UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Mecânica

PEDRO RAMON DE MELLO SILVA

Odometria Inercial de Robôs Móveis Assistida por Redes Neurais

CAMPINAS
2022

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

PEDRO RAMON DE MELLO SILVA

Odometria Inercial de Robô Móvel Assistida por Redes Neurais

Tese de doutorado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Doutor em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico.

Orientador: Prof. Dr. Paulo Roberto Gardel Kurka.
Coorientador: Prof. Dr. Calebe Paiva Gomes de Souza.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO PEDRO RAMON DE MELLO SILVA, E ORIENTADO PELO PROF. DR. PAULO ROBERTO GARDEL KURKA.

.....
ASSINATURA DO ORIENTADOR

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

Si38o

Silva, Pedro Ramon de Mello, 1987-
Odometria inercial de robô móvel assistida por redes neurais / Pedro Ramon de Mello Silva. – Campinas, SP : [s.n.], 2022.

Orientador: Paulo Roberto Gardel Kurka.
Coorientador: Calebe Paiva Gomes de Souza.
Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robôs autônomos. 2. Sistemas de navegação inercial. 3. Redes neurais de Elman. I. Kurka, Paulo Roberto Gardel, 1958-. II. Souza, Calebe Paiva Gomes de, 1979-. III. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. IV. Título.

Informações Complementares

Título em outro idioma: Inertial odometry of mobile robot assisted by neural networks
Palavras-chave em inglês:
Autonomous robots
Inertial navigation systems
Elman neural networks
Área de concentração: Mecânica dos Sólidos e Projeto Mecânico
Titulação: Doutor em Engenharia Mecânica
Banca examinadora:
Paulo Roberto Gardel Kurka [Orientador]
Ludmila Corrêa de Alkmin e Silva
Eleri Cardozo
Caio Fernando Rodrigues dos Santos
Romulo Gonçalves Lins
Data de defesa: 01-06-2022
Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-8026-6014
- Currículo Lattes do autor: http://lattes.cnpq.br/5497987370657942

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

TESE DE DOUTORADO ACADÊMICO

Odometria Inercial de Robô Móvel Assistida por Redes Neurais

Autor: Pedro Ramon de Mello Silva
Orientador: Paulo Roberto Gardel Kurka

A Banca Examinadora composta pelos membros abaixo aprovou esta Tese:

Prof. Dr. Paulo Roberto Gardel Kurka, Presidente
DSI/FEM/UNICAMP

Profa. Dra. Ludmila Corrêa de Alkmin e Silva
DSI/FEM/UNICAMP

Prof. Dr. Eleri Cardozo
DCA/FEEC/UNICAMP

Prof. Dr. Caio Fernando Rodrigues dos Santos
INSPER

Prof. Dr. Romulo Gonçalves Lins
CECS/UFABC

A Ata de defesa com as respectivas assinaturas dos membros encontram-se no SIGA/Sistema de Fluxo de Tese e na Secretaria do Programa da Unidade.

Campinas, 01 de Junho de 2022.

Dedicatória

Aos meus queridos pais, Stela Mello e Edson Marques, que acreditaram no meu sonho e sempre me apoiaram. Ao meu Professor Paulo Kurka, que fez toda minha carreira acadêmica acontecer e sempre ter sido um orientador presente e motivador. Ao meu Professor Franklyn, que foi meu primeiro professor, eterno mestre e segundo pai. Aos meus mestres de graduação : Max Trindade, Augusto Real, Alex Ferreira, Alex Guedes e Marcio Moscoso, que me deram bases e incentivos para ser um bom engenheiro e cientista. Este trabalho foi desenvolvido com o apoio do CNPq, com a concessão de bolsa de estudo.

Agradecimentos

Ao meu orientador, Prof. Dr. Paulo Roberto Gardel Kurka por ter acreditado no meu potencial, pela tutoria e pela orientação ao longo dos anos de minha pós-graduação.

Ao Prof. Dr. Calebe Paiva, que me deu suporte na escrita desta dissertação.

Aos meus amigos de pós-graduação, que acompanharam meu trabalho e desenvolvimento ao longo dos anos.

Aos meus pais, que me deram suporte e força para seguir na carreira acadêmica.

A Universidade Estadual de Campinas e todos os professores excelentes que conheci.

O presente trabalho foi realizado com apoio da Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo no 141220/2015-5.

Absorva aquilo que é útil, descarte o que não é e adicione aquilo que for unicamente seu.

Bruce Lee

Resumo

Silva, Pedro Ramon de Mello. Odometria Inercial de Robô Móvel Assistida por Redes Neurais. 2022. 95p. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

Atualmente, diversas técnicas de localização de robôs são implementadas para formar o seu sistema de navegação autônoma. Para isso, abordagens de odometria são exploradas utilizando sensores como *encoders*, câmeras, *lasers*, entre outros. Porém, técnicas de odometria de robôs utilizando unicamente sensores inerciais como IMU são escassas na literatura devido a complexidade de mitigar os erros acumulados no processo de inferir o deslocamento utilizando um acelerômetro e a orientação utilizando um giroscópio. Esta tese propõe uma técnica para viabilizar a odometria inercial de um robô terrestre utilizando uma IMU em sistemas com restrições computacionais (e.g. microcontroladores). A técnica é baseada no processamento dos sinais de uma unidade de medidas inerciais em complemento com os sinais de entrada de torque do atuador no robô. Este processamento realiza a estimativa das variáveis de odometria através de uma arquitetura de redes neurais recorrentes de Elman. O método proposto é capaz de minimizar os erros cumulativos de integração numérica dos dados do acelerômetro e giroscópio, a fim de obter a pose do robô ao longo de sua trajetória de forma consistente e com desempenho comparável à uma odometria de rodas via modelo cinemático, porém, sem a presença dos erros sistemáticos e não-sistemáticos desta última abordagem citada. A técnica foi validada em um robô diferencial em ambiente *indoor*, podendo ser aplicada em outros sistemas robóticos que atuam neste mesmo contexto, como robôs aspiradores e do tipo AGV (*Automated guided vehicle*).

Palavras-chave: Robôs autônomos, Sistemas de navegação inercial, Redes neurais de Elman.

