{bmatrix} l_k \\ v_k \end{bmatrix} &= \begin{bmatrix} 1 & K_{a_1}^L \\ 0 & K_{a_2}^L \end{bmatrix} \begin{bmatrix} l_{k-1} \\ v_{k-1} \end{bmatrix} + \begin{bmatrix} K_{b_1}^L \\ K_{b_2}^L \end{bmatrix} F_{m,k-1} \\ l_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} l_k \\ v_k \end{bmatrix} \end{aligned} \tag{2.15}$$

$$\begin{aligned} \begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix} &= \begin{bmatrix} 1 & K_{a_1}^\Theta \\ 0 & K_{a_2}^\Theta \end{bmatrix} \begin{bmatrix} \theta_{k-1} \\ \omega_{k-1} \end{bmatrix} + \begin{bmatrix} K_{b_1}^\Theta \\ K_{b_2}^\Theta \end{bmatrix} T_{m,k-1} \\ \theta_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_k \\ \omega_k \end{bmatrix} \end{aligned} \tag{2.16}$$

onde as constantes K_i^L e K_i^Θ são dadas por:

$$K_{a_1}^L = \frac{m}{b}\left(1 - e^{-\frac{b}{m}T}\right) \qquad K_{b_1}^L = \frac{1}{b}\left[T - \frac{m}{b}\left(1 - e^{-\frac{b}{m}T}\right)\right]$$

$$K_{a_2}^L = \left(e^{-\frac{b}{m}T}\right) \qquad K_{b_2}^L = \frac{1}{b}\left(1 - e^{-\frac{b}{m}T}\right)$$

$$K_{a_1}^\Theta = \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right) \qquad K_{b_1}^\Theta = \frac{1}{\beta}\left[T - \frac{J}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right)\right]$$

$$K_{a_2}^\Theta = \left(e^{-\frac{\beta}{J}T}\right) \qquad K_{b_2}^\Theta = \frac{1}{\beta}\left(1 - e^{-\frac{\beta}{J}T}\right)$$

Podemos reescrever (2.15) em função de transferência discreta da seguinte forma:

$$\begin{aligned} z\mathbf{X}_l(z) &= \mathbf{A}_l\mathbf{X}_l(z) + \mathbf{B}_l F_m(z) \\ L(z) &= \mathbf{C}\mathbf{X}_l(z) \end{aligned} \tag{2.17}$$

onde:

$$\mathbf{A}_l = \begin{bmatrix} 1 & K_{a_1}^L \\ 0 & K_{a_2}^L \end{bmatrix} \qquad \mathbf{B}_l = \begin{bmatrix} K_{b_1}^L \\ K_{b_2}^L \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad \mathbf{X}_l(z) = \begin{bmatrix} L(z) \\ V(z) \end{bmatrix}$$

e logo:

$$L(z) = \mathbf{C}(z\mathbf{I} - \mathbf{A}_l)^{-1}\mathbf{B}_l F_m(z) \tag{2.18}$$

Desenvolvendo-se o termo $\mathbf{C}(z\mathbf{I} - \mathbf{A}_l)^{-1}\mathbf{B}_l$ obtemos o seguinte:

$$L(z) = \frac{K_{b_1}^L z + K_{b_2}^L}{z^2 - K_{a_1}^L z - K_{a_2}^L} F_m(z) \tag{2.19}$$

que pode ser reescrita em equações de diferenças como:

$$l_k = K_{a_1}^L l_{k-1} + K_{a_2}^L l_{k-2} + K_{b_1}^L F_{m,k-1} + K_{b_2}^L F_{m,k-2} \tag{2.20}$$

e fazendo um desenvolvimento semelhante a partir de (2.16) podemos escrever θ em equações de diferenças como:

$$\theta_k = K_{a_1}^\Theta \theta_{k-1} + K_{a_2}^\Theta \theta_{k-2} + K_{b_1}^\Theta T_{m,k-1} + K_{b_2}^\Theta T_{m,k-2} \tag{2.21}$$

Temos então em (2.20) e (2.21) as equações de diferenças que descrevem a saída do modelo em função das medições anteriores e das forças e torques no robô. Da figura 2.2, temos que:

$$F_m = \frac{F_1 + F_2}{2} \qquad T_m = \frac{d_r}{2}(F_1 - F_2) \tag{2.22}$$

onde F_i é a força aplicada ao acionar o motor acoplado a roda i. As forças geradas pelos motores depende das tensões aplicadas nos seus terminais de entrada, que é o valor DC de um sinal PWM cujo *duty cycle* é definido pelo sinal u_i, onde $u_i = 0$ indica um *duty cycle* de 0% e $u_i = 1$ indica 100%. A relação entre a força F_i e a tensão gerada por u_i é dinâmica, porém vamos considerar que essa dinâmica é rápida em relação a dinâmica de movimentação do robô e dessa forma vamos supor que as forças F_1 e F_2 são linearmente

relacionadas com as entradas u_1 e u_2. Ou seja:

$$\begin{aligned} F_{1,k} &\approx c_1 u_{1,k-d} \\ F_{2,k} &\approx c_2 u_{2,k-d} \end{aligned} \tag{2.23}$$

onde d representa o atraso ou tempo morto do sistema e c_1, c_2 são constantes. Substituindo (2.23) em (2.20) e (2.21), temos o seguinte modelo geral para o robô:

$$\begin{aligned} l_k &= K_{a_1}^L l_{k-1} + K_{a_2}^L l_{k-2} + \frac{K_{b_1}^L}{2}\left(c_1 u_{1,k-1-d} + c_2 u_{2,k-1-d}\right) \\ &\quad + \frac{K_{b_2}^L}{2}\left(c_1 u_{1,k-2-d} + c_2 u_{2,k-2-d}\right) \\ \theta_k &= K_{a_1}^\Theta \theta_{k-1} + K_{a_2}^\Theta \theta_{k-2} + \frac{K_{b_1}^\Theta d_r}{2}\left(c_1 u_{1,k-1-d} - c_2 u_{2,k-1-d}\right) \\ &\quad + \frac{K_{b_2}^\Theta d_r}{2}\left(c_1 u_{1,k-2-d} - c_2 u_{2,k-2-d}\right) \end{aligned} \tag{2.24}$$

2.4 Modelo com zona-morta

Apesar da representação linear desenvolvida na seção 2.2 eliminar a não-linearidade cinemática, o sistema real ainda apresenta comportamentos não-lineares devido a efeitos do atrito na transmissão dos motores e do atrito do robô com o solo. Tais efeitos devem ser levados em consideração quando é desejado um modelo mais preciso e confiável, principalmente para sistemas que são excitados com sinais de baixa amplitude. Neste trabalho, iremos considerar que a não-linearidade resultante destes efeitos no robô é do tipo zona-morta.

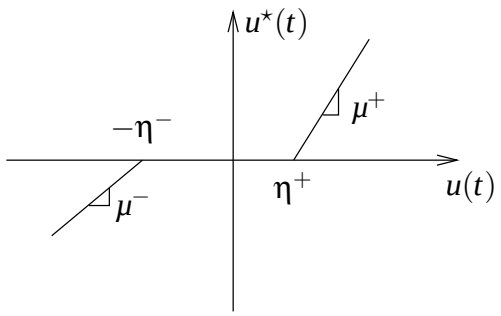

Figura 2.3: Modelo da zona-morta, com limites $(-\eta^-, \eta^+)$ e ganhos (μ^-, μ^+).

A zona-morta consiste em uma faixa dos possíveis valores do sinal de entrada que não gera resposta do sistema. Uma representação do modelo da zona-morta é mostrado na figura 2.3 [Vörös 2003]. Nesse modelo, temos que $u(t)$ é a entrada e $u^\star(t) = f^\star(u(t))$ é a saída da zona-morta, η^+ e μ^+ são respectivamente o limite e o ganho da zona-morta para $u(t) > 0$, e η^- e μ^- o equivalente para $u(t) < 0$. Considera-se que $\eta^+, \eta^- \geq 0$ e $\mu^+, \mu^- > 0$. O modelo da figura 2.3 pode ser definido matematicamente como:

$$u^\star(t) = f^\star(u(t)) = \begin{cases} \mu^+(u(t) - \eta^+) & \text{para } u(t) \geq \eta^+ \\ 0 & \text{para } -\eta^- < u(t) < \eta^+ \\ \mu^-(u(t) + \eta^-) & \text{para } u(t) \leq -\eta^- \end{cases} \tag{2.25}$$

Sendo $s(\cdot)$ a função degrau unitário, podemos reescrever (2.25) como:

$$u^\star(t) = \mu^+(u(t) - \eta^+)s(u(t) - \eta^+) + \mu^-(u(t) + \eta^-)s(-u(t) - \eta^-) \tag{2.26}$$

em sua forma contínua, ou na forma discreta:

$$u_k^\star = \mu^+(u_k - \eta^+)s(u_k - \eta^+) + \mu^-(u_k + \eta^-)s(-u_k - \eta^-) \tag{2.27}$$

Para o caso de uma zona-morta simétrica, temos $\eta^+ = \eta^-$ e $\mu^+ = \mu^-$ e (2.27) pode ser reescrita como:

$$u_k^\star = \mu u_k \left[s(u_k - \eta) + s(-u_k - \eta)\right] + \mu\eta \left[-s(u_k - \eta) + s(-u_k - \eta)\right] \tag{2.28}$$

Em um sistema SISO com zona-morta, uma entrada $-\eta < u < \eta$ não gera resposta na saída, e consequentemente têm-se o mesmo efeito de aplicar $u = 0$. Logo, pode-se considerar que o efeito da zona-morta altera somente o sinal de entrada antes de ser aplicado realmente ao sistema, ou seja, pode-se considerar a zona-morta como uma não-linearidade de entrada. Um sistema com tais características pode ser representado pelo *modelo de Hammerstein*. A figura 2.4 mostra um esquema desse modelo para um sistema SISO.

Até o presente momento u_1 e u_2 foram consideradas como sendo as entradas do robô, mas isso não é necessariamente verdade. Como o robô possui duas entradas, teremos uma zona-morta associada a cada entrada e as saídas $u_1^\star$ e $u_2^\star$ serão de fato as entradas do modelo linear, como mostrado na figura 2.5. Podemos então modificar as equações em (2.24) substituindo as entradas u_1 e u_2 por $u_1^\star$ e $u_2^\star$, resultando finalmente no modelo

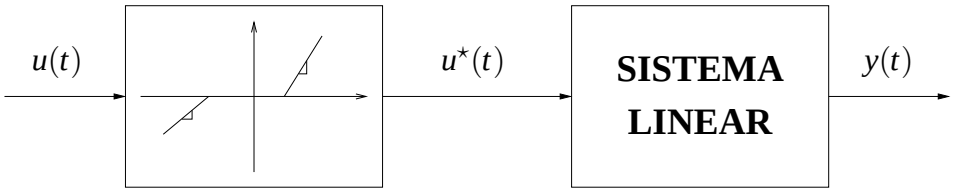

Figura 2.4: Modelo de Hammerstein aplicado a um sistema SISO. O modelo divide o sistema em um bloco não-linear em cascata com um bloco de linear. O sinal intermediário $u^\star$ é apenas uma representação matemática e não pode ser mensurado.

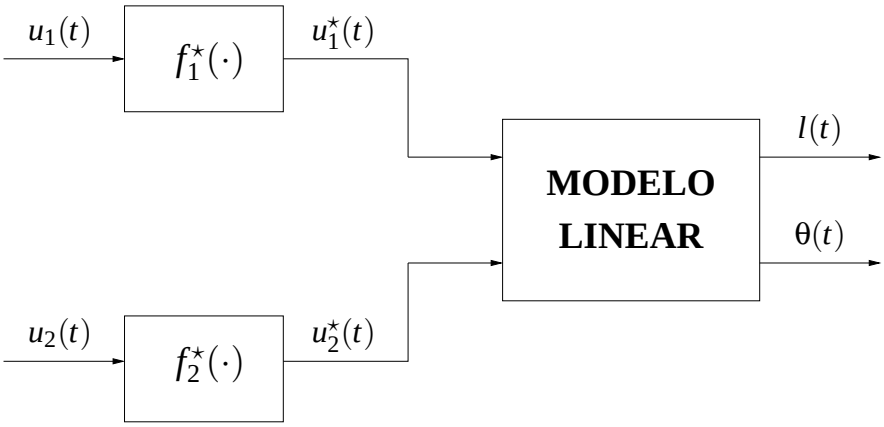

Figura 2.5: O robô descrito em modelos de Hammerstein. Os sinais virtuais $u_i^\star = f_i^\star(u_i)$ são utilizados como entradas para o modelo linear, onde $f_i^\star(\cdot)$ é a função que define a zona-morta da entrada i.

do robô com zona-morta:

$$
\begin{aligned}
l_k &= K_{a_1}^L l_{k-1} + K_{a_2}^L l_{k-2} + \frac{K_{b_1}^L}{2}\left(c_1 u_{1,k-1-d}^\star + c_2 u_{2,k-1-d}^\star\right) \\
&\quad + \frac{K_{b_2}^L}{2}\left(c_1 u_{1,k-2-d}^\star + c_2 u_{2,k-2-d}^\star\right) \\
\theta_k &= K_{a_1}^\Theta \theta_{k-1} + K_{a_2}^\Theta \theta_{k-2} + \frac{K_{b_1}^\Theta d_r}{2}\left(c_1 u_{1,k-1-d}^\star - c_2 u_{2,k-1-d}^\star\right) \\
&\quad + \frac{K_{b_2}^\Theta d_r}{2}\left(c_1 u_{1,k-2-d}^\star - c_2 u_{2,k-2-d}^\star\right)
\end{aligned}
\tag{2.29}
$$

sendo $u_i^\star$ definido pela equação (2.28).

Temos então dois modelos para considerar: o modelo (2.24) que descreve o robô sem zona-morta e o modelo definido pelas equações (2.29) e (2.28). No próximo capítulo, ambos os modelos serão parametrizados e será descrito o método utilizado para estimar a distância percorrida pelo robô.

Capítulo 3

Estimação paramétrica

No capítulo 2 foi derivado um modelo do robô móvel em questão neste trabalho. No modelo foram consideradas as não-linearidades de entrada e, através da mudança nas variáveis de saída, eliminou-se a não linearidade inerente à restrição não-holonômica do sistema. Porém, o seu uso está condicionado a uma boa estimação do valor do deslocamento linear l.

Este capítulo irá tratar de como proceder para estimar os parâmetros do modelo do robô. Para tal, o modelo derivado no capítulo 2 será parametrizado, afim de ser utilizado em um método de estimação. Será descrito um simulador do robô móvel, utilizado para validar o método de estimação de l, também descrito neste capítulo. Logo após será mostrado como identificar os parâmetros do modelo utilizando o método de mínimos quadrados e por fim, serão mostrados exemplos de modelos identificados a partir de dados do robô simulado.

3.1 Parametrização do modelo linear

O modelo (2.24) descreve fisicamente o sistema, considerando-se que é possível se obter o valor de l. O problema do modelo é que os valores de alguns parâmetros físicos são difíceis se obter, como as constantes de atrito, o momento de inércia e os ganhos dos motores. Para contornar essas dificuldades, pode-se parametrizar o modelo para que as incógnitas sejam estimadas através de dados experimentais. Pode-se então reescrever o modelo (2.24) como:

$$\begin{aligned} l_k &= a_1^L l_{k-1} + a_2^L l_{k-2} + b_1^L u_{1,k-1-d} + b_2^L u_{2,k-1-d} + b_3^L u_{1,k-2-d} + b_4^L u_{2,k-2-d} \\ \theta_k &= a_1^\Theta \theta_{k-1} + a_2^\Theta \theta_{k-2} + b_1^\Theta u_{1,k-1-d} + b_2^\Theta u_{2,k-1-d} + b_3^\Theta u_{1,k-2-d} + b_4^\Theta u_{2,k-2-d} \end{aligned} \tag{3.1}$$

onde as constantes K_i^L e K_i^Θ, a distância lateral do robô d_r e as constantes dos motores c_i estão contidas dentro dos parâmetros a_i e b_i, onde mais uma vez os índices L e Θ representam respectivamente os parâmetros relacionados com as saídas l e θ.

3.2 Parametrização do modelo com zona-morta

O modelo parametrizado com a zona-morta pode ser obtido fazendo-se a substituição do termo chave [Vörös 2003]. Para tal, primeiro, o modelo (2.29) é parametrizado de forma semelhante ao realizado na seção 3.1, resultando em:

$$\begin{aligned} l_k &= a_1^L l_{k-1} + a_2^L l_{k-2} + b_1^L u^\star_{1,k-1-d} + b_2^L u^\star_{2,k-1-d} + b_3^L u^\star_{1,k-2-d} + b_4^L u^\star_{2,k-2-d} \\ \theta_k &= a_1^\Theta \theta_{k-1} + a_2^\Theta \theta_{k-2} + b_1^\Theta u^\star_{1,k-1-d} + b_2^\Theta u^\star_{2,k-1-d} + b_3^\Theta u^\star_{1,k-2-d} + b_4^\Theta u^\star_{2,k-2-d} \end{aligned} \tag{3.2}$$

Observe que cada sinal $u_1^\star$ e $u_2^\star$ aparece duas vezes em cada equação de (3.2). A substituição do termo chave consiste em substituir apenas uma de suas instâncias em cada equação (ou seja, os termos chave), mantendo-se os demais termos. Dessa forma, substituindo alguns termos $u_i^\star$ de (3.2) pela equação (2.28), temos:

$$\begin{aligned} l_k =& \underline{a_1^L} l_{k-1} + \underline{a_2^L} l_{k-2} \\ &+ \underline{b_1^L \mu_1} u_{1,k-1-d} \left[s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_1^L \mu_1 \eta_1} \left[-s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_2^L \mu_2} u_{2,k-1-d} \left[s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_2^L \mu_2 \eta_2} \left[-s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_3^L} u^\star_{1,k-2-d} + \underline{b_4^L} u^\star_{2,k-2-d} \end{aligned} \tag{3.3a}$$

$$\begin{aligned} \theta_k =& \underline{a_1^\Theta} \theta_{k-1} + \underline{a_2^\Theta} \theta_{k-2} \\ &+ \underline{b_1^\Theta \mu_1} u_{1,k-1-d} \left[s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_1^\Theta \mu_1 \eta_1} \left[-s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_2^\Theta \mu_2} u_{2,k-1-d} \left[s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_2^\Theta \mu_2 \eta_2} \left[-s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_3^\Theta} u^\star_{1,k-2-d} + \underline{b_4^\Theta} u^\star_{2,k-2-d} \end{aligned} \tag{3.3b}$$

onde os termos sublinhados são os parâmetros a serem identificados.

Percebe-se que os ganhos μ_i sempre aparecem nas equações de (3.3) acompanhados de algum termo b_i^Θ ou b_i^L da parte linear do modelo, tornando impossível estimá-los iso-

ladamente. De fato, os parâmetros b_i representam os ganhos associados às entradas do sistema para cada saída e, sem nenhuma perda para o modelo final, pode-se considerar que $\mu_i = 1$. Se na prática $\mu_i \neq 1$, a estimação irá tratar de adaptar o valor do parâmetro b_i associado de forma a encontrar o valor que melhor adeque o modelo ao sistema. Assim sendo, o modelo com zona-morta parametrizado é dado por:

$$\begin{aligned} l_k =& \underline{a_1^L} l_{k-1} + \underline{a_2^L} l_{k-2} \\ &+ \underline{b_1^L} u_{1,k-1-d} \left[s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{e_1^L} \left[-s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_2^L} u_{2,k-1-d} \left[s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{e_2^L} \left[-s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_3^L} u^\star_{1,k-2-d} + \underline{b_4^L} u^\star_{2,k-2-d} \end{aligned} \tag{3.4a}$$

$$\begin{aligned} \theta_k =& \underline{a_1^\Theta} \theta_{k-1} + \underline{a_2^\Theta} \theta_{k-2} \\ &+ \underline{b_1^\Theta} u_{1,k-1-d} \left[s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{e_1^\Theta} \left[-s(u_{1,k-1-d} - \eta_1) + s(-u_{1,k-1-d} - \eta_1) \right] \\ &+ \underline{b_2^\Theta} u_{2,k-1-d} \left[s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{e_2^\Theta} \left[-s(u_{2,k-1-d} - \eta_2) + s(-u_{2,k-1-d} - \eta_2) \right] \\ &+ \underline{b_3^\Theta} u^\star_{1,k-2-d} + \underline{b_4^\Theta} u^\star_{2,k-2-d} \end{aligned} \tag{3.4b}$$

onde:

$$e_1^L = b_1^L \eta_1 \qquad e_2^L = b_2^L \eta_2 \qquad e_1^\Theta = b_1^\Theta \eta_1 \qquad e_2^\Theta = b_2^\Theta \eta_2 \tag{3.5}$$

3.3 Simulador do sistema

Um programa que simula o movimento de um robô móvel como o descrito no capítulo 2 foi desenvolvido e utilizado neste trabalho. O uso do simulador facilita a verificação das técnicas sendo desenvolvidas pois é possível ter acesso as variáveis internas do sistema sendo simulado, o que muitas vezes não ocorre com os sistemas reais.

O simulador utilizado se baseia na modelagem fenomenológica de um robô móvel para uma versão anterior do time POTI de futebol de robôs, desenvolvida em Guerra (2005). Neste trabalho, foram considerados também a dinâmica dos atuadores além da cinemática do robô e da dinâmica das forças e velocidades envolvidas. Esse modelo é mais completo, mas recai no modelo (2.24) quando considera-se o robô simétrico e com

dinâmica dos motores desprezível, o que é aproximadamente verdade. O modelo utilizado no simulador será descrito a seguir de forma simplificada.

O modelo do simulador considera uma relação dinâmica entre as velocidades do robô e os sinais de entrada dado pela seguinte equação matricial:

$$K_v\mathbf{u} = M_v\dot{\mathbf{v}} + B_v\mathbf{v} \tag{3.6}$$

onde K_v é um ganho das entradas de controle e M_v e B_v são, respectivamente, a matriz de inércia generalizada e a matriz de amortecimento generalizada, que inclui termos de coeficientes de atrito viscoso e resistência elétrica, ambas consideradas constantes. Os vetores $\mathbf{u}$, $\mathbf{v}$ e $\dot{\mathbf{v}}$ são definidos como:

$$\mathbf{u} = \begin{bmatrix} u_1^\star \\ u_2^\star \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v \\ \omega \end{bmatrix} \qquad \dot{\mathbf{v}} = \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix}$$

O modelo cinemático do robô pode ser obtido por:

$$\dot{\mathbf{q}} = {}^{q}T_v(\theta)\mathbf{v} \tag{3.7}$$

onde:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \qquad {}^{q}T_v(\theta) = \begin{bmatrix} \cos(\theta) & 0 \\ \text{sen}(\theta) & 0 \\ 0 & 1 \end{bmatrix}$$

sendo que $\mathbf{q}$ representa a configuração do robô. A partir de (3.6) e (3.7) pode-se obter o modelo contínuo no espaço de estados do robô como:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}(\theta)\mathbf{x} + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{C}\mathbf{x} \end{cases} \tag{3.8}$$

sendo:

$$\mathbf{A}(\theta) = \begin{bmatrix} -\mathbf{M_v^{-1}B_v} & \vdots & \mathbf{0} \\ \cdots & \cdots & \cdots \\ \mathbf{^qT_v}(\theta) & \vdots & \mathbf{0} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{M_v^{-1}K_v} \\ \cdots \\ \mathbf{0} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{q} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{q} \end{bmatrix} = \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \end{bmatrix}$$

onde $\mathbf{0}$ define uma matriz nula.

A equação (3.8) é de fato o modelo que representa o sistema do robô móvel. Porém, é interessante gerar um simulador que também ofereça o deslocamento linear do robô como saída, a fim de utilizá-lo para a validação do algoritmo de estimação de l que será necessário para estimar os parâmetros do robô real. O modelo utilizado pelo simulador é então dado por:

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{l} \end{bmatrix} = \begin{bmatrix} -\mathbf{M_v^{-1}B_v} & \vdots\ \mathbf{0} \\ \cdots\cdots\cdots & \\ \cos(\theta) & 0 \\ \operatorname{sen}(\theta) & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \\ l \end{bmatrix} + \begin{bmatrix} \mathbf{M_v^{-1}K_v} \\ \cdots\cdots \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} u_1^\star \\ u_2^\star \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ \theta \\ l \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \\ x \\ y \\ \theta \\ l \end{bmatrix} + \begin{bmatrix} e_x \\ e_y \\ e_\theta \\ e_l \end{bmatrix} \tag{3.9}$$

onde e_x, e_y, e_θ, e e_l representam os ruídos de leitura da saída do sistema, e são considerados brancos, de média nula, e de com variâncias obtidas experimentalmente do sistema real.

Com o modelo (3.9) é possível simular a evolução da posição, orientação, e deslocamento linear do robô, podendo este último ser utilizado para validar o algoritmo de estimação de l. Apesar de ser contínuo, considera-se que a aplicação das entradas e a leitura dos dados de saída são realizadas no tempo $t = kT$, $k = 0, 1, 2, \ldots$ sendo T o período de amostragem do sistema e que a entrada permanece fixa durante esse período. Para calcular deslocamento percorrido entre duas amostras é feita uma integração do movimento do robô para deslocamentos infinitesimais durante o tempo de amostragem T. Os valores

de $\mathbf{K_v}$, $\mathbf{M_v}$ e $\mathbf{B_v}$ utilizados para simular o modelo são:

$$\mathbf{K_v} = \begin{bmatrix} 0.0468957 & 0.0457395 \\ 0.00175859 & -0.00171523 \end{bmatrix}$$
$$\mathbf{M_v} = \begin{bmatrix} 0.361424 & 4.08119 \times 10^{-6} \\ 4.08119 \times 10^{-6} & 2.6273 \times 10^{-5} \end{bmatrix}$$
$$\mathbf{B_v} = \begin{bmatrix} 0.196477 & 8.17498 \times 10^{-5} \\ 8.17498 \times 10^{-5} & 0.00020543 \end{bmatrix}$$

e foram calculados a partir de valores dos parâmetros físicos de um robô móvel. Tais parâmetros foram levantados através de medições e cálculos, e estão disponíveis em Guerra (2005). A zona-morta foi simulada aplicando os sinais de entrada no modelo (2.28) e utilizando as saídas para excitar o modelo (3.9).

3.4 Estimação do deslocamento linear

O problema em calcular o deslocamento linear reside em não se saber ao certo qual trajetória o robô executou em um período de amostragem. Ou seja, sendo:

$$l_k = l_{k-1} + \Delta l_k \tag{3.10}$$

o problema é definir o valor da distância percorrida Δl_k em um período de amostragem a partir de $(x_{k-1}, y_{k-1}, \theta_{k-1})$ e (x_k, y_k, θ_k), que são a posição e orientação do robô no início e no fim do movimento. Algumas abordagens podem ser utilizadas para resolver o problema, como mostra a figura 3.1.

Na figura 3.1 vemos que uma das opções é considerar que o robô executa uma trajetória linear entre duas configurações, o que matematicamente consiste em aproximar Δl_k pela distância euclidiana entre as posições (x_{k-1}, y_{k-1}) e (x_k, y_k). O problema desta abordagem é que ela viola as restrições não-holonômicas do sistema, ignorando tanto θ_{k-1} quanto θ_k. Uma outra opção, apresentada em Guerra (2005), é utilizar equações paramétricas de segundo grau para descrever a evolução de Δl durante um período de amostragem. É possível obter uma solução analítica através de uma integral de linha sobre tal equação paramétrica. Essa solução funciona para a maioria dos casos, mas também viola a restrição de movimento do robô para situações como a descrita na figura 3.1, onde θ_k seria desconsiderado.

Neste trabalho optou-se por definir a trajetória como uma curva de Bezier cúbica entre

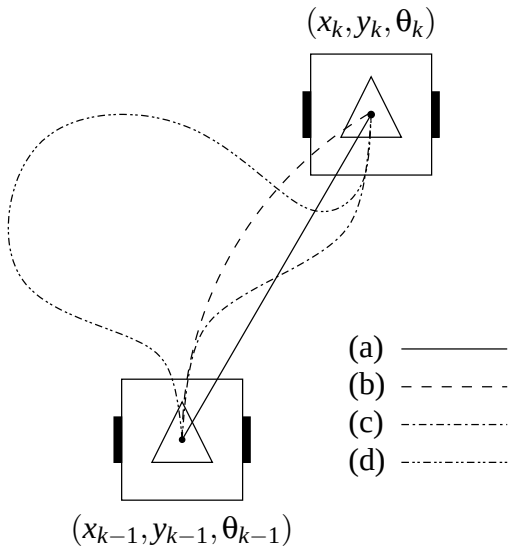

Figura 3.1: Possíveis aproximações para a trajetória do robô em um período de amostragem: (a) distância euclidiana; (b) curva de segundo grau; (c) curva de terceiro grau; (d) uma possível, porém improvável trajetória.

duas poses consecutivas do robô. Tal curva foi escolhida devido a sua simplicidade e por gerar uma trajetória que respeita as condições inicias e finais do movimento. O cálculo de Δl_k a partir da curva de Bezier foi dividido em duas partes: cálculo do módulo de Δl_k e cálculo do sinal de Δl_k, apresentadas nas seções 3.4.1 e 3.4.2. É importante notar que a estimação proposta ainda pode ser inapropriada em alguns casos, como a curva (d) da figura 3.1 ou trajetórias que seriam melhor descritas por curvas superiores às de terceira ordem, mas considera-se que o tempo de amostragem do sistema é suficientemente pequeno para garantir que tais situações não ocorram no sistema real.

3.4.1 Cálculo do módulo de Δl

O cálculo do módulo da distância percorrida pelo robô é feito através de uma integral de linha em uma curva de Bezier cúbica que define o movimento do robô em um período de amostragem. As curvas de Bezier são curvas paramétricas frequentemente utilizadas em computação gráfica, e são definidas através de um ponto inicial, um ponto final e pontos de controle. Podemos definir uma curva de Bezier $\mathbf{B}(\gamma)$ de ordem n como:

$$\mathbf{B}(\gamma) = \sum_{i=0}^{n} \binom{n}{i} (1-\gamma)^{n-i} \gamma^i \mathbf{P}_i \tag{3.11}$$

onde $0 \leq \gamma \leq 1$ é o parâmetro da equação. Para o caso específico de $n = 3$, temos:

$$\mathbf{B}(\gamma) = (1-\gamma)^3\mathbf{P}_0 + 3\gamma(1-\gamma)^2\mathbf{P}_1 + 3\gamma^2(1-\gamma)\mathbf{P}_2 + \gamma^3\mathbf{P}_3 \tag{3.12}$$

onde $\mathbf{P}_0$ e $\mathbf{P}_3$ são respectivamente o ponto inicial e final do movimento, dados por:

$$\mathbf{P}_0 = (x_{k-1}, y_{k-1}) \qquad\qquad \mathbf{P}_3 = (x_k, y_k)$$

e $\mathbf{P}_1$ e $\mathbf{P}_2$ são os pontos de controle da curva. O problema agora consiste em definir os pontos de controle de forma que a curva de Bezier melhor represente o movimento do robô durante um período de amostragem. Os pontos $\mathbf{P}_1, \mathbf{P}_2$ são calculados como mostrado na figura 3.2.

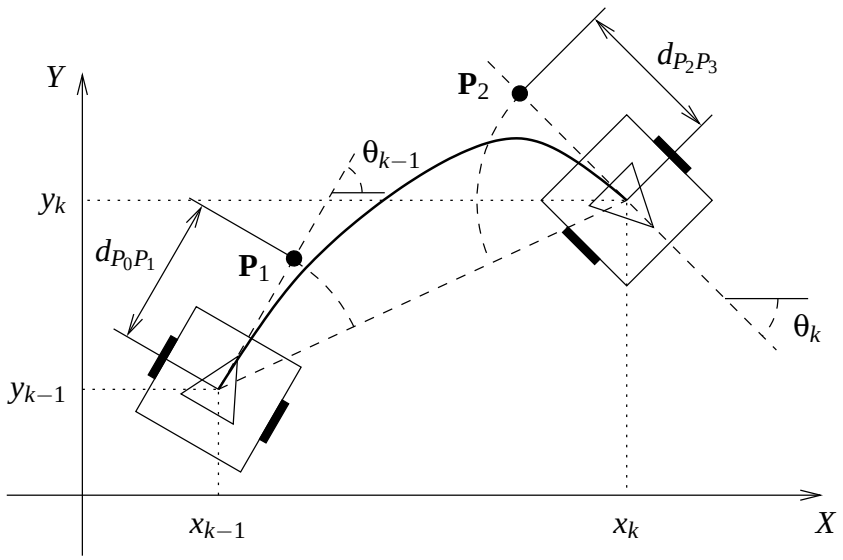

Figura 3.2: Cálculo dos pontos de controle da curva de Bezier para estimar Δl. Os pontos $\mathbf{P}_0$ e $\mathbf{P}_3$ são as posições de inicio e fim do movimento enquanto os pontos $\mathbf{P}_1$ e $\mathbf{P}_2$ são calculados em função de $\mathbf{P}_0$ e $\mathbf{P}_3$ e dos ângulos iniciais e finais do movimento

Como visto na figura 3.2, o ponto $\mathbf{P}_1$ é calculado para estar na reta passando pelo ponto $\mathbf{P}_0$ com a orientação θ_{k-1}, à uma distância $d_{P_0P_1}$ do ponto inicial no sentido do movimento, onde:

$$d_{P_0P_1} = \frac{\sqrt{(P_{3y} - P_{0y})^2 + (P_{3x} - P_{0x})^2}}{3} \tag{3.13}$$

sendo P_{ix} e P_{iy} as coordenadas x e y do ponto i. O ponto $\mathbf{P}_2$ é calculado para estar a uma distância $d_{P_2P_3} = d_{P_0P_1}$ do ponto $\mathbf{P}_3$ sobre a reta passando por $\mathbf{P}_3$ com orientação θ_k, no sentido inverso ao do movimento.