Abstract

Silva, Pedro Ramon de Mello Silva. Inertial Odometry of Mobile Robot Assisted By Neural Networks. 2022. 95p. Tese (Doutorado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

Currently, several robot localization techniques are implemented to form its autonomous navigation system. Therefore, odometry approaches are explored using sensors such as *encoders*, cameras, *lasers*, among others. However, robot odometry techniques using only inertial sensors such as IMU are scarce in the literature due to the complexity of mitigating the errors accumulated in the process of inferring displacement using an accelerometer and orientation using a gyroscope. This thesis proposes a technique to enable the inertial odometry of a terrestrial robot using an IMU in systems with computational constraints (e.g. microcontrollers). The technique is based on the processing of signals from an inertial measurement unit in addition to the torque input signals from the actuator to the robot. This processing performs the estimation of odometry variables through an architecture of recurrent Elman neural networks. The proposed method is able to minimize the cumulative errors of numerical integration of the accelerometer and gyroscope data, in order to obtain the robot pose along its trajectory consistently and with performance comparable to a wheel odometry via kinematic model, however , without the systematic and non-systematic errors of the latter. The technique was validated in a differential robot in a *indoor* environment, and can be applied to other robotic systems that work in this same context, such as vacuum robots and AGV (*Automated guided vehicle*) robots.

Key words: Autonomous robots, Inertial navigation systems, Elman neural networks.

Lista de Ilustrações

3.1 Sistema de coordenadas do acelerômetro 26
3.2 Sistema de coordenadas do giroscópio 28
4.1 Rede Neural Perceptron Multicamadas 35
4.2 Rede de Elman 36
5.1 Dependências geométricas do robô diferencial 39
5.2 Diagrama de blocos do motor CC 42
5.3 Arquitetura das *RNRs* utilizadas 43
6.1 Diagrama de blocos do modelo do robô 46
6.2 Organização dos dados para treinamento das redes 47
6.3 Torque nos motores para treinamento 48
6.4 Torques utilizado para validação 49
6.5 Velocidade linear real e estimada em uma iteração 50
6.6 Velocidade angular real e estimada em uma iteração 50
6.7 Erros de trajetória para uma rede treinada 51
6.8 *Boxplot* para a rede de velocidade linear 52
6.9 *Boxplot* para a rede de velocidade angular 52
6.10 Torques de teste 53
6.11 Deslocamentos estimados e reais 54
6.12 Orientações estimadas e reais 54
6.13 Trajetória estimada e real 55
6.14 Trajetória estimada com integração direta 56
7.1 Plataforma de Teste. 58
7.2 Diagrama de Blocos de Funcionamento do Sistema Robótico 58
7.3 Diagrama de fluxo de aquisição do *dataset* 59
7.4 Kinovea rastreando o movimento do robô capturado por câmera 60
7.5 Arquitetura da Rede Neural para Velocidade Linear e Angular 61
7.6 Correntes do motor esquerdo e direito para treinamento das redes 62
7.7 Aceleração longitudinal e lateral para treinamento das redes 63
7.8 Velocidade angular para treinamento das redes 63
7.9 Velocidade Linear do *Ground Truth* 64

7.10 Velocidade angular do *Ground Truth* 64
7.11 Correntes nos motores para validação 65
7.12 Aceleração linear e lateral para validação 66
7.13 Velocidade angular para validação 66
7.14 Comparativo entre velocidades lineares 67
7.15 Comparativo entre velocidades angulares 68
7.16 Comparativo entre deslocamentos estimados 69
7.17 Comparativo entre orientações estimadas 69
7.18 Comparativo entre trajetórias estimadas 70
7.19 Comparativo entre deslocamentos estimados com integração direta 71
7.20 Comparativo entre orientações estimadas com integração direta 72
7.21 Deslocamento estimado no teste 1 74
7.22 Orientação estimada no teste 1 74
7.23 Trajetória estimada no teste 1 75
7.24 Deslocamento estimado com integração direta do sinal do acelerômetro 76
7.25 Orientação estimada com integração direta do sinal do giroscópio 76
7.26 Distribuição de probabilidade do erro para V_l 77
7.27 Distribuição de probabilidade do erro para ω_z 78
7.28 Distribuição de probabilidade do erro para V_l 78
7.29 Distribuição de probabilidade do erro para ω_z 79
7.30 Deslocamento estimado no teste 2 80
7.31 Orientação estimada no teste 2 81
7.32 Trajetória estimada no teste 2 81
7.33 Deslocamento estimado com integração direta do sinal do acelerômetro 82
7.34 Orientação estimada com integração direta do sinal do giroscópio 83
7.35 Distribuição de probabilidade do erro para V_l 84
7.36 Distribuição de probabilidade do erro para ω_z 84
7.37 Distribuição de probabilidade do erro para V_l 85
7.38 Distribuição de probabilidade do erro para ω_z 85
7.39 Trajetória do teste 1 com filtro de kalman 87

Nomenclatura

Variáveis

$\vec{a}_m$	-	Vetor de aceleração medida pelo acelerômetro
S_n	-	Matriz de sensibilidade
T_n	-	Matriz de desalinhamento
$\vec{a}_b$	-	Vetor de aceleração no referencial móvel
$\vec{b}_n(t,T)$	-	Vetor de *bias*
$\vec{n}_n(t)$	-	Vetor de ruído
R_ψ	-	Matriz de rotação em torno do ângulo ψ
R_θ	-	Matriz de rotação em torno do ângulo θ
R_ϕ	-	Matriz de rotação em torno do ângulo ϕ
$\vec{\omega_m}$	-	Velocidade angular medida pelo giroscópio
θ_g	-	Ângulo estimado pelo giroscópio
$\vec{V}_I(t)$	-	Velocidade linear estimada pelo acelerômetro no referencial inercial
$P(t)$	-	Posição estimada pelo acelerômetro
$\delta_n(k)$	-	Incremento de deslocamento no eixo n
W_h	-	Matriz de pesos da camada h
σ_h	-	Função de ativação da camada h
ω_l	-	Velocidade angular da roda esquerda
ω_r	-	Velocidade angular da roda direita
V_l	-	Velocidade linear no eixo longitudinal do robô
θ_z	-	Ângulo de orientação do robô no plano
v_a	-	Tensão no motor
$T(t)$	-	Torque do motor
i_a	-	Corrente na armadura do motor