Com a equação (3.12) e os pontos P_i pode-se obter o módulo valor estimado de Δl:

$$\Delta\hat{l}_M = \int_0^1 \sqrt{\left(\frac{B_x}{d\gamma}\right)^2 + \left(\frac{B_y}{d\gamma}\right)^2}\, \mathrm{d}\gamma \tag{3.14}$$

através de métodos de integração numérica.

3.4.2 Cálculo do sinal de Δl

Obter o sinal de $\hat{\Delta l}$ significa saber se o robô moveu para frente ou para trás. Para isso, nós calculamos o valor de x_{proj}, que é a projeção da posição do robô no instante k em relação a um eixo fixado na posição $k-1$ e com a orientação θ_{k-1}, como indicado na figura 3.3. Se x_{proj} for positivo, o robô andou para a frente, e logo $\hat{\Delta l} > 0$. Caso contrário, o robô andou para trás e $\hat{\Delta l} < 0$. Uma simples transformação de coordenadas permite o cálculo de x_{proj}:

$$x_{\text{proj}} = x_k \cos\theta_{k-1} + y_k \text{sen}\theta_{k-1} - x_{k-1}\cos\theta_{k-1} - y_{k-1}\text{sen}\theta_{k-1} \tag{3.15}$$

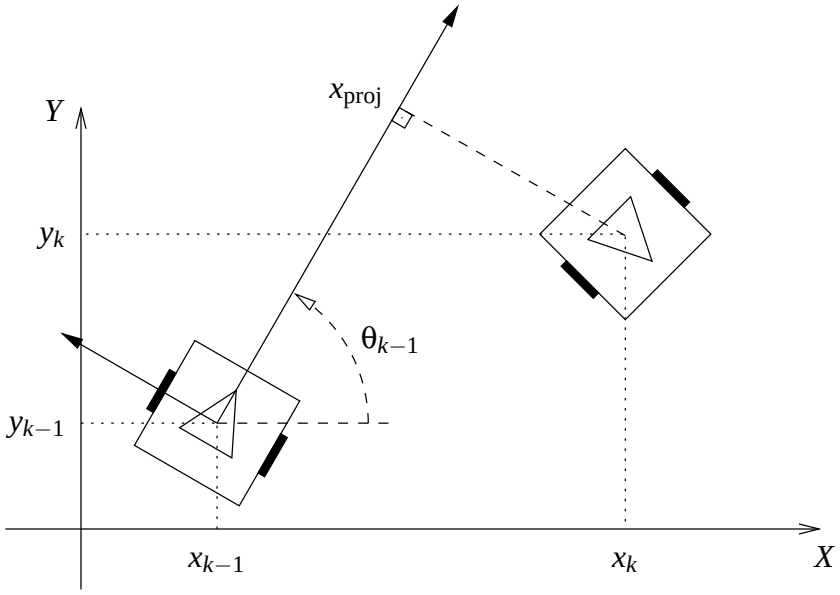

Figura 3.3: Esquema de cálculo do sinal de Δl: a posição final do robô é projetada na reta formada pela posição e angulo inicias do movimento. A estimativa de Δl vai ter o mesmo sinal do valor calculado para a projeção.

Finalmente, a partir das equações (3.14) e (3.15) pode-se calcular a estimativa de Δl_k por:

$$\Delta\hat{l}_k = \begin{cases} \Delta\hat{l}_M & \text{se } x_{\text{proj}} \geq 0 \\ -\Delta\hat{l}_M & \text{se } x_{\text{proj}} < 0 \end{cases} \tag{3.16}$$

e utilizando (3.10) pode-se definir a estimativa do deslocamento linear $\hat{l}_k$ como:

$$\hat{l}_k = \begin{cases} 0 & \text{se } k = 0 \\ \hat{l_{k-1}} + \Delta\hat{l}_k & \text{se } k > 0 \end{cases} \tag{3.17}$$

3.4.3 Validação do cálculo de $\Delta\hat{l}$

Para validar o algoritmo de cálculo de $\hat{l}$ é necessário validar o cálculo de $\Delta\hat{l}$. Para tal, foram obtidos valores de x, y e θ aplicando sinais de entrada aleatórios ao simulador descrito na seção 3.3. O valor de $\Delta\hat{l}_k$ foi então calculado utilizando (3.16). O valor de Δl foi calculado a partir da saída l do simulador, utilizando a equação (3.10). O resultado pode ser visto na figura 3.4. Percebe-se que o algoritmo estimou bem a variação de deslocamento linear, com desvio padrão de $\pm 1.43315 \times 10^{-5}$.

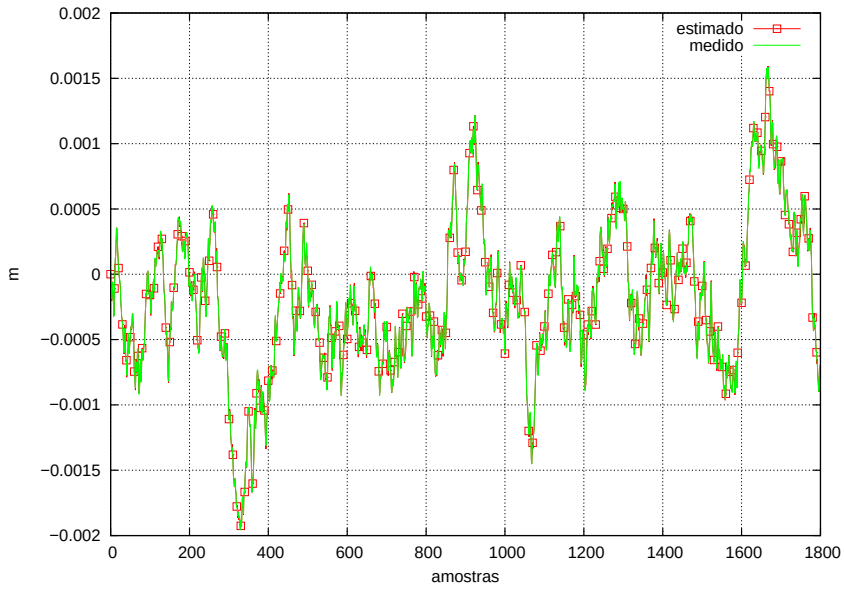

Figura 3.4: Comparação entre $\Delta\hat{l}$ e Δl necessária para validar o cálculo de $\hat{l}$. Observa-se que o valor de $\Delta\hat{l}$ representa bem o valor de Δl

Uma última observação importante é que para validar a estimação de l fizemos a comparação de $\Delta\hat{l}_k$ com Δl_k, sendo este último obtido de um sistema simulado. Essa

restrição se deve a impossibilidade de mensurar l no robô real.

3.5 Identificação dos parâmetros

Na seção 3.1 foram derivados os modelos (3.1) e (3.4) que descrevem o sistema do robô, sendo que o segundo considera as zonas-mortas. Para encontrar o valor dos parâmetros que permitam os modelos representar o sistema é necessário utilizar um algoritmo de estimação. Neste trabalho foi utilizado o algoritmo de mínimos quadrados devido a ser um dos mais conhecidos e mais utilizados nas mais diversas áreas de ciência e tecnologia [Aguirre 2007].

3.5.1 Identificação do modelo linear

Seja um modelo ARX do tipo:

$$y(k) = \psi^T(k-1)\hat{\boldsymbol{\theta}} + \xi(k) \tag{3.18}$$

onde $y(k)$ é a observação do sistema, $\hat{\boldsymbol{\theta}}$ é o vetor de parâmetros a serem estimados, $\xi(k)$ é o erro de estimação do modelo, $\psi(k-1)$ é um vetor de variáveis independentes (ou regressores), de ordem n_θ, na forma:

$$\begin{aligned}\psi(k-1) &= \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_{n_\theta} \end{bmatrix}^T \\ &= \begin{bmatrix} y(k-1) & \cdots & y(k-n_y) & u(k-1) & \cdots & u(k-n_u) \end{bmatrix}^T\end{aligned}$$

sendo n_y sendo o numero máximo de atraso entre os regressores de saída, n_x o número máximo de atraso entre os regressores de entrada, e k o instante considerado. O modelo (3.18) pode ser reescrito para uma massa de dados como:

$$\boldsymbol{y} = \Psi\hat{\boldsymbol{\theta}} + \boldsymbol{\xi} \tag{3.19}$$

sendo:

$$\begin{aligned}\boldsymbol{y} &= \begin{bmatrix} y(k) & \cdots & y(k-N) \end{bmatrix}^T \\ \Psi &= \begin{bmatrix} \psi(k-1) & \cdots & \psi(k-N) \end{bmatrix}^T \\ \boldsymbol{\xi} &= \begin{bmatrix} \xi(k) & \cdots & \xi(k-N) \end{bmatrix}^T\end{aligned}$$

e N sendo o número de amostras. O algoritmo de mínimos quadrados garante encontrar um vetor de parâmetros estimados que minimiza o valor da função:

$$J_{MQ}(\hat{\boldsymbol{\theta}}) = \boldsymbol{\xi}^T \boldsymbol{\xi} = \| \boldsymbol{\xi} \|^2 \tag{3.20}$$

e esse vetor sendo dado por:

$$\hat{\boldsymbol{\theta}}_{MQ} = \left[\Psi^T \Psi \right]^{-1} \Psi^T \boldsymbol{y} \tag{3.21}$$

O modelo (3.1) é MIMO, porém iremos identificá-lo como dois sistemas MISO, cada um composto por uma equação de (3.1). Podemos escrever então:

$$\begin{gathered}
\hat{\theta}^L = \begin{bmatrix} a_1^L & a_2^L & b_1^L & b_2^L \end{bmatrix}^T \\
\boldsymbol{y}^L = \begin{bmatrix} \hat{l}_k & \hat{l_{k-1}} & \cdots & \hat{l_{k-N}} \end{bmatrix}^T \\
\Psi^L = \begin{bmatrix} \hat{l_{k-1}} & \hat{l_{k-2}} & u_{1,k-1-d} & u_{2,k-1-d} \\ \hat{l_{k-2}} & \hat{l_{k-2}} & u_{1,k-1-d} & u_{2,k-1-d} \\ \vdots & \vdots & \vdots & \vdots \\ \hat{l_{k-1-N}} & \hat{l_{k-2-N}} & u_{1,k-1-d-N} & u_{2,k-1-d-N} \end{bmatrix}^T
\end{gathered} \tag{3.22}$$

$$\begin{gathered}
\hat{\boldsymbol{\theta}}^\Theta = \begin{bmatrix} a_1^\Theta & a_2^\Theta & b_1^\Theta & b_2^\Theta \end{bmatrix}^T \\
\boldsymbol{y}^\Theta = \begin{bmatrix} \theta_k & \theta_{k-1} & \cdots & \theta_{k-N} \end{bmatrix}^T \\
\Psi^\Theta = \begin{bmatrix} \theta_{k-1} & \theta_{k-2} & u_{1,k-1-d} & u_{2,k-1-d} \\ \theta_{k-2} & \theta_{k-2} & u_{1,k-1-d} & u_{2,k-1-d} \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{k-1-N} & \theta_{k-2-N} & u_{1,k-1-d-N} & u_{2,k-1-d-N} \end{bmatrix}^T
\end{gathered} \tag{3.23}$$

onde os valores de $\hat{\boldsymbol{\theta}}^L$ e $\hat{\boldsymbol{\theta}}^\Theta$ podem ser estimados por:

$$\hat{\boldsymbol{\theta}}^L = \left[\Psi^{L\,T} \Psi^L \right]^{-1} \Psi^{L\,T} \boldsymbol{y}^L \tag{3.24}$$

$$\hat{\boldsymbol{\theta}}^\Theta = \left[\Psi^{\Theta\,T} \Psi^\Theta \right]^{-1} \Psi^{\Theta\,T} \boldsymbol{y}^\Theta \tag{3.25}$$

3.5.2 Identificação do modelo com zona-morta

Para identificar o modelo com zona-morta, vamos reescrever as equações em (3.4) no formato matricial como:

$$\begin{aligned} l_k &= \psi_{k-1}^L \boldsymbol{\theta}^L \\ \theta_k &= \psi_{k-1}^\Theta \boldsymbol{\theta}^\Theta \end{aligned} \tag{3.26}$$

onde:

$$\boldsymbol{\theta}^L = \begin{bmatrix} a_1^L & a_2^L & b_1^L & b_2^L & b_3^L & b_4^L & e_1^L & e_2^L \end{bmatrix}^T$$

$$\boldsymbol{\theta}^\Theta = \begin{bmatrix} a_1^\Theta & a_2^\Theta & b_1^\Theta & b_2^\Theta & b_3^\Theta & b_4^\Theta & e_1^\Theta & e_2^\Theta \end{bmatrix}^T$$

$$\psi_{k-1}^L = \begin{bmatrix} l_{k-1} \\ l_{k-2} \\ u_{1,k-1-d}\left(d(u_{1,k-1-d}-\eta_1)+d(-u_{1,k-1-d}-\eta_1)\right) \\ u_{2,k-1-d}\left(d(u_{2,k-1-d}-\eta_2)+d(-u_{2,k-1-d}-\eta_2)\right) \\ u_{1,k-2-d}^\star \\ u_{2,k-2-d}^\star \\ \left(-d(u_{1,k-1-d}-\eta_1)+d(-u_{1,k-1-d}-\eta_1)\right) \\ \left(-d(u_{2,k-1-d}-\eta_2)+d(-u_{2,k-1-d}-\eta_2)\right) \end{bmatrix}^T$$

$$\psi_{k-1}^\Theta = \begin{bmatrix} \theta_{k-1} \\ \theta_{k-2} \\ u_{1,k-1-d}\left(d(u_{1,k-1-d}-\eta_1)+d(-u_{1,k-1-d}-\eta_1)\right) \\ u_{2,k-1-d}\left(d(u_{2,k-1-d}-\eta_2)+d(-u_{2,k-1-d}-\eta_2)\right) \\ u_{1,k-2-d}^\star \\ u_{2,k-2-d}^\star \\ \left(-d(u_{1,k-1-d}-\eta_1)+d(-u_{1,k-1-d}-\eta_1)\right) \\ \left(-d(u_{2,k-1-d}-\eta_2)+d(-u_{2,k-1-d}-\eta_2)\right) \end{bmatrix}^T$$

Analisando (3.26) nota-se que os vetores regressores ψ_{k-1}^L e ψ_{k-1}^Θ contém termos $u_i^\star$, que dependem de η_i pela equação (2.28), além da dependência direta de η_i em outros termos. O problema é que os termos η_i fazem parte dos parâmetros do modelo que estão sendo identificados utilizando os vetores regressores. Tal configuração sugere a utilização de um algoritmo recursivo de estimação. Com a estimação recursiva, os valores de η_i podem ser calculados através de (3.5) a partir dos parâmetros estimados no passo anterior, e utilizados para atualizar ψ_{k-1}^L e ψ_{k-1}^Θ.

Deve-se atentar ao fato de existirem duas formas de calcular η_i em (3.5). Isso mostra que, pelo menos matematicamente, podem existir limites diferentes da zona-morta dependendo da saída considerada. Para contornar isso, vamos definir η_i como sendo:

$$\eta_i = \frac{\eta_i^L + \eta_i^\Theta}{2} \tag{3.27}$$

onde:

$$\eta_1^L = \frac{e_1^L}{b_1^L} \qquad \eta_2^L = \frac{e_2^L}{b_2^L} \qquad \eta_1^\Theta = \frac{e_1^\Theta}{b_1^\Theta} \qquad \eta_2^\Theta = \frac{e_2^\Theta}{b_2^\Theta} \tag{3.28}$$

e que podem ser calculados a partir dos parâmetros estimados a cada iteração do algoritmo de estimação.

3.6 Resultados simulados

Nessa sessão serão apresentados alguns resultados obtidos a partir de dados do robô simulado. Em cada experimento foi simulado o equivalente a 60 segundos de movimento do robô real, o que é equivalente a 1800 amostras. Nesses experimentos foi considerado que o robô não possui tempo de atraso, e as zonas mortas utilizadas foram $\eta_1 = 0.12$ e $\eta_2 = 0.08$.

Inicialmente vamos identificar o modelo sem zona-morta. Para tal, foram utilizados sinais de excitação do tipo PRBS (do inglês *pseudo-random binary signal*), com amplitude de 70% do valor máximo de entrada. O sinal PRBS foi escolhido por sempre excitar a planta com valores de amplitude acima do limite da zona-morta, evitando assim que os efeitos dessa não-linearidade influenciem na estimação dos parâmetros. Em outras palavras, estamos identificando um modelo sem zona-morta para um robô simulado funcionando em um ponto de operação linear. Essa é a ideia utilizada nas identificações de sistemas reais onde os modelos não englobam as parcelas não-lineares. O sinal utilizado na identificação é mostrado na figura 3.5. A saída do sistema é mostrada na figura 3.6, assim como o $\hat{l}$ estimado a partir das posições e orientações do robô. Os parâmetros estimados são mostrados na tabela 3.1.

Para validar o modelo é necessário obter um novo conjunto de dados e realizar a predição das saídas utilizando o modelo com os parâmetros estimados. De forma geral, a maioria dos modelos apresentam bons resultados para predições um passo à frente [Aguirre 2007], sendo então esse método pouco eficiente para verificar o modelo obtido. Neste trabalho, foi utilizada uma predição de 1 segundo à frente, que equivale a 33 passos. A

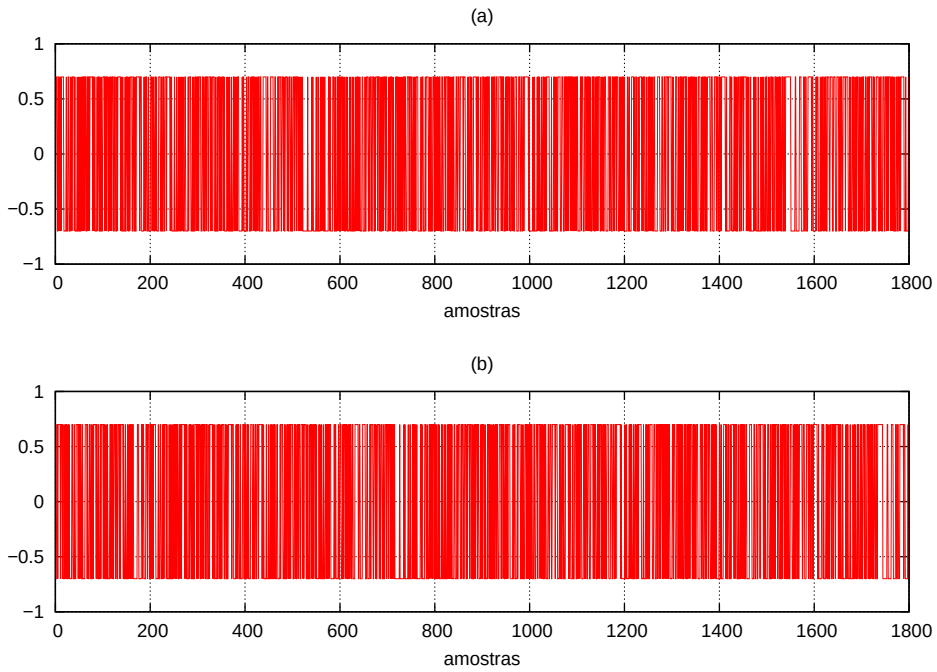

Figura 3.5: Sinais PRBS aplicados na entrada direita (a) e entrada esquerda (b) do robô simulado.

Parâmetro	Valor estimado médio	Variância
a_1^L	1.45084	3.58233×10^{-3}
a_2^L	-0.450695	3.59503×10^{-3}
b_1^L	6.67985×10^{-3}	4.67049×10^{-6}
b_2^L	9.93478×10^{-3}	4.99907×10^{-6}
b_3^L	8.38995×10^{-3}	2.56265×10^{-4}
b_4^L	1.25043×10^{-2}	2.43057×10^{-4}
a_1^Θ	1.67179	6.97902×10^{-3}
a_2^Θ	-0.672286	6.38176×10^{-3}
b_1^Θ	0.0278656	1.40317×10^{-5}
b_2^Θ	0.0286019	4.92724×10^{-6}
b_3^Θ	-0.0294452	1.4681×10^{-5}
b_4^Θ	-0.0301466	1.05109×10^{-5}

Tabela 3.1: Resultado de 100 estimações dos parâmetros do modelo sem zona-morta com dados obtidos do robô simulado.

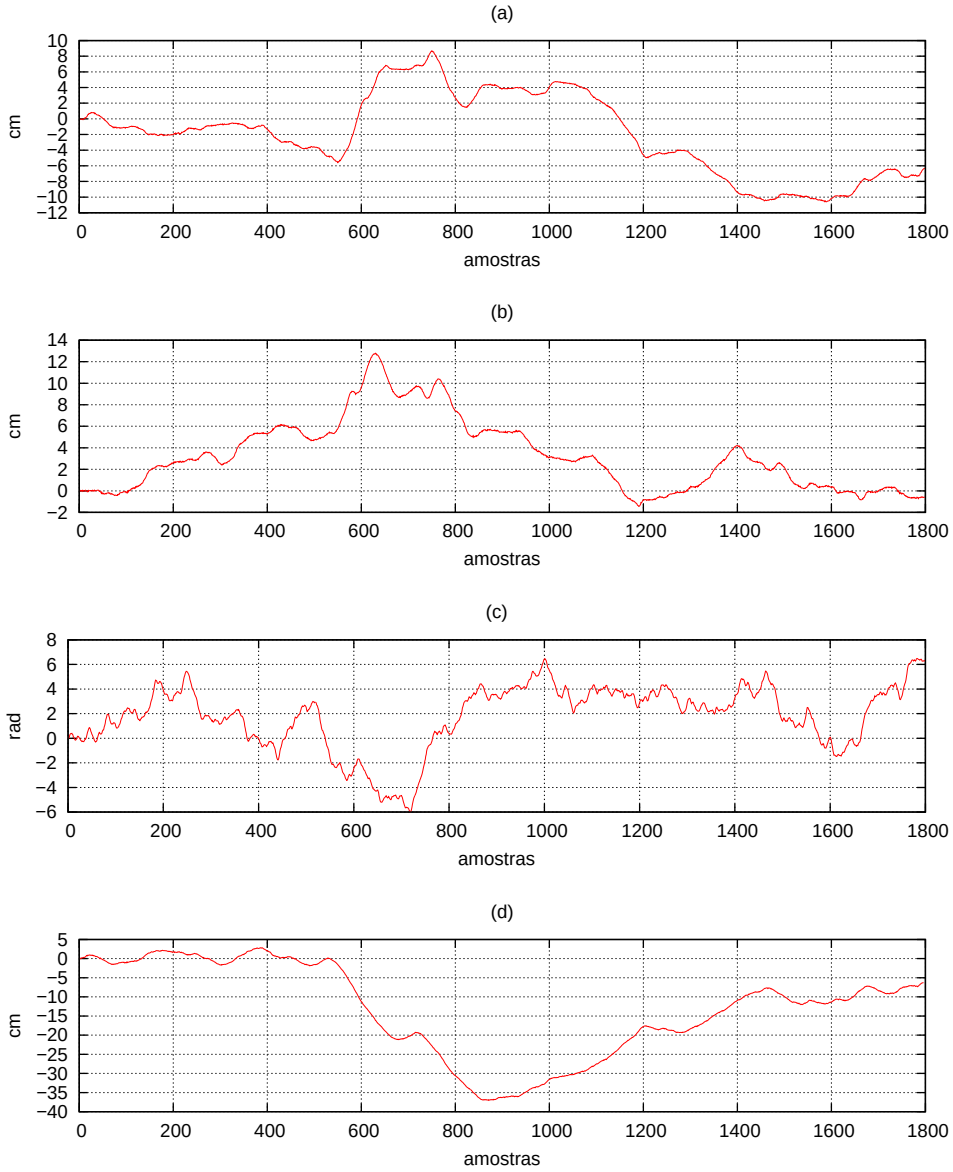

Figura 3.6: Resposta do robô simulado às entradas PRBS mostradas na figura 3.5: (a) x, (b) y, (c) θ e (d) $\hat{l}$.

figura 3.7 mostra o resultado da validação. Uma forma de mensurar o quão bom é um modelo é calcular o valor RMSE (do inglês *root mean square error*):

$$\text{RMSE} = \frac{\sqrt{\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2}}{\sqrt{\sum_{k=1}^{N}(y(k)-\bar{y})^2}} \tag{3.29}$$

onde y é a saída medida, $\hat{y}$ é a estimativa para a saída e $\bar{y}$ é o valor médio do sinal. Para essa identificação, os valores RMSE foram 0.227757 para a saída l e 0.201625 para θ.

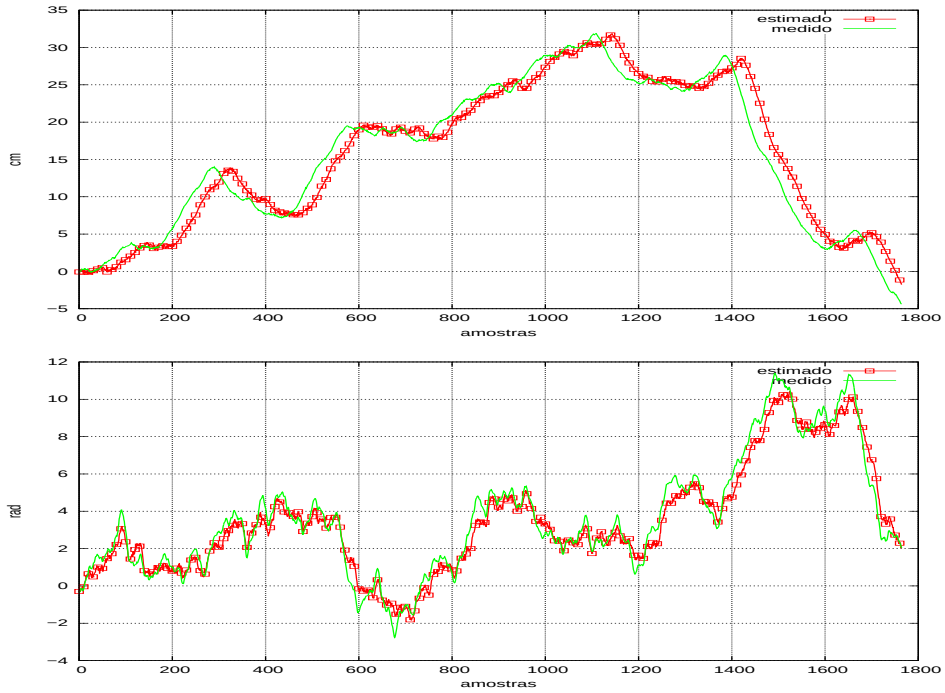

Figura 3.7: Validação do modelo sem zona-morta para o robô simulado, sendo a evolução de l ao topo e a evolução de θ abaixo.

A identificação do modelo com zona-morta se deu através do algoritmo de mínimos quadrados recursivo. A cada iteração, os valores da zona-morta foram estimados e utilizados juntamente das medições e sinais de entrada anteriores para montar o próximo vetor de regressores. Antes de executar o algoritmo, é necessário definir um limite $\eta_i^{\max}$, que é o máximo que o valor η_i pode assumir durante a recursão, de forma que $\eta_i^{\max} < u_i^{\max}$, que é o valor máximo que os sinais de entrada podem assumir. Isso é necessário pois se $\eta_i > u_i^{\max}$ durante a evolução dos parâmetros na identificação recursiva, temos por (2.28) que $u_i^{\star}$ será zero, independente do valor de u_i, ou seja, os sinais de entrada não serão mais considerados. Para esta identificação considera-se $\eta_i^{\max} = 0.5$ como o valor para esse

limite.

Em uma identificação de parâmetros, só é possível estimar algum efeito do sistema se esse efeito aparecer nos dados sendo utilizados [Aguirre 2007]. Ou seja, é impossível estimar os limites da zona-morta aplicando um sinal PRBS de amplitude maior que η_i, pois as medições do sistema não refletem o efeito da não-linearidade. Dessa forma, foram utilizados sinais de entrada gaussianos de média nula, e com desvio padrão de ± 0.3, garantindo que algumas entradas estejam dentro dos limites da zona-morta e alguns estejam fora desses limites. A figura 3.8 mostra o sinal utilizado enquanto a figura 3.9 mostra os dados medidos do robô simulado, assim como l estimado a partir desses dados. As figuras 3.10 à 3.15 mostram a evolução da estimação dos parâmetros do modelo e a tabela 3.2 mostra os valores finais estimados.

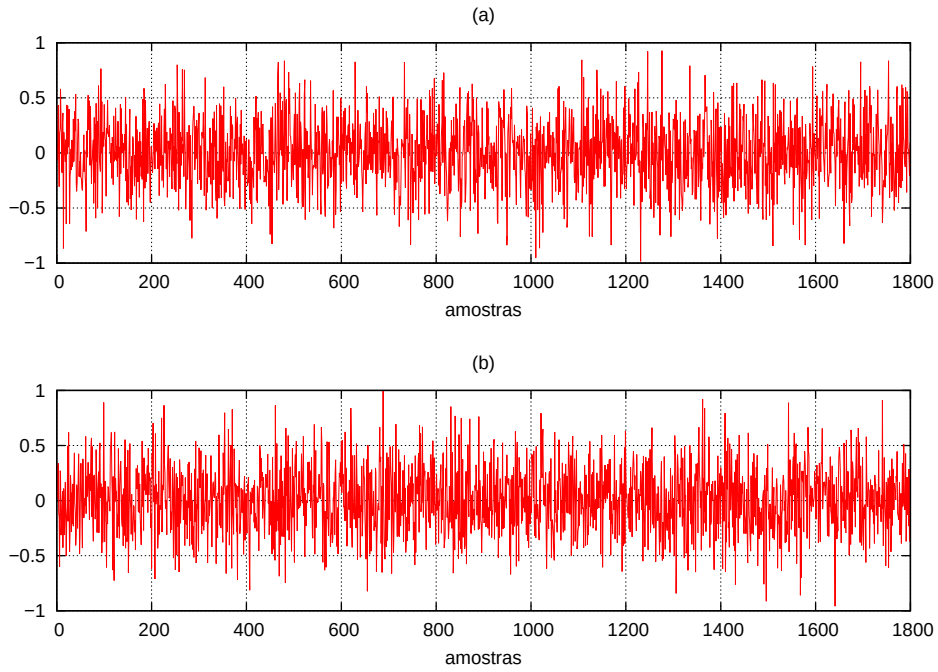

Figura 3.8: Sinais Gaussianos aplicados na entrada direita (a) e entrada esquerda (b) do robô simulado. Esses sinais são melhores para identificar o modelo com zona-morta.

O modelo obtido foi validado excitando-o com novos sinais semelhantes aos mostrados na figura 3.8 e comparados com as medições do sistema simulado ao serem aplicadas estas mesmas entradas. O resultado pode ser visto na figura 3.17. Os valores de RMSE obtidos nas estimações das saídas foram 0.109892 para l e 0.0213132 para θ.