Acrônimos

IMU	-	Inertial Measurement Unit
MEMS	-	Micro-Electro-Mechanical-System
ASIC	-	Application Specific Integrated Circuit
FPGA	-	Field Programmable Gate Array
RNA	-	Rede Neural Artificial
MLP	-	Multilayer Perceptron
RNR	-	Rede Neural Recorrente
ZVU	-	Zero Velocity Update
LSTM	-	Long Short-Term Memory
GRU	-	Gated Recurrent Unit
SoC	-	System-On-a-Chip
LIDAR	-	Light Detection and Ranging

SUMÁRIO

Lista de Abreviaturas e Siglas **x**

1 Introdução **16**
1.1 Objetivo 18
1.2 Contribuição Científica 19
1.3 Estrutura da Tese 20

2 Estado da arte **21**

3 Sistemas Inerciais em Odometria **25**
3.1 Acelerômetro 25
3.2 Giroscópio 28
3.3 Odometria Inercial 29
3.3.1 Estimativa de Orientação 29
3.3.2 Estimativa de Deslocamento 31
3.3.3 Estimativa de Trajetória via Odometria Inercial 33

4 Fundamentos de Redes Neurais Artificiais **34**
4.1 Redes Neurais Artificiais 34
4.2 Redes Neurais Recorrentes 36
4.3 Treinamento de Redes Neurais via Retropropagação 37

5 Método de Odometria Inercial via Redes Neurais **38**
5.1 Modelo Dinâmico de um Robô Diferencial 38
5.2 Arquitetura de Redes Neurais Recorrentes 42

6 Verificação Numérica da Técnica Proposta **46**
6.1 Ambiente de Simulação 46
6.2 Treinamento das *RNRs* Aplicadas ao Robô Diferencial para Odometria Inercial . . 47
6.3 Resultados Simulados da Estimativa de Trajetória 53

7 Aplicação Experimental **57**

7.1 Plataforma Robótica e Configuração Experimental 57
7.2 Treinamento e Validação das Redes Neurais para Odometria Inercial 61
7.3 Testes das Redes Neurais para Odometria Inercial 73
7.3.1 Trajetória de Teste 1 73
7.3.2 Trajetória de Teste 2 79
7.3.3 Considerações para Aplicação em Fusão Sensorial 86
7.4 Link para dataset 87

8 Conclusões **88**
8.1 Propostas para Trabalhos Futuros 89

Referências **90**

1 Introdução

A navegação autônoma é uma tecnologia chave em pesquisa no campo de robótica móvel (Cai et al., 2012). Sondas robóticas espaciais e carros autônomos, por exemplo, necessitam dessa funcionalidade a fim de cumprir a missão designada. Para que a navegação seja realizada, é necessário que algumas capacidades sejam implementadas, como: planejamento de rota, controle de trajetória, localização e mapeamento (Yuan et al., 2016). Enquanto planejamento de rota e controle de trajetória são abordados de forma consistente na literatura (Sariff e Buniyamin, 2006), o processo de localização e mapeamento em robôs é uma área que ainda possui diversas abordagens a serem exploradas e estudadas. O mapeamento trata da questão de construir um mapa métrico ou topológico do ambiente em que o robô está inserido, enquanto a localização trata de estimar a sua posição e orientação neste mapa. Esse processo simultâneo é chamado de SLAM (*Simultaneous Localization and Mapping*) (Mahrami et al., 2013).

Diversos sensores têm sido utilizados na tarefa de realizar a localização de um robô, como *encoders*, LIDAR (*Light Detection and Ranging*), câmeras, GPS, radares, ou sistemas inerciais como *IMU* (Inertial Measurement Unit). Esses instrumentos contribuem no objetivo de estimar a posição e orientação do robô definido como pose, enquanto o mesmo se movimenta em um determinado ambiente. Uma forma adicional de aprimorar a estimação da pose de um robô é através da aplicação de técnicas de fusão sensorial, principalmente com a aplicação de filtros de Kalman e suas variantes (Ullah et al., 2019). Estes filtros podem realizar a fusão de medidas complementares de sensores que aferem grandezas físicas diferentes (Gutmann e Fox, 2002).

Os sensores podem ser categorizados como proprioceptivos ou exteroceptivos (DEBEUNNE E VIVET, 2020). Os primeiros são definidos como aqueles que estimam variáveis internas do sistema, como velocidades lineares e angulares. Já os exteroceptivos fazem observações do robô, à partir do ambiente onde este está inserido, tendo utilidade tanto para localização quanto para mapeamento.

O *encoder* é um sensor que mede a velocidade angular das rodas de um robô, sendo muito aplicado como um sensor proprioceptivo em combinação com outros sensores exteroceptivos (e.g câmera e *laser*) (Lynen et al., 2013). Geralmente, quanto maior a redundância de informações sensoriais, mais precisamente se pode realizar a localização de um robô (Rollinson et al., 2013). Muitas vezes porém, a busca por soluções otimizadas de custo para aquisição, peso e custo energético exige uma configuração instrumental mais simples possível.

A realização de localização utilizando apenas odometria com *encoder* apresenta implementação computacional simples. Porém, há introdução de erros sistemáticos e não-sistemáticos que são cumulativos ao longo da trajetória devido a imprecisões e incertezas do modelo cinemático utilizado (Borenstein e Feng, 1995). Além disso, O *encoder* pode registrar deslocamentos falsos devido ao escorregamento das rodas e irregularidades do terreno. Por outro lado, uma odometria realizada com sensores baseados em *laser* fornecem uma estimativa de pose com alta precisão, mas o seu custo de aquisição e consumo energético tendem a ser altos (Lee et al., 2020). Uma alternativa candidata é o uso da odometria baseada em câmeras e visão computacional, que requer dispositivos de menor custo, porém, esta abordagem necessita de algoritmos mais complexos e custo computacional considerável (Aqel et al., 2016), principalmente quando implementada numa configuração estereoscópica. Um exemplo de aplicação de odometria visual monocular é o trabalho de Bastos et al. (2018) que aplica a técnica em um ambiente virtual com o objetivo de fornecer o trajeto e localização de uma cadeira de rodas.