Para finalizar os experimentos simulados, foi feita a validação do modelo sem zona-morta utilizando o sinal da figura 3.8, ou seja, que exita o sistema na não-linearidade. O

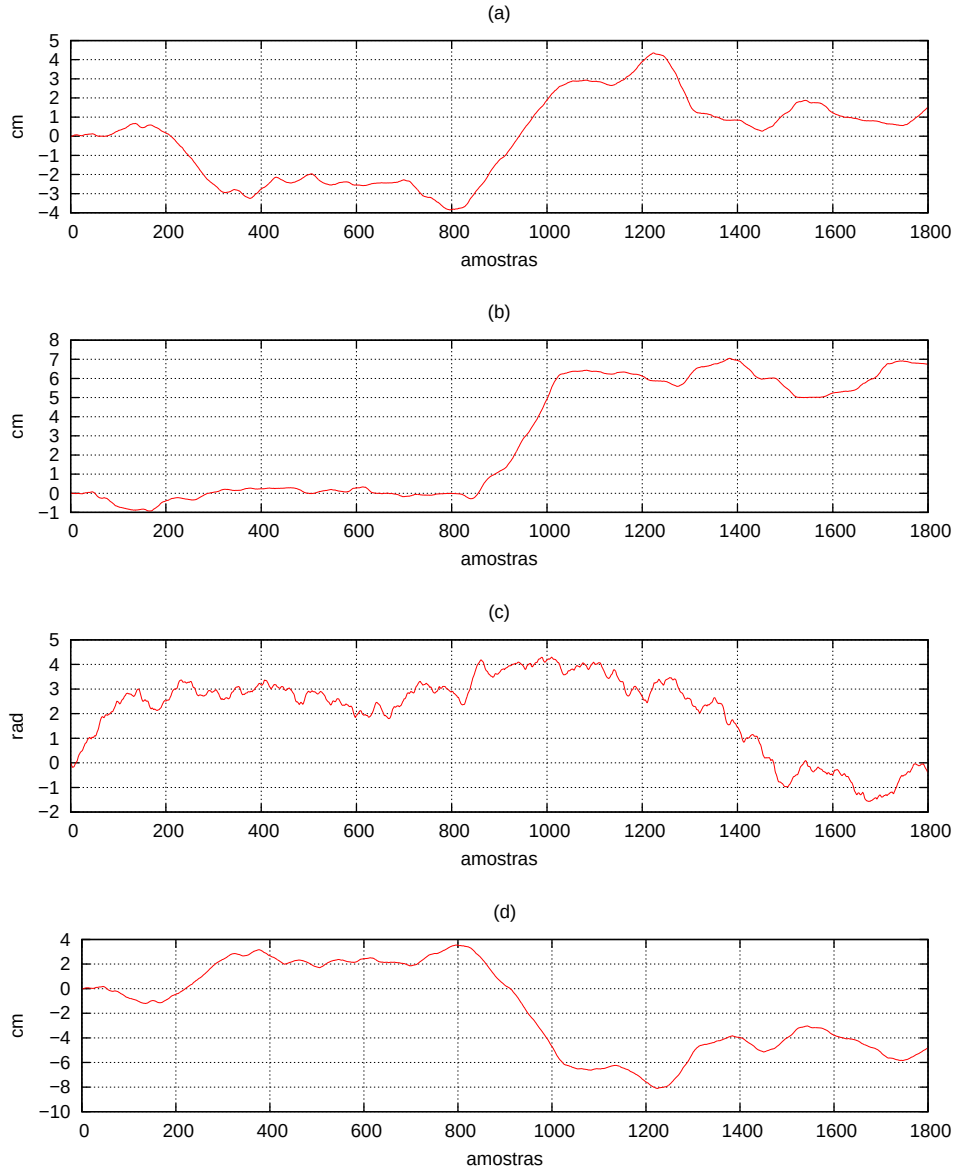

Figura 3.9: Resposta do robô simulado às entradas gaussianas mostradas na figura 3.8: (a) x, (b) y, (c) θ e (d) $\hat{l}$.

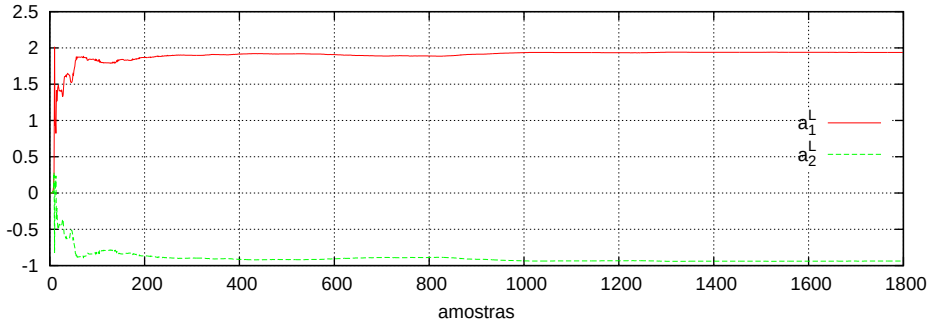

Figura 3.10: Evolução da estimação dos parâmetros a_1^L e a_2^L do modelo com zona-morta.

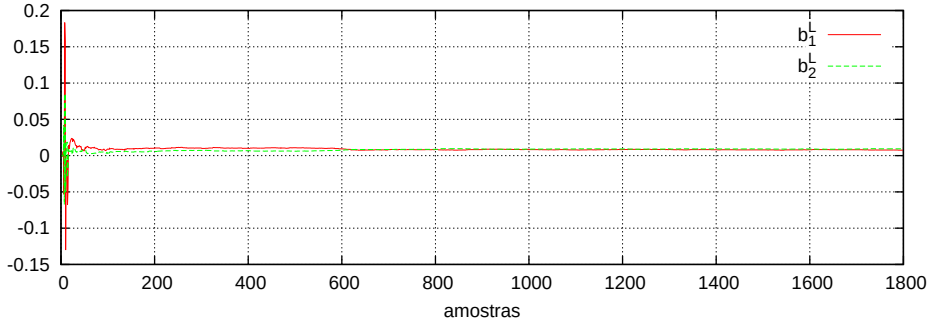

Figura 3.11: Evolução da estimação dos parâmetros b_1^L e b_2^L do modelo com zona-morta.

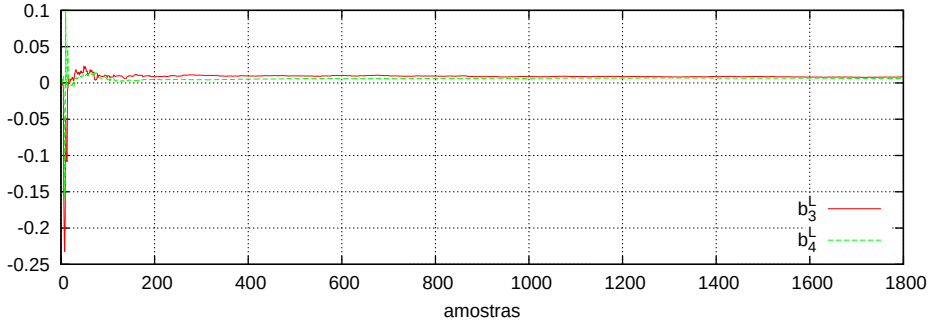

Figura 3.12: Evolução da estimação dos parâmetros b_3^L e b_4^L do modelo com zona-morta.

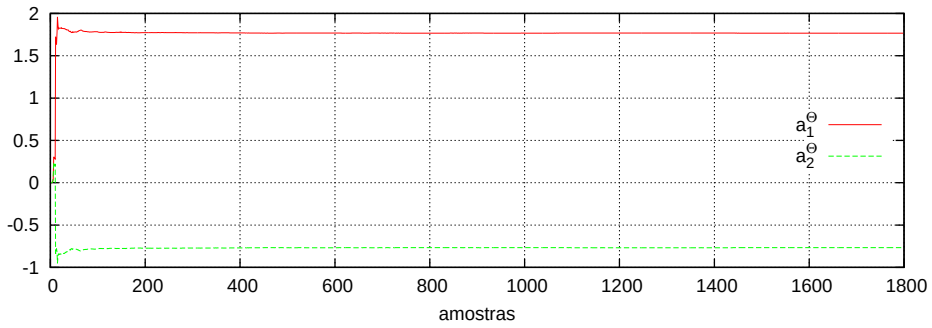

Figura 3.13: Evolução da estimação dos parâmetros a_1^Θ e a_2^Θ do modelo com zona-morta.

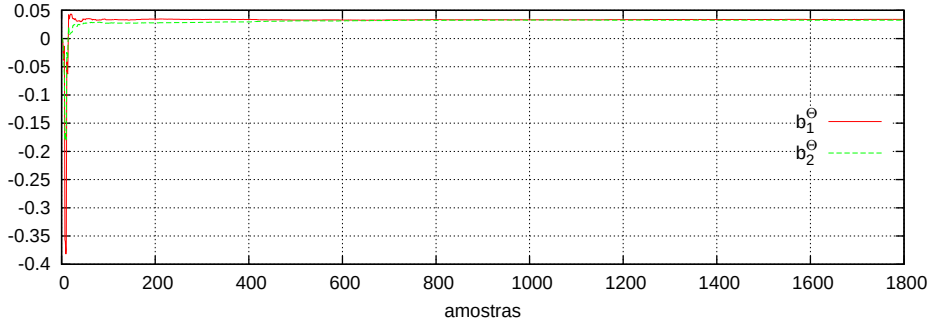

Figura 3.14: Evolução da estimação dos parâmetros b_1^Θ e b_2^Θ do modelo com zona-morta.

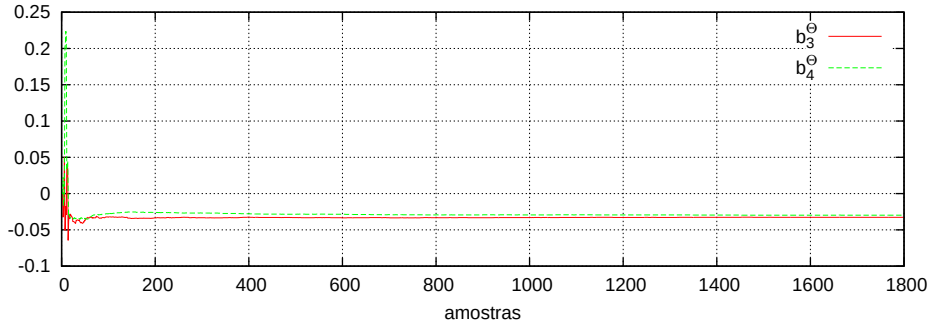

Figura 3.15: Evolução da estimação dos parâmetros b_3^Θ e b_4^Θ do modelo com zona-morta.

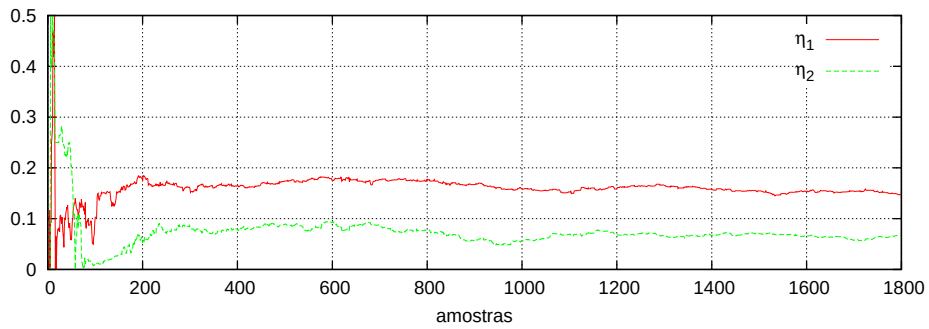

Figura 3.16: Evolução da estimação dos parâmetros η_1 e η_2 do modelo com zona-morta.

Parâmetro	Valor estimado médio	Variância
a_1^L	1.92722	1.9783×10^{-4}
a_2^L	-0.927181	198112×10^{-4}
b_1^L	0.00731866	2.38951×10^{-6}
b_2^L	0.00752753	2.0049×10^{-6}
b_3^L	0.00732369	2.15182×10^{-6}
b_4^L	0.0078648	1.56632×10^{-6}
a_1^Θ	1.76654	8.89917×10^{-6}
a_2^Θ	-0.766542	8.59039×10^{-4}
b_1^Θ	0.0341052	5.7082×10^{-7}
b_2^Θ	0.0319905	3.19816×10^{-6}
b_3^Θ	-0.0335148	3.35361×10^{-7}
b_4^Θ	-0.0316199	1.82281×10^{-6}
η_1	0.118022	0.767342×10^{-4}
η_2	0.0831863	0.730115×10^{-4}

Tabela 3.2: Resultado de 100 estimações dos parâmetros do modelo com zona-morta a partir de dados obtidos do robô simulado.

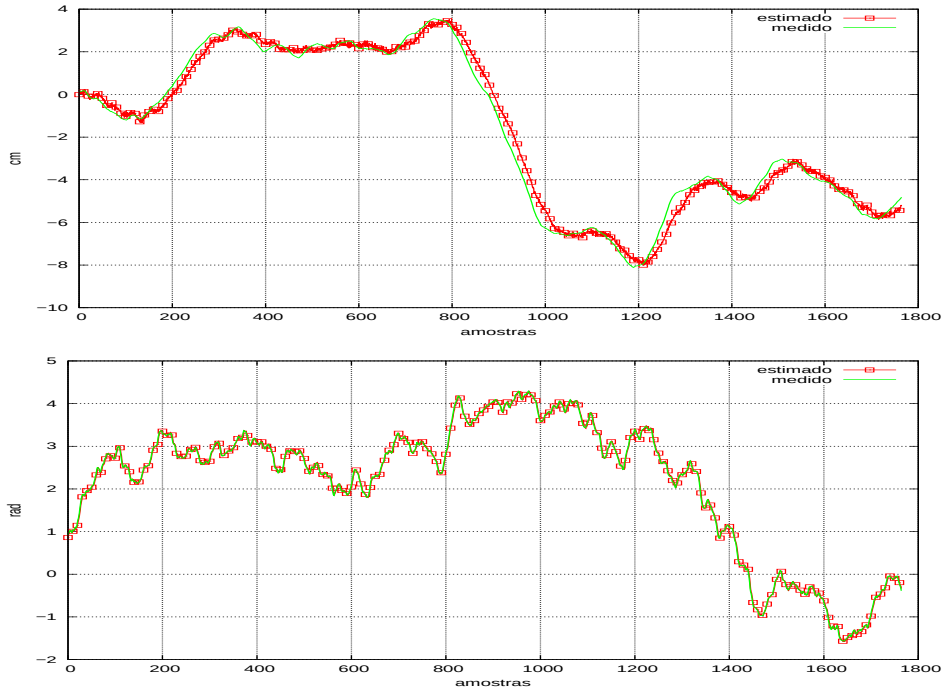

Figura 3.17: Validação do modelo com zona-morta para o robô simulado, sendo a evolução de l ao topo e a evolução de θ a baixo.

resultado pode ser visto na figura 3.18. Para esse experimento, foram calculados valores RMSE de 0.20228 para l e 0.123293 para θ, que são maiores do que os obtidos com o modelo com zona-morta.

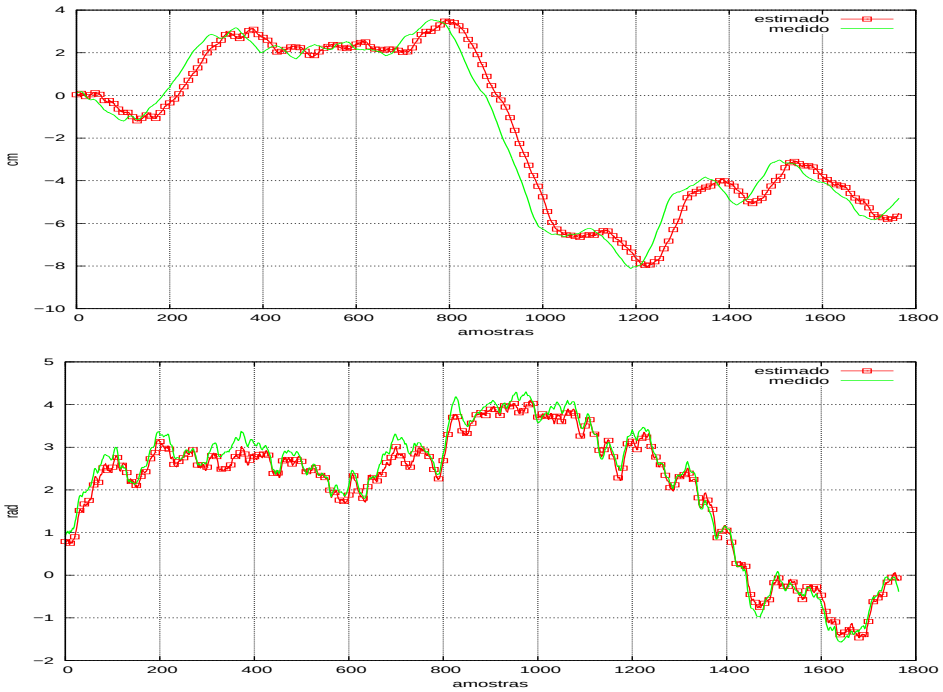

Figura 3.18: Predição usando modelo sem zona-morta e entradas gaussianas de baixa amplitude, sendo a evolução de l ao topo e a evolução de θ a baixo.

Capítulo 4

Resultados experimentais

No capítulo 3 foi mostrado como proceder para estimar os parâmetros do modelo sem e com zona-morta. Porém, as identificações apresentadas até então foram realizadas com dados de um robô simulado. Neste capítulo serão apresentados os resultados obtidos ao se identificar o modelo do sistema real. Primeiramente será feita uma descrição dos elementos que compõem o sistema assim como algumas restrições impostas do ambiente de testes que devem ser consideradas durante a estimação. Em seguida, serão mostrados o resultado da estimação dos parâmetros do modelo do robô real.