Uma alternativa de sensoriamento com transdutores de baixo custo, menor esforço computacional e simples instalação em um veículo é o uso de uma *IMU*, sendo capaz de medir diretamente grandezas como aceleração e velocidade angular. Com a tecnologia MEMS (*Microelectromechanical Systems*), acelerômetros e giroscópios se tornaram candidatos convenientes como sensores proprioceptivos em um sistema de localização de robô (Samatas e Pachidis, 2022). Uma *IMU* é composta tipicamente por acelerômetros e giroscópios triaxiais e algumas vezes, magnetômetros, encapsulados todos em um único dispositivo.

A implementação de um sistema de odometria puramente inercial, utilizando uma *IMU* de baixo-custo, representa ainda um grande desafio, especialmente em aplicações robóticas que exijam precisão de localização (Solin et al., 2018). A forma usual da utilização da *IMU* é como um sensor robótico proprioceptivo, integrando numericamente os dados de aceleração e velocidade angular medidos para estimar o seu deslocamento e orientação. Na prática, as medidas fornecidas por tais sensores são contaminadas por sinais elétricos espúrios como tensões de valor médio constante (*bias*) e de variação aleatória, consistindo em ruído branco. Tal contaminação produz erros crescentes de tendência linear para a estimativa de orientação e quadrática para a estimativa de deslocamento, uma vez que para o cômputo do ultimo, é necessário realizar uma integração dupla do sinal de aceleração (Seo e Baek, 2017). Estes erros de estimação divergem em poucos segundos de processo, o que limita a aplicabilidade do sensor.

A utilização de Redes Neurais Artificiais (RNA) em aplicações de robótica móvel têm se mos-

trado promissora para uma grande variedade de contextos (Shabbir e Anwer, 2018). Dessa forma, este trabalho propõe uma método para tornar factível a realização de odometria inercial de um robô móvel aplicando redes neurais em um contexto com restrições instrumentais, utilizando apenas medidas de sensores de baixo custo de torque dos atuadores e de IMU. Além disso, também é utilizado como central de processamento e aquisição de dados em um microcontrolador/system-on-a-chip), enfatizando a possibilidade de aplicação da técnica em um sistema com restrições computacionais (i.e. baixo *clock* e memória RAM). O trabalho de Alajlan e Ibrahim (2022) discute sobre a implementação de redes neurais em sistemas microcontrolados utilizando a tecnologia TinyML, incluindo o microcontrolador, ESP32, utilizado na experimentação desta tese.

Utilizar medidas dos sinais de controle do robô (e.g. torque) responsáveis pela sua dinâmica juntamente com as medidas inerciais, podem permitir encontrar relações causais de entrada e saída na rede neural, apresentando maior robustez na estimativa de movimento devido a complementação de informação frente às contaminações do viés (*bias*) e ruído dos sinais da *IMU*. Essa estratégia é utilizada em assistência pelas redes neurais para a realização da odometria de um robô.

1.1 Objetivo

O objetivo do trabalho é desenvolver um método de estimativa de odometria robótica com o uso de sensores proprioceptivos inerciais do tipo *IMU* e sinais de torque de entrada, aplicando técnicas de aprendizado de máquina. Para a validação do método, deseja-se comparar a precisão relativa do método proposto com a técnica de odometria proprioceptiva baseada em *encoder* e modelo cinemático. A motivação em realizar a comparação apenas entre técnicas proprioceptivas da-se pela proposta de avaliar alternativas de localização de um robô em um contexto onde informações medidas por sensores exteroceptivos são de baixa confiabilidade ou acessibilidade devido ao ambiente em que o robô está inserido, portanto, este trabalho busca explorar uma alternativa de *dead reckoning* do robô focando em sensores inerciais. A técnica busca como alvo aplicações em ambientes *indoor* para veículos robóticos do tipo AGV (Automated guided vehicle) e robôs domésticos (e.g. robôs aspiradores).

1.2 Contribuição Científica

De acordo com as investigações desenvolvidas e descritas no Capítulo 2, abordagens relacionadas a métodos com odometria inercial pura em sistemas com rodas são escassas. Diante disso, este trabalho se propõe a desenvolver um método inovador para realizar odometria inercial pura (sem sensores de medidas absolutas), utilizando uma *IMU* e a medição dos sinais de entrada dos motores de um veículo robótico em um ambiente *indoor*. O método é implementado com o auxílio de Redes Neurais Recorrentes (RNR).

De maneira mais específica, lista-se abaixo as principais contribuições científicas que decorrem do atual trabalho.

1. Uma proposta e avaliação de um método inovador para viabilizar a odometria inercial em robôs móveis utilizando redes neurais recorrentes, combinando os sinais de acelerômetros, giroscópios e torque de entrada, aproximando resultados da abordagem consolidada baseada em *encoder* e modelo cinemático do robô.

2. Estimativa de odometria aplicando um método baseado em redes neurais recorrentes de Elman, de custo computacional consideravelmente menor do que outras arquiteturas modernas como LSTM (*Long Short-Term Memory*) e GRU (*Gated Recurrent Unit*). A arquitetura projetada para a rede não necessita de um treinamento com um *dataset* grande se comparado aos métodos utilizados na literatura, neste trabalho utilizou-se apenas $1.4MB$ de dados para validar a técnica mostrada nesta pesquisa, como apresentado no capítulo de testes experimentais.

3. Verificamos que o algoritmo desenvolvido é viável para implementação em sistemas embarcados com recursos computacionais restritos, como microcontroladores e SoC (*System-on-a-chip*) e utilizando sensores de baixo custo.

4. Criação de um *dataset* de navegação de um robô móvel de pequeno porte, em ambiente *indoor*, contendo dados de *encoder*, acelerações lineares, angulares, corrente dos motores e rastreamento de trajetória baseado em câmera como *ground truth*. Assim, como não foi encontrado um *dataset* prévio a esta tese que contém estas medidas combinadas, é provável que o uso deste banco de dados seja de interesse para outros pesquisadores aplicar novas abordagens.