4.1 Descrição do sistema

Nesta seção será feita uma descrição da estrutura utilizada para a obtenção dos dados do robô real. O robô a ser identificado, que pode ser visto na figura 4.1, faz parte de um time de futebol de robôs da categoria *IEEE Very-Small*, que é semelhante à *FIRA Mirosot*. Nessa categoria, dois times formados por até três robôs são colocados em um campo com dimensões de 1.5×1.3m e devem disputar uma partida de futebol de maneira autônoma, dividida em dois tempos de cinco minutos. Cada time é controlado por um computador, que obtém a posição dos robôs e da bola no campo através de um sistema de visão global e se comunica com os robôs via transmissão de rádio, informando qual o sinal a ser aplicado nos atuadores do robô que resulte no movimento desejado de acordo com a estratégia da equipe. Cada robô é projetado de forma a caber dentro de um cubo de $7.5cm$ de aresta, o que limita consideravelmente o numero de componentes que podem ser embarcados. Um exemplo da estrutura descrita pode ser vista na figura 4.2. A seguir será feita uma descrição um pouco mais detalhada dos elementos da estrutura que são mais relevantes para a realização deste trabalho.

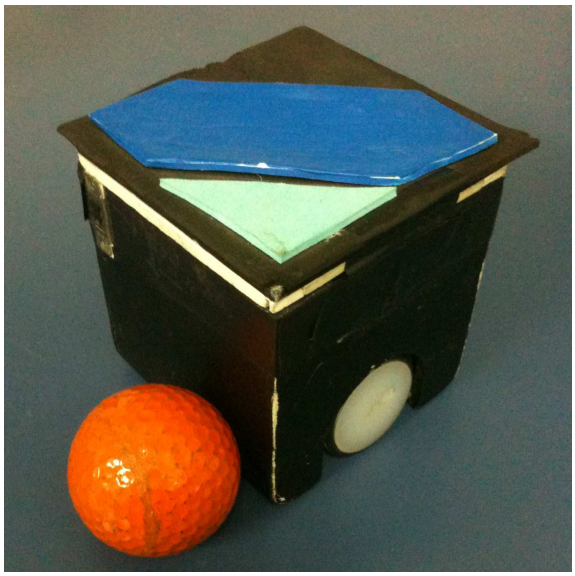

Figura 4.1: O robô utilizado neste trabalho

4.1.1 Sistema de visão e localização

O sistema de visão consiste em uma câmera e um algoritmo de localização. A câmera é posicionada sobre o campo a uma altura de onde seja possível ver todo o campo, como pode ser visto na figura 4.2. O algoritmo de localização calcula a posição em metros dos robôs e da bola em relação a um referencial fixado no centro do campo, com o eixo das ordenadas coincidindo com a linha central, a partir das imagens obtidas da câmera.

A câmera possui uma taxa de amostragem fixa de 30 quadros por segundo, o que fixa o período de amostragem do sistema em:

$$T = \frac{1}{30} = 0.0333\text{s} = 33.3\text{ms}$$

O tempo gasto pelo pelo sistema de visão ao processar uma nova imagem e extrair as posições é aproximadamente 3.3ms e o tempo gasto pelo restante do algoritmo até a transmissão dos dados no robô é 0.25ms, resultando em um tempo total de 3.55ms, que é menor do que T.

Cada robô tem na parte superior uma marca colorida que é utilizada para identificar a posição e orientação do mesmo. O formato da marca é predefinido por cada equipe. A marca utilizada para localizar o robô nesse trabalho pode ser vista em 4.3. Ela consiste em uma marca principal de onde é extraída a posição em pixeis (x_p, y_p) do robô, e uma marca auxiliar que é utilizada para definir a orientação.

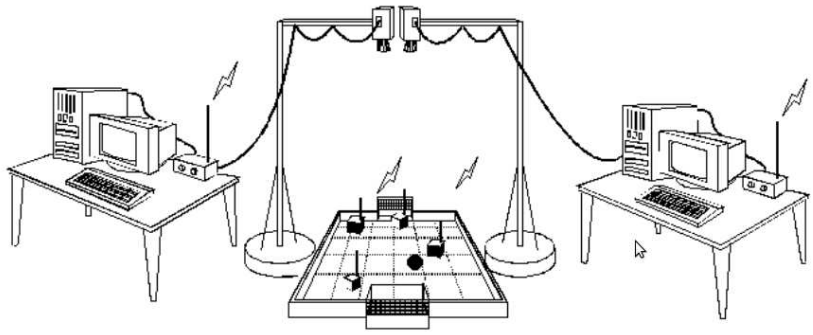

Figura 4.2: Estrutura utilizada na categoria *IEEE Very Small* de futebol de robôs, que foi utilizada para obter os dados do robô real. Imagem extraída de *Rules for the IEEE Very Small Competition* (2008)

A posição (x_p, y_p) é corrigida por um algoritmo que elimina a distorção radial e convertida para (x, y) em metros utilizando uma técnica de homografia que pode ser vista com mais detalhes em Mendes & Medeiros (2008). A figura 4.4 mostra a posição e orientação obtidas pela visão durante 5 segundos para um robô parado em $(x, y) \approx (0.60, 0.35)$ com $\theta \approx 0$rad. O erro na localização de cada variável tem média nula e os seguintes desvios padrões:

$$\begin{aligned} \sigma_x &= 2.99243 \times 10^{-4} \\ \sigma_y &= 3.41511 \times 10^{-4} \\ \sigma_\theta &= 1.44964 \times 10^{-2} \end{aligned} \tag{4.1}$$

sendo (4.1) utilizada para definir os ruídos da equação 3.9 do simulador desenvolvido na seção 3.3. Para a saída l do simulador, foi considerado que o erro e_l tem uma variância $\sigma_l = \sigma_x = \sigma_y$.

4.1.2 Transmissão e tempo de atraso

A cada passo o computador envia u_1 e u_2 para o robô via um transmissor de rádio que trabalha na faixa ISM de 2.41GHz, onde $-1 \leq u_i \leq 1$. Na prática, $|u_i|$ representa a porcentagem do valor máximo de tensão permitido nos terminais dos motores e o sinal de u_i indica o sentido de giro, de forma que:

$$u_i > 0 \implies v_i > 0$$

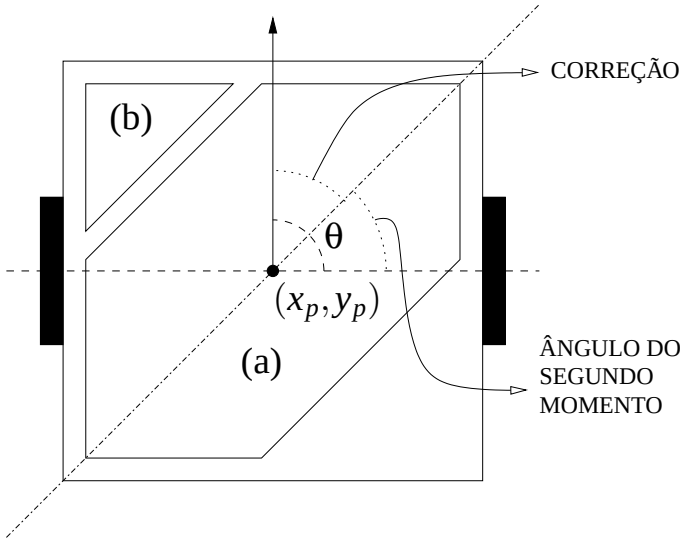

Figura 4.3: Marca utilizada pelo sistema de visão para localizar o robô. A média das coordenadas e dos pixeis da região principal (a) define a posição (x_p, y_p) do robô em pixeis. O segundo momento de área fornece a orientação da região principal que é corrigida para a orientação do robô pela região secundária (b), posicionada sempre na região frontal esquerda do robô.

Além disso, o sistema tem um atraso de tempo $d = 3$ amostras, obtido experimentalmente da resposta do robô a entradas do tipo degrau, como mostrado na figura 4.5.

4.1.3 Limites do espaço de trabalho

O espaço de trabalho é delimitado pelos limites do campo. Para estimar corretamente os parâmetros do modelo é necessário garantir que o robô não colidiu com as bordas do campo durante a coleta de dados. Uma forma de reduzir o numero de colisões é iniciar o movimento do robô no centro do campo e excitá-lo com sinais persistentemente excitantes de média zero e com suas amplitudes máximas limitadas a um valor abaixo do valor máximo. Apesar de não evitar colisões, tais sinais tendem a deixar o robô mais próximo da sua posição inicial por um tempo suficiente para coletar os dados. Para uma identificação recursiva tal técnica não é suficiente sendo necessário uma forma de detectar as colisões e parar a atualização dos parâmetros do modelo durante esse tempo. Uma forma de resolver esse problema é monitorar em tempo real a variância dos parâmetros sendo estimados e impedir a atualização dos parâmetros caso a variância ultrapasse um determinado limite e retornar a atualização assim que o valor estiver novamente abaixo deste limite [Pereira 2000]. A utilização dessa técnica só é possível se o algoritmo recursivo for inicializado

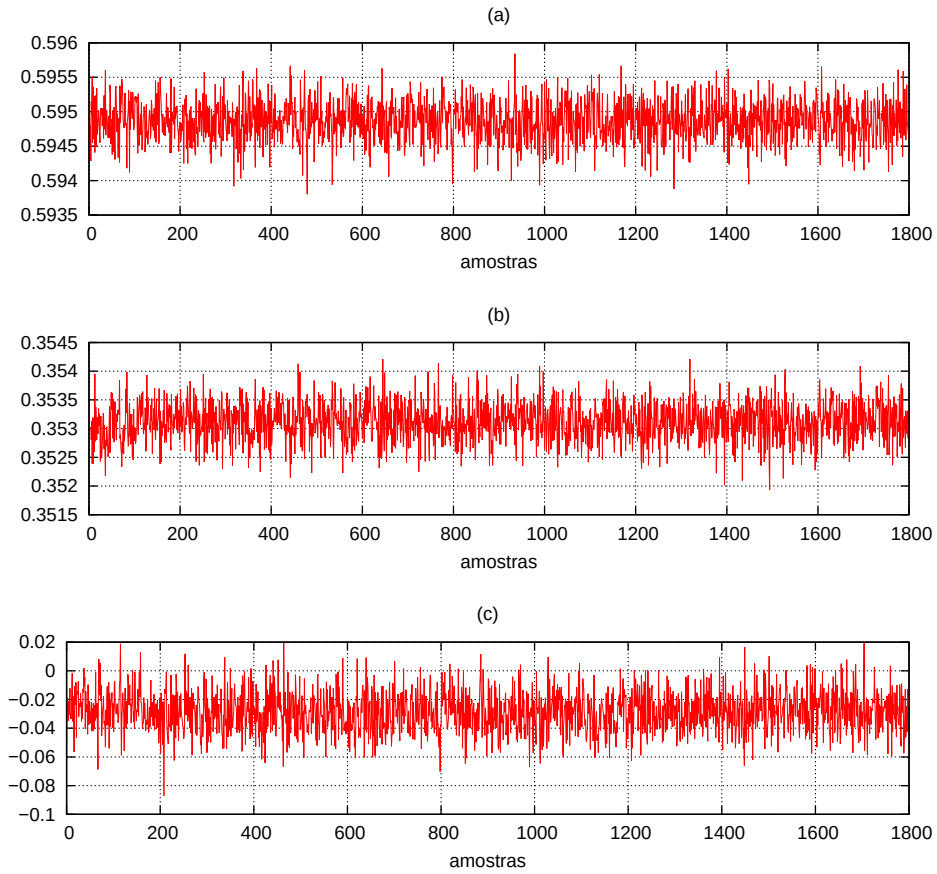

Figura 4.4: Cálculo de x (a), y (b) e θ (c) pelo sistema de visão durante 5 segundos. O robô foi mantido fixo na posição $(x,y) \approx (0.60, 0.35)$ e $\theta \approx 0$rad. Os desvios dos erros de localização são $\sigma_x = 2.99243 \times 10^{-4}$, $\sigma_y = 3.41511 \times 10^{-4}$ e $\sigma_\theta = 1.44964 \times 10^{-2}$.

com o valor dos parâmetros próximo do valor correto para o sistema e com variância pequena.

4.2 Estimação dos parâmetros

Para estimar os valores do modelo para o robô real foi realizado um procedimento semelhante ao descrito na seção 3.6 para o robô simulado. Como descrito na seção 4.1.2, foi considerado um tempo de atraso de 3 passos de amostragem. Para identificar o modelo linear, o robô foi acionado por 60 segundos com sinais PRBS mostrados na figura 4.6. No robô real, é importante que cada sinal seja aplicado por um tempo mínimo suficiente para os efeitos de tal entrada serem refletidos na saída do sistema, ou seja, o sinal tem que ser persistentemente excitante. Nos experimentos, cada novo nível de sinal de entrada

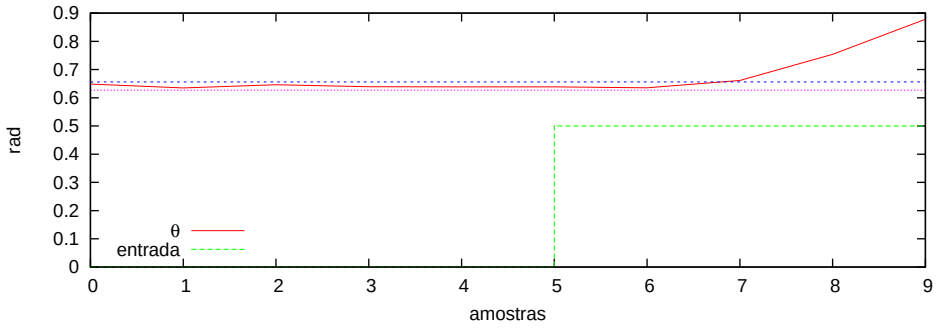

Figura 4.5: Variação do angulo θ_k em resposta a entradas do tipo degrau. Foram utilizados $e_1 = -e_2$ que resulta num giro do robô sobre seu próprio eixo. Observa-se um atraso na resposta do robô de 3 amostras. As linhas horizontais delimitam a variância de θ.

permaneceu inalterado por um tempo entre 5 a 8 períodos de amostragem. As medições são mostradas na figura 4.7. Os parâmetros estimados são mostrados na tabela 4.1.

Parâmetro	Valor estimado
a_1^L	1.48421
a_2^L	-0.483865
b_1^L	0.109806
b_2^L	0.0366508
b_3^L	0.141303
b_4^L	0.063934
a_1^Θ	1.53268
a_2^Θ	-0.532663
b_1^Θ	0.0321024
b_2^Θ	-0.00147427
b_3^Θ	-0.0428024
b_4^Θ	0.0063318

Tabela 4.1: Resultado da estimação dos parâmetros do modelo sem zona-morta com dados obtidos do sistema real.

A validação do modelo é mostrada na figura 4.8, para um conjunto de dados obtidos ao excitar o robô com sinais PRBS diferentes dos utilizados na identificação. Na validação, foram obtidos valores RMSE de 0.122571 para l e 0.284868 para θ.

Para o modelo com zona-morta, também foi seguido o mesmo procedimento realizado para o robô simulado, com a particularidade dos sinais de entrada serem mantidos fixos por alguns períodos de amostragem. Foram aplicados sinais gaussianos de média nula e desvio padrão de ± 0.3, de forma excitar o robô no ponto onde existe zona-morta.

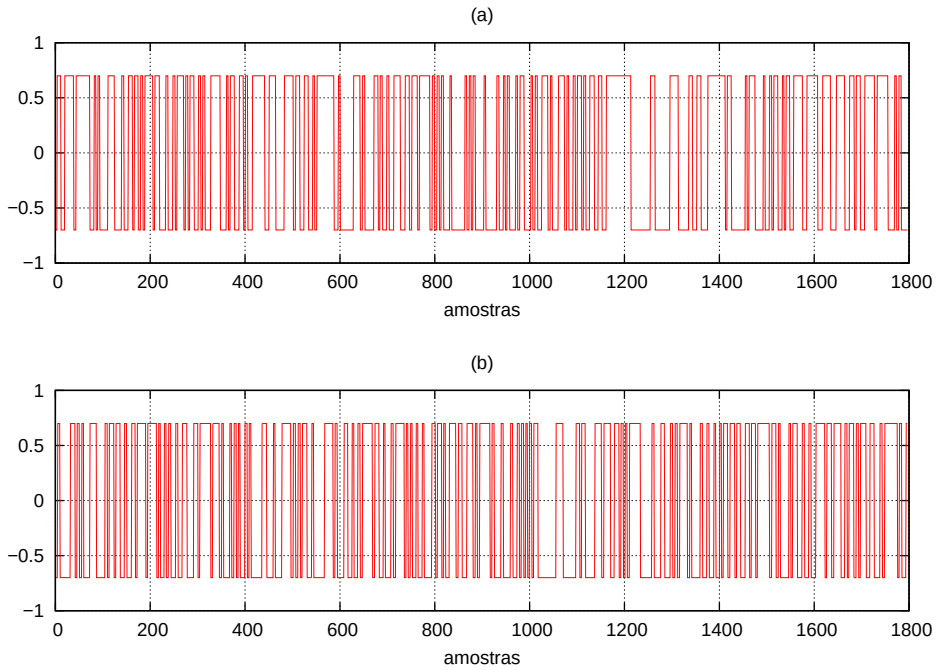

Figura 4.6: Sinais PRBS aplicados na entrada direita (a) e entrada esquerda (b) do robô real. Cada nível de sinal é mantido fixo por um tempo de 5 à 8 períodos de amostragem.