1.3 Estrutura da Tese

O presente documento está organizado da seguinte maneira. O Capítulo 2 apresenta os principais resultados da revisão bibliográfica feita sob os temas da pesquisa. No Capítulo 3, é descrito a composição de uma *IMU*, explicando o funcionamento do acelerômetro e giroscópio aplicados à odometria inercial, demonstrando os desafios de realizar a estimativa de deslocamento e orientação utilizando estes sensores O Capítulo 4 apresenta as técnicas de RNAs utilizadas para a aplicação de odometria inercial deste trabalho. No Capítulo 5, apresenta-se o modelo cinemático e dinâmico do robô com acionamento diferencial e a arquitetura específica de rede neural utilizada para processar a sua odometria. O capítulo 6 verifica o uso da técnica proposta, aplicando-a em uma simulação numérica de navegação de um robô terrestre de tração diferencial. O Capítulo 7 emprega a técnica proposta em experimentos práticos de estimação de movimento de um robô terrestre. O Capítulo 8 aborda as conclusões da pesquisa e comenta sugestões para trabalhos futuros.

2 Estado da arte

Neste capítulo, algumas referências importantes que utilizaram *IMU* para realizar odometria e localização de sistemas mecânicos, serão analisadas com o intuito de fundamentar a contribuição inovadora desta tese.

M. Li e Mourikis (2013) desenvolveram um sistema de odometria Visual-Inercial implementado em um *smartphone*. A tecnologia permite estimar a pose de um corpo, sendo um humano ou um robô em três dimensões. A técnica utiliza o filtro de Kalman estendido com a *IMU*, como um sensor proprioceptivo, e a câmera, como exteroceptivo. O método, apesar de ter uma performance muito interessante, ainda precisa do suporte do processamento visual dado pela câmera para a realização da odometria. A Tabela 2.1 resume as contribuições e restrições desta pesquisa.

Tabela 2.1: Contribuições e limitações

Prós	Odometria de alta precisão	Utiliza apenas IMU e câmera
Contras	Necessário utilizar câmera para mitigar erros da IMU	-

Fourati (2015) desenvolveu um sistema para rastreamento de pedestre, utilizando uma *IMU* de alto desempenho. A *IMU* está acoplada no pé do pedestre para que seja medido seus parâmetros cinemáticos como orientação, posição, velocidade, aceleração e detecção de passo. O algoritmo de filtragem do movimento do passo inclui métodos para estimação destes parâmetros. Foi implementado um filtro complementar, utilizando os sinais obtidos pelo acelerômetro, giroscópio e magnetômetro para realizar a estimação de orientação. A cada passo realizado, aplica-se um detector de estados baseado nas medidas de aceleração. A detecção de passo é necessário para a aplicação do ZVU (*Zero Velocity Update*), uma heurística que zera os valores de velocidade quando o detector de estados retornar falso (i.e. pé em repouso) e integra a aceleração quando o detector de estados retornar verdadeiro (pé em movimento). Esta pesquisa foi inovadora ao realizar odometria inercial pura, porém, com aplicação apenas em pedestres onde os sensores precisavam ser acoplados. A Tabela 2.2 resume as contribuições e restrições desta pesquisa.

Tabela 2.2: Contribuições e limitações

Prós	Odometria inercial pura
Contras	Aplicação restrita em pedestres

Silva et al. (2015) propuseram uma técnica de atuação em robôs móveis diferenciais para que seja possível a realização da sua odometria inercial pura (utilizando apenas acelerômetro e giroscópio). A técnica é baseada na aplicação de uma aceleração longitudinal periódica com perfil senoidal no robô, ou seja, a plataforma se desloca e repousa ciclicamente devido a velocidade angular dos seus motores se comportar de forma senoidal em consequência das entradas de tensão nos mesmos. Isso permite que o robô emule o procedimento de passo descrito no trabalho proposto por Fourati (2015) e seja possível implementar a heurística de ZVU para que a velocidade e, posteriormente, deslocamento sejam corrigidos, evitando que o erro devido ao *bias*, presente no sinal do acelerômetro, seja propagado em curto prazo. Nos experimentos, foi utilizado um *smartphone Android* com um acelerômetro e giroscópio embutidos. Este foi um dos primeiros trabalhos a realizar odometria inercial pura em veículos com rodas; porém, o fato da aceleração precisar, necessariamente, ter um perfil periódico para emular a condição de *ZVU*, tornava esta técnica inviável para aplicações práticas utilizando controle de trajetória. A Tabela 2.3 resume as contribuições e restrições desta pesquisa.

Tabela 2.3: Contribuições e limitações

Prós	Odometria inercial pura em robô com rodas
Contras	Restrição para movimentos periódicos

Para aprimorar o trabalho desenvolvido em Silva et al. (2015), Lima et al. (2017) adotaram como abordagem a utilização da mesma técnica de movimento periódico, mas aplicando redes neurais para regularização da trajetória, o que melhorou a estimativa da mesma por minimizar os erros numéricos e de calibração derivados da odometria inercial pura. Esta pesquisa aprimorou significativamente os resultados obtidos em Silva et al. (2015) com relação a precisão de estimativa da trajetória. Contudo, este método, além de ter sido sujeito a trajetórias simples, não trata a limitação relacionada à exigência de periodicidade da aceleração, pré-requisito presente no trabalho anterior. A Tabela 2.4 resume as contribuições e restrições desta pesquisa.

Tabela 2.4: Contribuições e limitações

Prós	Odometria inercial pura em robô com rodas
Contras	Restrição para movimentos periódicos

Ferrero et al. (2016) desenvolveram uma abordagem utilizando filtro de Kalman para me-

lhorar a estimativa de deslocamento realizado por acelerômetro nos eixos x,y. Além de resultados modestos em estimativa de deslocamento, o mesmo não se propôs a estimar orientação. A Tabela 2.5 resume as contribuições e restrições desta pesquisa.

Tabela 2.5: Contribuições e limitações

Prós	Estimativa de deslocamento com acelerômetro
Contras	Restrição para movimentos lineares

Solin et al. (2018) implementam odometria inercial pura em plataformas com rodas. Um *Iphone* foi colocado em um carro de bebê conduzido por um ser humano em circuito fechado e fazendo paradas periódicas para aplicar o algoritmo *ZVU*. Foi utilizado uma abordagem de "pseudo" atualização de velocidade, o que limita a velocidade máxima que o algoritmo pode estimar e minimiza a divergência da trajetória calculada. Este método se provou promissor devido a generalização de aplicação em odometria tanto com pedestres quanto com um dispositivo mecânico com rodas, além de obter uma boa precisão. Porém, não foram realizados testes em sistemas robóticos móveis e as trajetórias demostradas para o dispositivo com rodas (carro de bebê) é simples, não contendo curvas mais fechadas nem giro em torno do próprio eixo. A Tabela 2.6 resume as contribuições e restrições desta pesquisa.

Tabela 2.6: Contribuições e limitações

Prós	Odometria inercial pura	Aplicação com *smartphone*
Contras	Validação com trajetória simples	-

Brossard et al. (2020) aplicam *deep learning* com *IEFK* (*Invariant Extended Kalman Filter*) em conjunto com redes neurais *LSTM* para fazer o processamento dos dados de uma *IMU* acoplada em um automóvel para estimação da trajetória realizada pelo mesmo. Esta pesquisa demostrou a aplicação de odometria inercial pura em um automóvel, onde excelentes resultados foram obtidos na estimativa de trajetória realizada. Contudo, o *setup* instrumental (*IMU* utilizada) e computacional são consideravelmente altos. Além da *IMU* ser de alta precisão, a taxa de amostragem dos dados é de $100Hz$ e a rede *LSTM* demora várias horas para ser treinada. A técnica mostrada nesta pesquisa realiza o treinamento das redes neurais em alguns minutos. A Tabela 2.7 resume as contribuições e restrições desta pesquisa.

Tabela 2.7: Contribuições e limitações

Prós	Odometria inercial pura em veículo com rodas
Contras	Instrumental de alto custo

Kim et al. (2021) implementam odometria inercial pura utilizando *LSTM* em um ambiente *indoor*, sendo que o pedestre utiliza um *smartphone*. Esta metodologia apresentou resultados interessantes. No entanto, este trabalho, além de não apresentar aplicações em dispositivo robóticos, valida o método com trajetórias relativamente simples. A Tabela 2.8 resume as contribuições e restrições desta pesquisa.

Tabela 2.8: Contribuições e limitações

Prós	Odometria inercial pura
Contras	Trajetória simples

Os trabalhos descritos acima, selecionados entre aqueles que apresentam temática similar, mostram que ideias e técnicas de uso de odometria inercial continuam a se desenvolver, paralelamente à maior acessibilidade a sensores e novos métodos computacionais de processamento de sinais. Nesse sentido, o presente trabalho também representa uma contribuição relevante a essa categoria de técnicas de odometria robótica utilizando sensores inerciais.

3 Sistemas Inerciais em Odometria

Os sensores inerciais são dispositivos eletrônicos ou mecânicos que medem grandezas cinemáticas de um corpo, como velocidade e aceleração. Estes instrumentos têm sido amplamente utilizados em aeronaves, navios, automóveis, $smartphones$ e robôs móveis (Hazry e Mohd Sofian, 2009; M. Li e Mourikis, 2013).

A recorrência da utilização desses dispositivos tornou-se alta na instrumentação moderna de sistemas mecânicos, sendo necessário o desenvolvimento de uma unidade que encapsula esses sensores de forma prática. Uma *IMU* é o nome que é dado ao encapsulamento de sensores inerciais contendo acelerômetros e giroscópios integrados e algumas vezes, magnetômetros.

Com o advento da tecnologia MEMS (*Micro-Electro-Mechanical-System*), a implementação de uma *IMU* diminuiu drasticamente o custo de fabricação e consumo de potência elétrica (Gui et al., 2015). Esta tecnologia é baseada na implementação de sistemas mecânicos miniaturizados utilizando *wafers* de silício em circuitos integrados (Zotov et al., 2010).

Esse capítulo apresenta o modelo de medição para o acelerômetro e giroscópio de tecnologia MEMS.

3.1 Acelerômetro

Acelerômetro é um dispositivo capaz de mensurar a aceleração linear em relação a um ou mais eixos no seu sistema de referência. Um acelerômetro típico é composto por três eixos de medição, como mostrado na Figura 3.1. O modelo de um acelerômetro triaxial pode ser representado pela Equação 3.1.

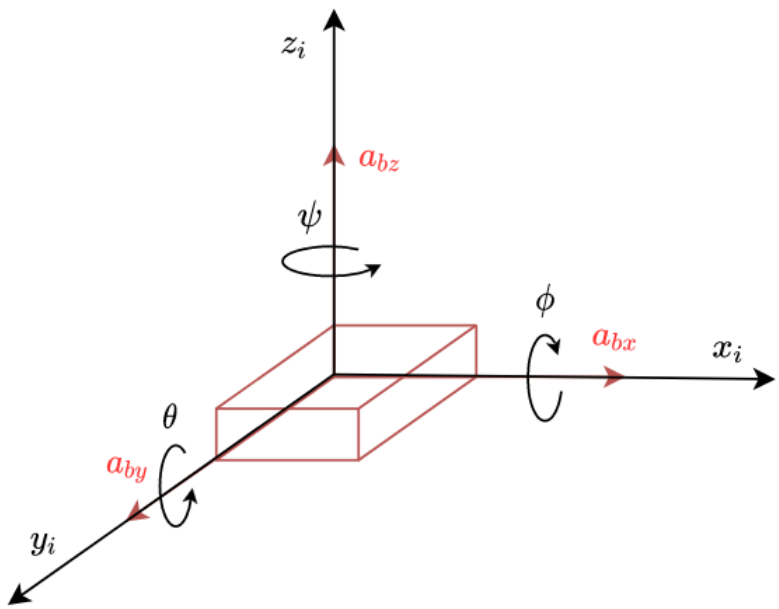

Figura 3.1: Sistema de coordenadas do acelerômetro

$$\vec{a}_m = S_a.T_a.\vec{a}_b - R_I^b.\vec{g} + \vec{b}_a(t,T) + \vec{n}_a(t) \tag{3.1}$$

onde:

- $\vec{a}_m$ é o vetor de aceleração medida pelo sensor no referencial móvel.
- S_a é a matriz diagonal de sensibilidade referente a escala de medição em cada eixo.
- T_a é a matriz de desalinhamento entre o sistema de coordenadas ortogonal do sensor e o sistema de coordenadas de medição, sendo α, β e γ os ângulos de rotação entre os sistemas.
- $\vec{a}_b = [a_{bx}\ a_{by}\ a_{bz}]^T$ é o vetor de aceleração do referencial móvel.
- R_I^b é a matriz de rotação que transforma o vetor de gravidade $\vec{g}$ do referencial inercial i para o referencial móvel b.
- $\vec{b}_a(t,T)$ é o vetor de cada componente do *bias* na medição em cada eixo. Este valor muda lentamente em função da temperatura T.

- $\vec{n}_a(t)$ é o vetor de cada componente do ruído presente na medição em cada eixo.

As Equações 3.2 e 3.3 descrevem as matrizes S_a, T_a, respectivamente, no qual podem ser identificadas por um método de calibração, como realizado por Delgado et al. (2016), que demonstra o processo de calibração para acelerômetro e giroscópio. A matriz de rotação R_I é definida como $R_I^b = R_\psi.R_\theta.R_\phi$, demostrada nas Equações 3.4, 3.5 e 3.6.

$$S_a = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & S_z \end{bmatrix} \tag{3.2}$$

$$T_a = \begin{bmatrix} 1 & 0 & 0 \\ cos(\alpha) & 1 & 0 \\ cos(\beta) & cos(\gamma) & 1 \end{bmatrix} \tag{3.3}$$

$$R_\psi = \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

$$R_\theta = \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix} \tag{3.5}$$

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & sin(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix} \tag{3.6}$$

3.2 Giroscópio

Giroscópio é um dispositivo capaz de mensurar a velocidade angular de um corpo em relação a um ou mais eixos referenciais. Este sensor é tipicamente composto por três eixos de medição, como mostrado na Figura 3.2. O modelo de um giroscópio pode ser representado pela Equação 3.7.

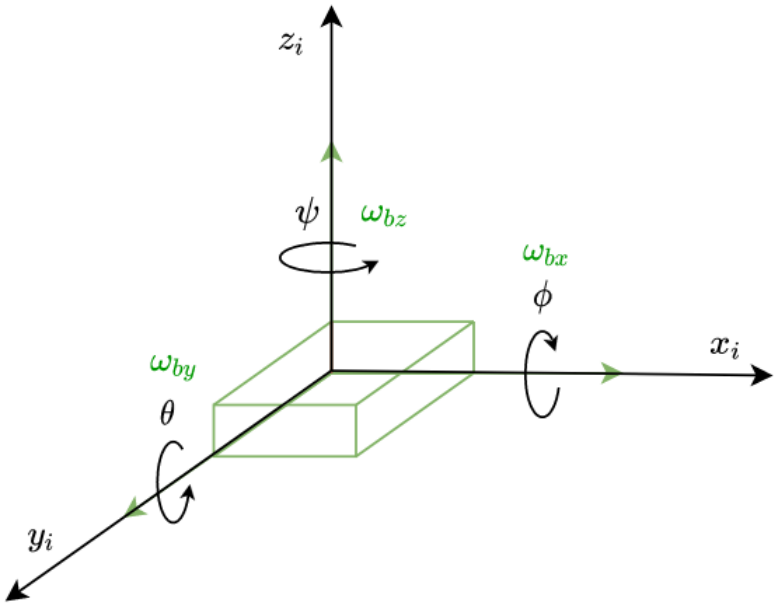

Figura 3.2: Sistema de coordenadas do giroscópio

$$\vec{\omega_m} = S_g.T_g.\vec{\omega_b} + \vec{b_g}(t,T) + \vec{n_g}(t) \tag{3.7}$$

onde:

- $\vec{\omega_m}$ é o vetor de velocidade angular medida pelo sensor no referencial móvel.
- S_g a matriz diagonal de sensibilidade referente a escala de medição em cada eixo. Essa matriz é análoga à Equação 3.2
- T_g a matriz de desalinhamento entre o sistema de coordenadas ortogonal do sensor e o sistema de coordenadas de medição. Essa matriz é análoga à Equação 3.3

- $\vec{\omega}_b$ é o vetor $[\omega_{bx}\ \omega_{by}\ \omega_{bz}]^T$ de velocidade angular do referencial móvel.
- $\vec{b}_g(t,T)$ é o vetor de cada componente do *bias* na medição em cada eixo. Assim como $\vec{b_a}(t,T)$, aquele valor muda lentamente em função da temperatura T.
- $\vec{n}_g(t)$ é o vetor de cada componente do ruído presente na medição em cada eixo.

3.3 Odometria Inercial

Esta seção apresenta como as medidas do acelerômetro e giroscópio podem ser utilizadas para implementação de odometria inercial. Além disso, também será apresentado como pode ser realizada a estimativa de orientação de um corpo através das medidas de velocidade angular do giroscópio, a estimativa de deslocamento de um corpo através das medidas de aceleração linear do acelerômetro e, por último, o cômputo da trajetória realizada utilizando ambas as estimativas em composição.