Esse sinal é mostrado na figura 4.9. As medições das saídas do robô podem ser vistas na figura 4.10 e as figuras 4.11 à 4.17 mostram a evolução dos parâmetros durante a identificação. Mais uma vez foi escolhido um limite máximo para η_i de 0.5. Os parâmetros no fim da identificação podem ser vistos na tabela 4.2.

Finalmente, a validação do modelo com zona-morta pode ser visto na figura 4.18. Para tal, foram utilizados sinais de entrada semelhantes aos da figura 4.9. Foram obtidos valores RMSE de 0.257272 para a saída l e 0.0950367 para θ.

Assim como desenvolvido no capítulo anterior, o modelo sem zona-morta foi excitado com o mesmo sinal utilizado na verificação do modelo com zona-morta. Mais uma vez o objetivo é mostrar que o modelo com zona-morta apresenta melhor desempenho. O resultado do experimento pode ser visto na figura 4.19. Os valores RMSE foram 0.496254 para a saída l e 0.183715 para θ, e como esperado, são maiores do que as obtidas através do modelo que considera a zona-morta.

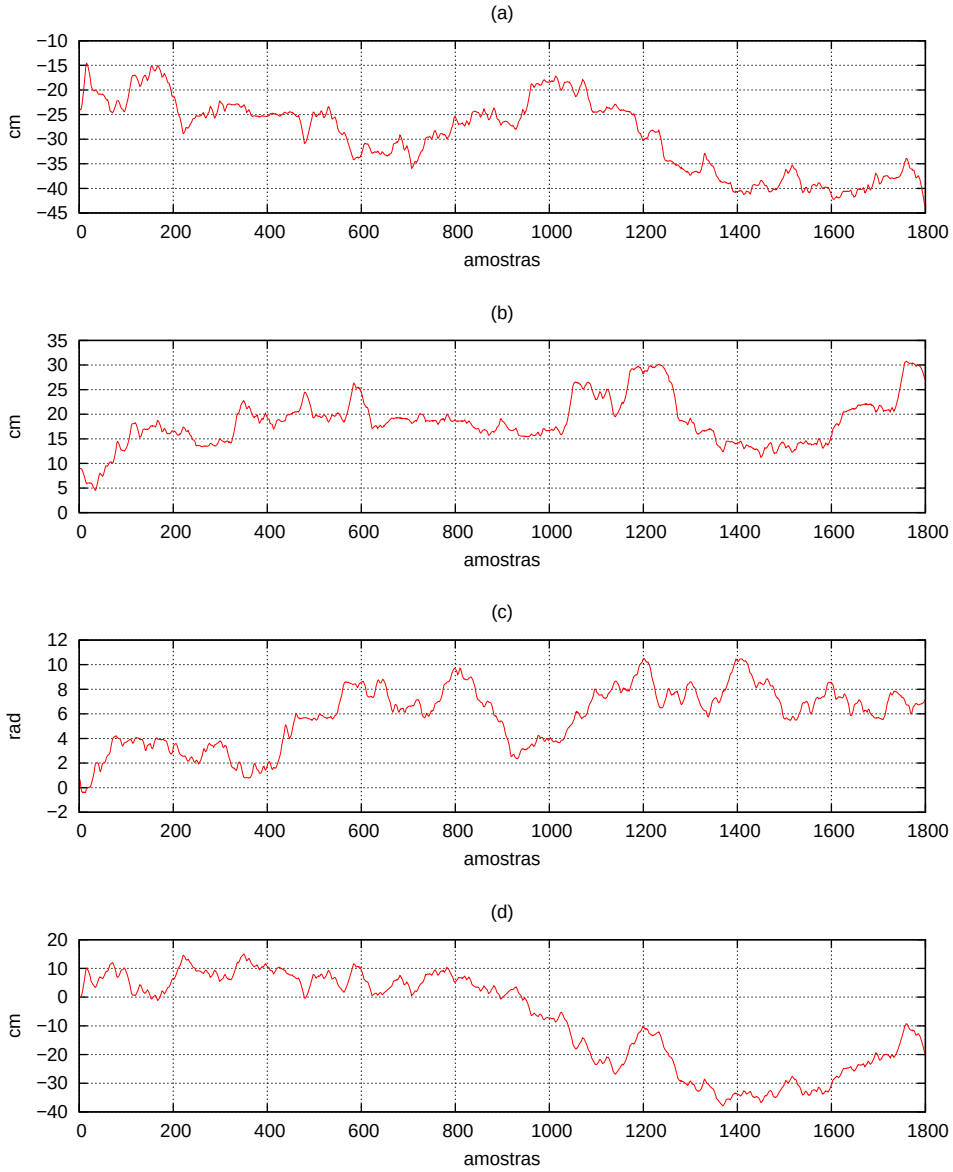

Figura 4.7: Resposta do robô às entradas PRBS mostradas na figura 4.6: (a) x, (b) y, (c) θ e (d) $\hat{l}$.

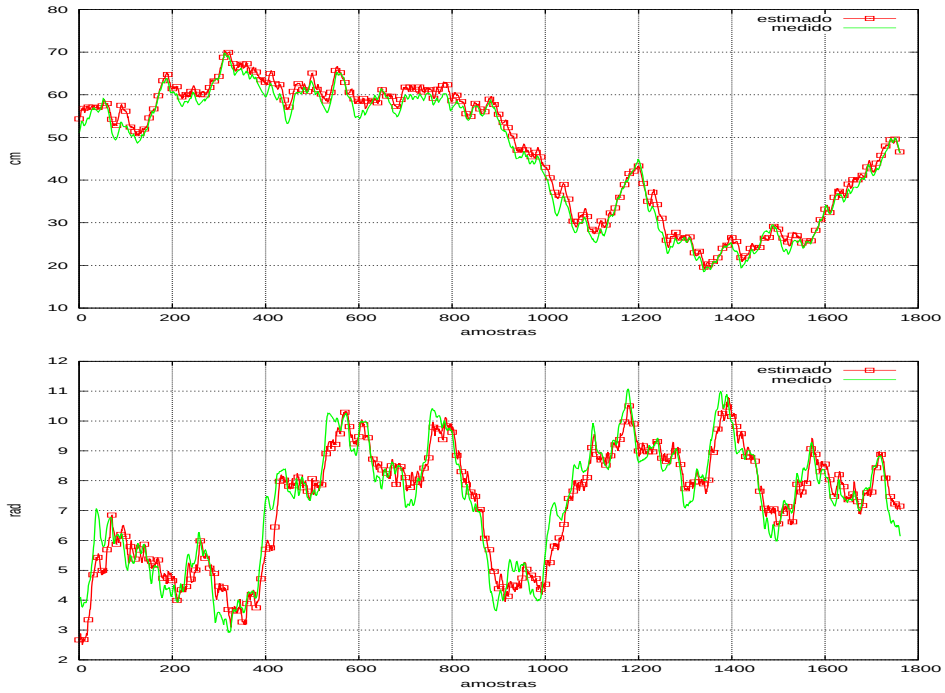

Figura 4.8: Validação do modelo sem zona-morta para o robô real, sendo a evolução de l ao topo e a evolução de θ a baixo.

Parâmetro	Valor médio estimado	Variância
a_1^L	1.58827	1.6920×10^{-3}
a_2^L	-0.58823	1.6830×10^{-3}
b_1^L	0.34727	9.4835×10^{-4}
b_2^L	0.11353	1.0293×10^{-3}
b_3^L	0.38260	1.0403×10^{-3}
b_4^L	0.10499	2.3726×10^{-3}
a_1^Θ	1.59984	2.4796×10^{-4}
a_2^Θ	-0.59988	2.4911×10^{-4}
b_1^Θ	0.10548	1.4409×10^{-4}
b_2^Θ	-0.00417	2.0968×10^{-4}
b_3^Θ	-0.10844	2.4055×10^{-5}
b_4^Θ	-0.00325	8.5837×10^{-5}
η_1	0.14665	1.1666×10^{-4}
η_2	0.12005	1.1973×10^{-4}

Tabela 4.2: Resultado de 6 estimações dos parâmetros do modelo com zona-morta a partir de dados obtidos do robô real

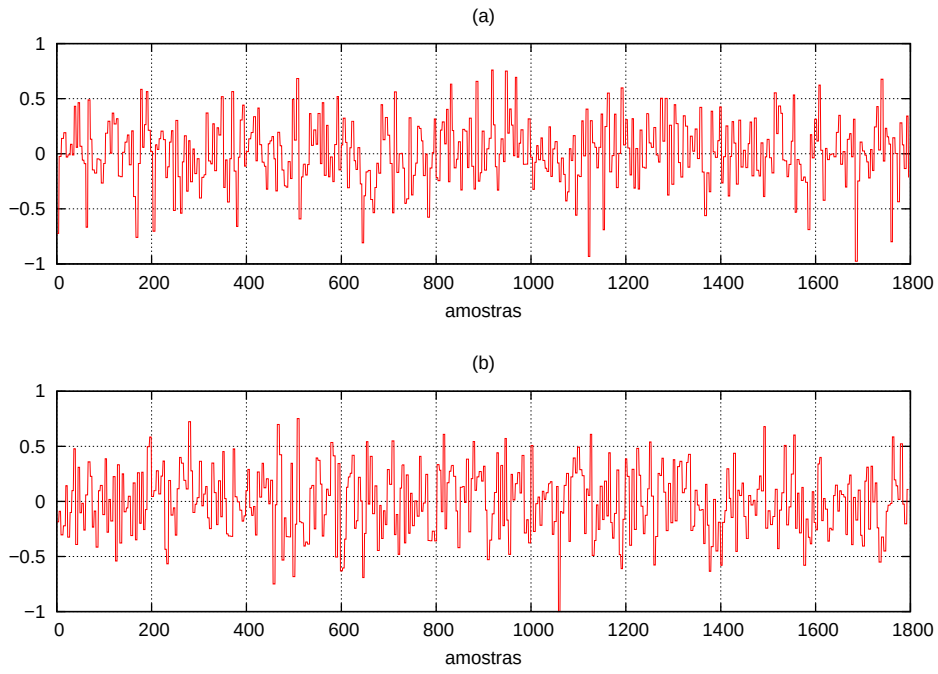

Figura 4.9: Sinais Gaussianos aplicados na entrada direita (a) e entrada esquerda (b) do robô real. Cada nível do sinal de entrada foi mantido fixo por um tempo de 5 a 8 períodos de amostragem.

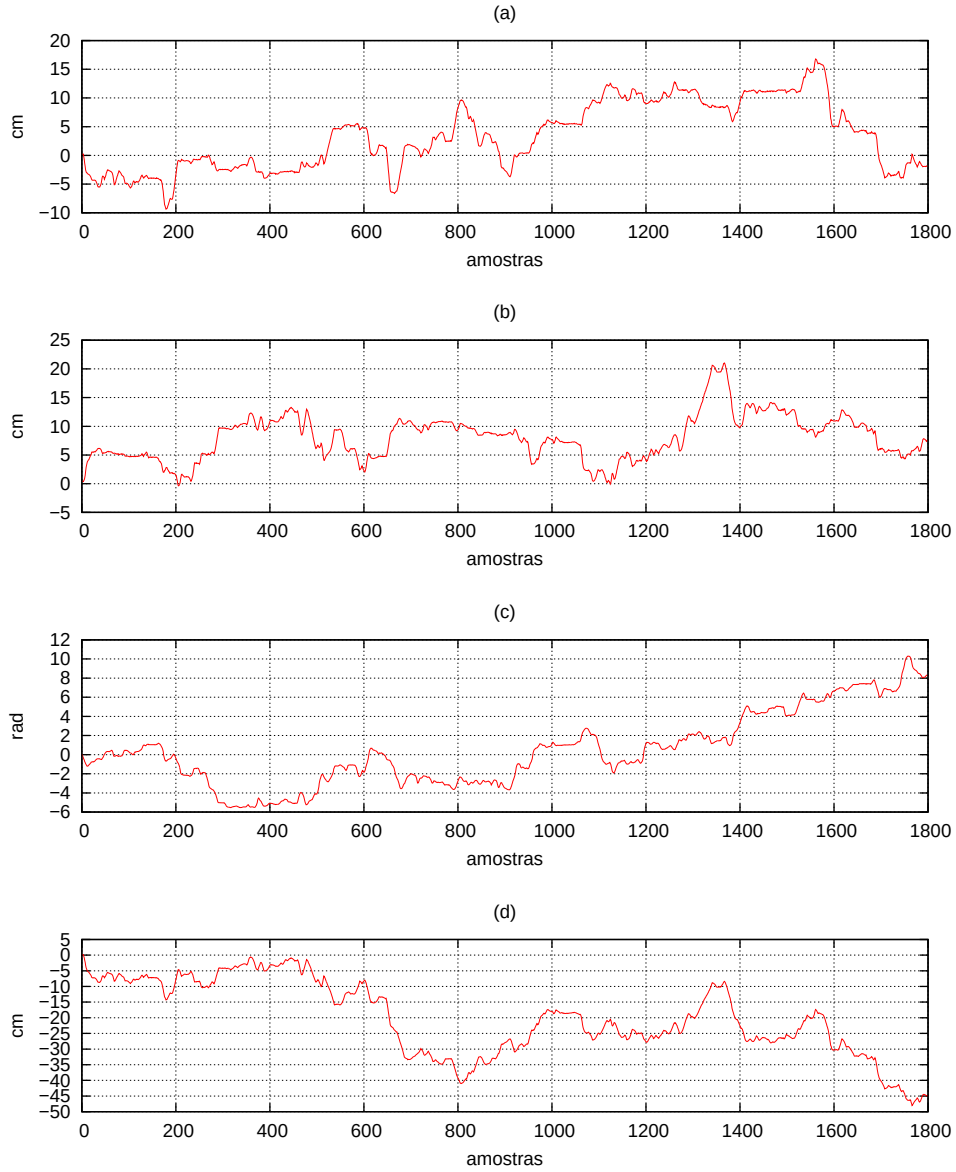

Figura 4.10: Resposta do robô real às entradas gaussianas mostradas na figura 4.9: (a) x, (b) y, (c) θ e (d) $\hat{l}$.

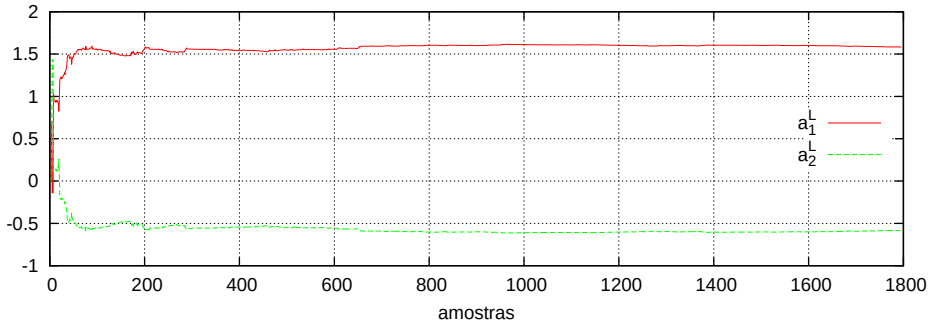

Figura 4.11: Evolução da estimação dos parâmetros a_1^L e a_2^L do modelo com zona-morta para o robô real.

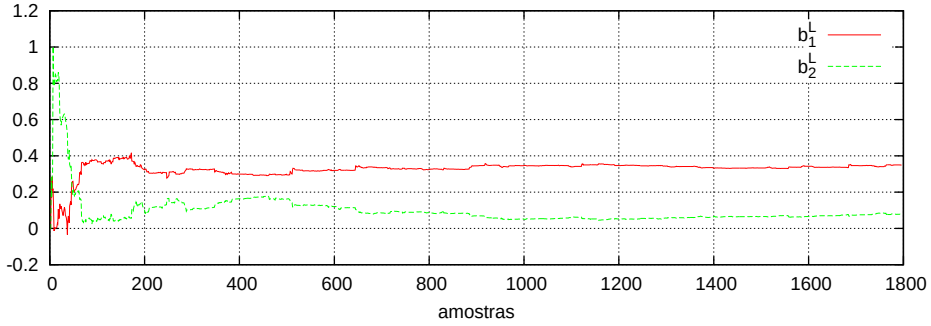

Figura 4.12: Evolução da estimação dos parâmetros b_1^L e b_2^L do modelo com zona-morta para o robô real.

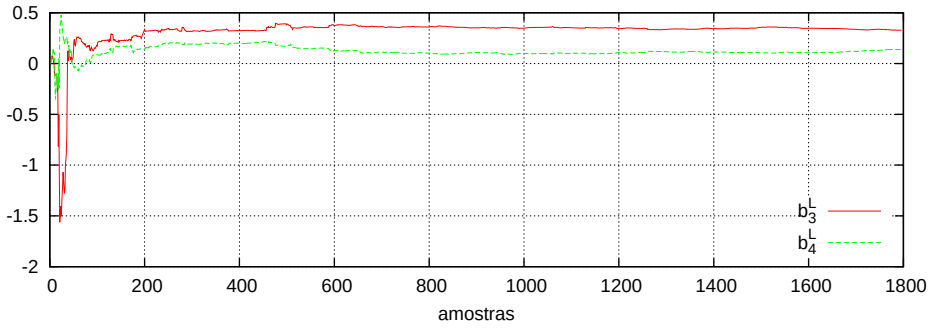

Figura 4.13: Evolução da estimação dos parâmetros b_3^L e b_4^L do modelo com zona-morta para o robô real.

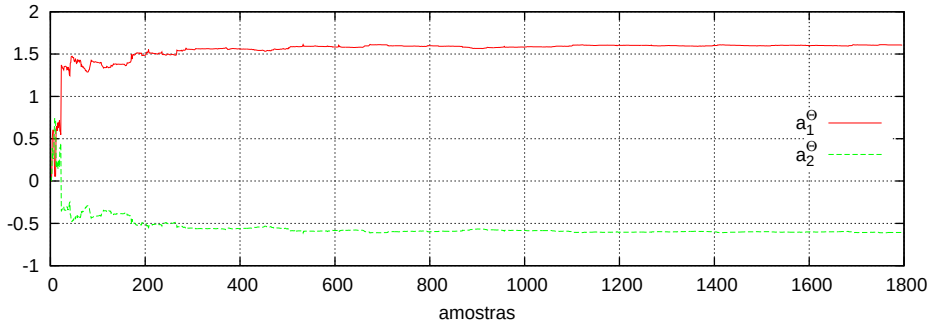

Figura 4.14: Evolução da estimação dos parâmetros a_1^Θ e a_2^Θ do modelo com zona-morta para o robô real.

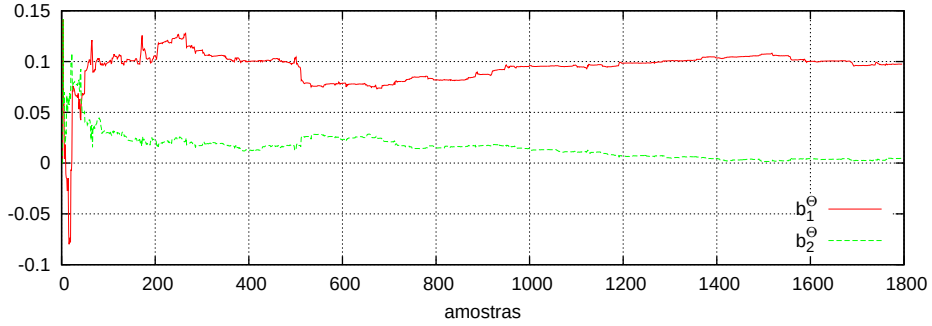

Figura 4.15: Evolução da estimação dos parâmetros b_1^Θ e b_2^Θ do modelo com zona-morta para o robô real.

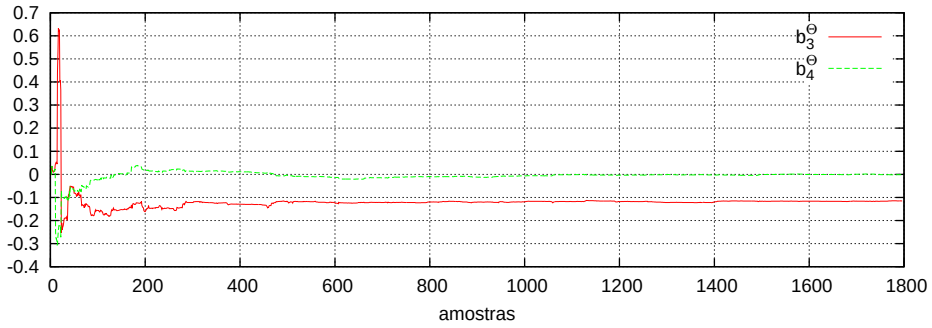

Figura 4.16: Evolução da estimação dos parâmetros b_3^Θ e b_4^Θ do modelo com zona-morta para o robô real.

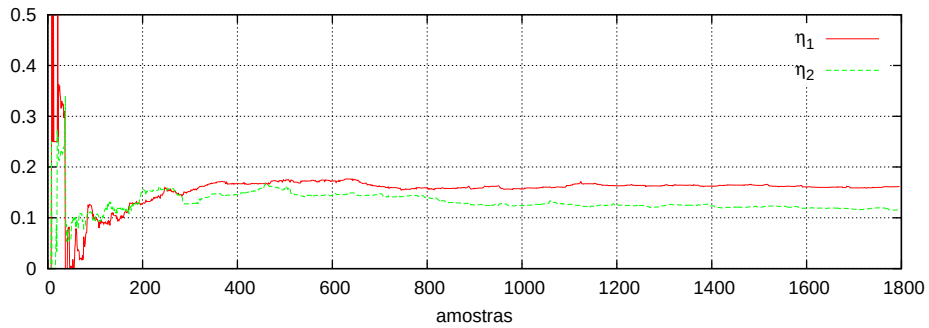

Figura 4.17: Evolução da estimação dos parâmetros η_1 e η_2 do modelo com zona-morta para o robô real.

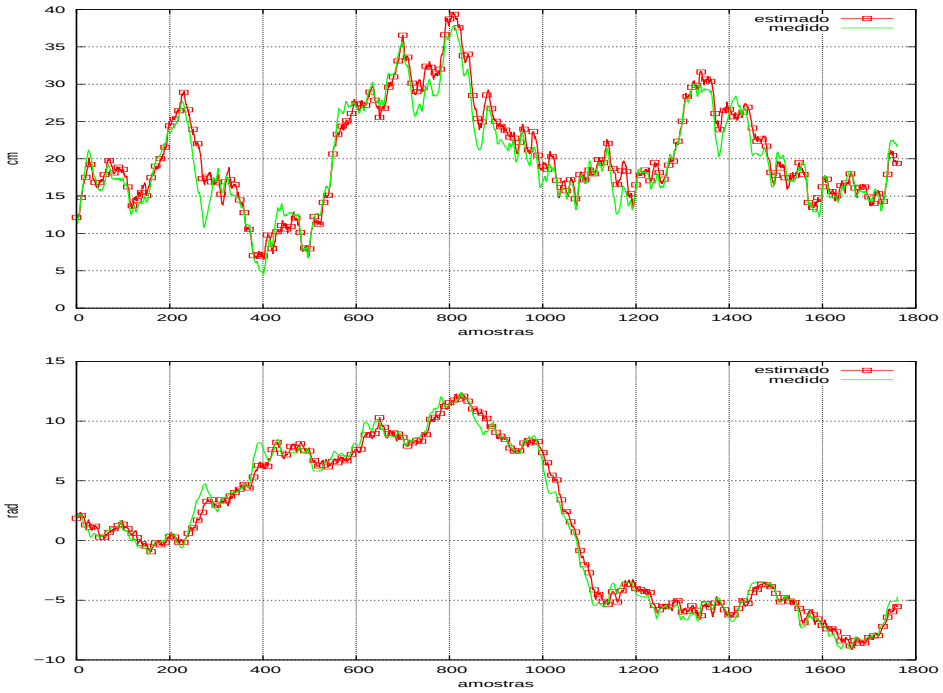

Figura 4.18: Validação do modelo com zona-morta para o robô real, sendo a evolução de l ao topo e a evolução de θ abaixo.

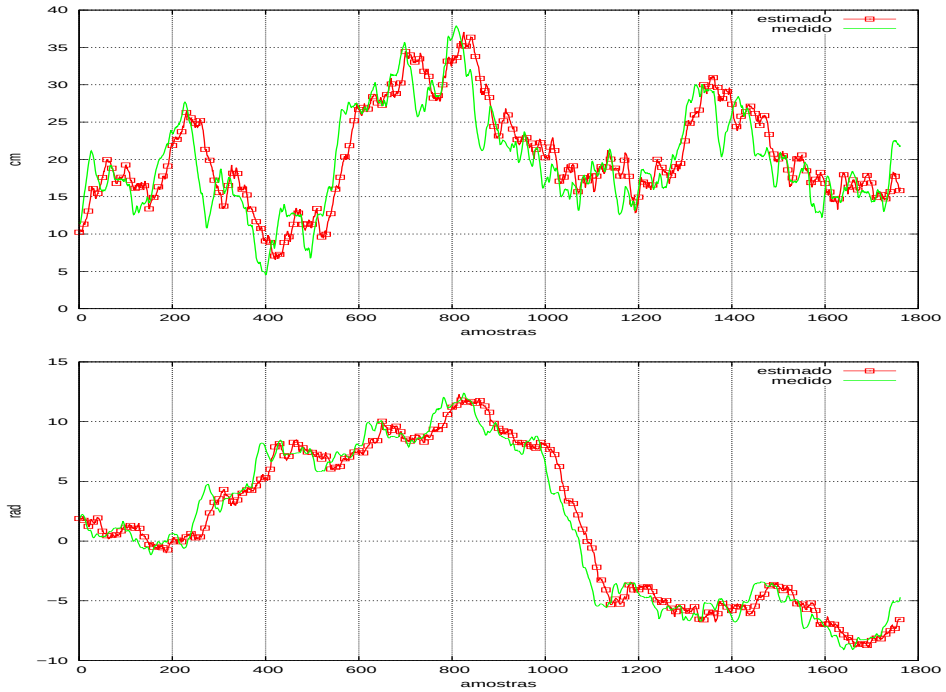

Figura 4.19: Saídas do modelo sem zona-morta obtido de dados do robô real e entradas gaussianas de baixa amplitude, sendo a evolução de l ao topo e a evolução de θ abaixo.

Capítulo 5

Conclusões e trabalhos futuros

Neste trabalho foi desenvolvido um método de identificação de modelos para robôs móveis com acionamento diferencial que considera a zona-morta presente no sistema. Foi utilizado um micro-robô para os testes reais, cuja posição e orientação são obtidos por um sistema de visão global. Porém o método pode ser aplicado em outros veículos móveis com as mesmas características, desde que utilizados sistemas de sensoriamento que forneçam o posicionamento do robô com um certo grau de precisão.

Os modelos obtidos se mostraram melhores do que os equivalentes que não consideram as não-linearidades do sistema, principalmente para pequenos sinais de entrada. Porém, a abordagem utilizada permite somente identificar os limites da zona-morta e não os seus ganhos para entradas positivas e negativas, visto a impossibilidade de desvinculá-los dos ganhos do sistema linear. É necessário utilizar uma identificação recursiva, pois é preciso estimar os limites da zona-morta a cada iteração e utilizá-los para atualizar o vetor de regressores. Uma outra restrição do método é a necessidade de decidir empiricamente um valor máximo que os limites da zona-morta podem assumir durante a recursão, e saturá-lo nesse valor caso o ultrapasse, sendo importante que o valor máximo para os limites da zona-morta seja menor do que o valor máximo do sinal de entrada.

A modelagem do robô se baseou algumas aproximações. Por exemplo, foi considerado um robô de características simétricas, e centro de massa no centro do robô. O modelo utilizado no simulador porém é mais completo, obtido de uma modelagem fenomenológica que leva em conta também o modelo dos motores e possíveis assimetrias do robô. Um trabalho futuro é realizar a identificação do robô real utilizando esse modelo, e através do modelo obtido, estimar os valores dos parâmetros físicos do sistema.

A zona-morta do sistema foi considerada como estática e simétrica, o que simplificou o nosso problema. Como trabalho futuro é possível descrever equações paramétricas que considerem a zona-morta assimétrica e estimar os valores de seus parâmetros. Além disso, existem outros modelos de zona-morta disponíveis na literatura e que podem ser usados

como substitutos ao utilizado neste trabalho.

Atualmente, o sistema utiliza um algoritmo de controle PID para gerar os sinais de controle que levem o robô a referência desejada. As constantes do controlador são definidas empiricamente, através de tentativa e erro e da análise do comportamento do robô. O modelo obtido neste trabalho também pode ser utilizado para sintonizar automaticamente os parâmetros do controlador.

Durante a realização deste trabalho, foram gerados os trabalhos Mendes et al. (2010) e Mendes & Medeiros (2010), publicados respectivamente no *Congresso Brasileiro de Automática* (CBA) em 2010, e no *Latin American Robotics Symposium* (LARS) também 2010.

Referências Bibliográficas

Aguirre, Luis Antonio (2007), *Introdução à identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais*, Editora UFMG.

Conceição, A.S., A.P. Moreira & P.J. Costa (2009), 'Practical approach of modeling and parameters estimation for omnidirectional mobile robots', *Mechatronics, IEEE/ASME Transactions on* **14**(3), 377 –381.

Er-Wei & Bai (2002), 'Identification of linear systems with hard input nonlinearities of known structure', *Automatica* **38**(5), 853 – 860.

Guerra, Patricia N. (2005), Modelagem linear e identificação do modelo dinâmico de um robô móvel com acionamento diferencial, Dissertação de mestrado, UFRN, Natal, RN.

Malik, F.M., M.B. Malik & K. Munawar (2009), Parameter estimation of a single-link robotic manipulator in presence of dead-band and dead-zone, *em* 'Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on', pp. 1 –5.

Mendes, Ellon P. & Adelardo A. D. Medeiros (2008), 'Sistema de localização visual da equipe de futebol de robôs poti-ufrn (versão 2008) na categoria very small size', *TDP submetido a Competição Brasileira de Robótica 2008* .

Mendes, Ellon P. & Adelardo A.D. Medeiros (2010), Identification of quasi-linear dynamic model with dead zone for mobile robot with differential drive, *em* 'VII LARS - Latin American Robotics Symposium'.

Mendes, Ellon P., Adelardo A.D. Medeiros & Luiz H.R. Silva (2010), Identification of dynamic model with dead-zone for mobile robot with differential drive, *em* 'XVIII CBA - Congresso Brasileiro de Automática'.

Pereira, Guilherme Augusto Silva (2000), Identificação e controle de micro-robôs móveis, Dissertação de mestrado, Universidade Federal de Minas Gerais.

Rules for the IEEE Very Small Competition (2008), *7th LATIN AMERICAN IEEE STUDENT ROBOTICS COMPETITION* .

Sanca, Armando Sanca (2006), Controle com compensação de atrito para estruturas de base móveis de robôs manipuladores, Dissertação de mestrado, UFBA, Salvador, BA.

Tao, Gang & Petar V. Kokotović (1994), Adaptive control of plants with unknow dead-zones, *em* 'IEEE Transactions on Automatic Control', Vol. 39, pp. 59–68.

Vieira, Frederico Carvalho (2006), Controle dinâmico de robôs móveis com acionamento diferencial, Dissertação de mestrado, UFRN, Natal, RN.

Vörös, Jozef (2001), 'Parameter identification of wiener systems with discontinuous nonlinearities', *Systems & Control Letters* **44**, 363–372.

Vörös, Jozef (2003), 'Recursive identification of hammerstein systems with discontinuous nonlinearities containing dead-zones', *IEEE Transactions on Automatic Control* **48**(12), 2203–2206.

Vörös, Jozef (2004), 'Parameter identification of hammerstein systems with asymmetric dead-zones', *Journal of ELECTRICAL ENGINEERING* **55**(1-2), 46–49.

Vörös, Jozef (2006), 'Recursive identification of hammerstein systems with polynomial nonlinearities', *Journal of Electrical Engineering* **57**(1), 42–46.