3.3.1 Estimativa de Orientação

Ao considerar a medição de velocidade angular ω_m em torno de um dos eixos de um giroscópio ideal, pode-se calcular o deslocamento angular integrando esse sinal medido, como mostrado na Equação 3.8.

$$\theta_g = \int_0^t \omega_m dt, \tag{3.8}$$

onde θ_g é o ângulo computado ao integrar a velocidade angular ω_m ao longo do tempo t. Porém, como mostrado na Equação 3.7, a medida do giroscópio está sujeita a várias perturbações que alteram a medida real de velocidade angular. Supondo que as variáveis T_g e S_g, podem ser omitidas no seguinte modelo após um processo de calibração (Delgado et al., 2016), a estimativa de orientação pode ser calculada pela Equação 3.9. Esta omissão implica em aproximar as contribuições de T_g e S_g por uma matriz identidade.

$$\theta_g = \int_0^t (\omega_b + b_g(t,T) + n_g(t))dt. \tag{3.9}$$

Ao levar em consideração que o termo do *bias* $b_g(t,T)$ varia lentamente em função da temperatura (Aggarwal et al., 2008), ele é modelado como uma constante b_g. Além disso, a perturbação $n(t)$ é representada como um ruído branco $n(0,\sigma)$, onde σ é o desvio padrão com distribuição gaussiana. Com essas considerações, a Equação 3.9 resulta em 3.10.

$$\theta_g = \int_0^t \omega_g dt + b_g.t + \int_0^t n(0,\sigma).dt. \tag{3.10}$$

A Equação 3.10 mostra que o termo $b_g.t$ gera a propagação de erro linear ao longo do tempo na estimativa do ângulo, enquanto o ruído n idealmente tem pouca contribuição no erro devido a sua média ser zero. Portanto, quanto menor for o valor de b_g, melhor será a estimativa de ângulo em cada eixo do giroscópio.

Para que a orientação do dispositivo seja estimada é necessário que os três ângulos do sensor ϕ, θ e ψ sejam calculados em torno de três eixos ortogonais x, y e z do referencial inercial. A partir de uma matriz de rotação $R_{\phi\theta\psi}$ definida na Equação 3.11, pode-se obter a orientação a partir dos ângulos de *Euler*.

$$R_{\phi\theta\psi} = R_\phi R_\theta R_\psi = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}, \tag{3.11}$$

sendo $\mathbf{R}_\phi$, $\mathbf{R}_\theta$ e $\mathbf{R}_\psi$ as matrizes de rotação em torno dos três eixos ortogonais definidas em 3.4, 3.5 e 3.6. Assim, é possível extrair os ângulos de *Euler* para determinar a orientação do sensor pelas Equações 3.12, 3.13, 3.14 (Slabaugh, 1999).

$$\phi = arctg(\frac{R_{32}}{R_{33}}), \tag{3.12}$$

$$\theta = -sin(R_{31}), \tag{3.13}$$

$$\psi = atan2(\frac{R_{21}}{cos(\theta)}, \frac{R_{11}}{cos(\theta)}). \tag{3.14}$$

3.3.2 Estimativa de Deslocamento

Um acelerômetro triaxial permite realizar a medida de aceleração ao longo de cada eixo individualmente. Assumindo um acelerômetro ideal em uma aplicação de estimativa de deslocamento de um referencial móvel ao longo de um eixo, pode-se calcular sua posição $P(t)$ ao longo do tempo integrando duas vezes a medida de aceleração linear A_l.

$$P(t) = \int_0^t \int_0^t A_l(t)dtdt \tag{3.15}$$

Porém, como mostrado na Equação 3.1, assim como o giroscópio, a medida do acelerômetro está sujeita a várias perturbações que alteram a medida real de aceleração linear. Assim como feito na Equação 3.9, supondo que as matrizes T_a e S_a podem ser omitidas no seguinte modelo após um processo de calibração, a aceleração medida pelo sensor é definida como:

$$\vec{a_m} = \vec{a_b} - R_I^b.\vec{g} + \vec{b_a}(t,T) + \vec{n_a}(t) \tag{3.16}$$

Para calcular a velocidade e o deslocamento do sensor em relação ao referencial inercial é necessário subtrair a componente do vetor de gravidade $\vec{g}$ da Equação 3.16. Multiplicando em ambos os lados da equação pela matriz de rotação R_b^I, gera-se a Equação 3.17, no qual retira a contribuição dada pelo vetor de gravidade $\vec{g}$ na medida do sensor, onde a_b torna-se a_I, sendo portanto a aceleração linear realizada pelo corpo em relação ao referencial inercial.

$$\vec{a}_I = R_b^I(\vec{a}_m + \vec{b}_a(t,T) + \vec{n}_a(t)) + \vec{g} \tag{3.17}$$

Supondo a medida de aceleração sem a componente da gravidade, integra-se a_I para obter as velocidades lineares no plano.

$$\vec{V_I} = \int_0^t (\vec{a}_I + \vec{b}_a(t,T) + \vec{n}(t))dt \tag{3.18}$$

Considerando $\vec{b}_a(t,T)$ uma função que varia lentamente com a temperatura T como uma constante b_a e n um ruído branco de distribuição normal com média zero, tem-se :

$$\vec{V_I}(t) = \int_0^t \vec{a}_I dt + \vec{b}_a.t \tag{3.19}$$

Assim como no giroscópio, nota-se que o *bias* $\vec{b_a}$ gera um desvio linear ao longo do tempo na estimativa de velocidade. Para o cálculo do deslocamento do sensor em relação ao referencial inercial I, integra-se a Equação 3.19, obtendo-se :

$$\vec{P_I}(t) = \int_0^t \int_0^t \vec{a}_I dtdt + \vec{b}_a.t^2 \tag{3.20}$$

A Equação 3.20 mostra como a componente do *bias* $\vec{b}_a$ faz com que a estimativa de posição $\vec{P_I}$ divirja quadraticamente ao longo do tempo.

O acumulo de erros gerados por b_a e b_g na estimação de deslocamento e orientação via integração numérica dos dados do acelerômetro e giroscópio, respectivamente, tornam inviáveis a realização de odometria inercial de longo prazo, onde pequenos valores de *bias* nos sensores causam grandes erros de integração, em especial nos cálculos de deslocamento lineares.

Na próxima seção, será apresentado como computar a trajetória de um robô utilizando o processamento das medidas dos sensores mostrados nesta seção.

3.3.3 Estimativa de Trajetória via Odometria Inercial

A trajetória realizada pelo robô pode ser calculada recursivamente a partir dos incrementos de deslocamento e orientação, conforme descritos na Equação 3.21.

$$\begin{bmatrix} X(k) \\ Y(k) \\ Z(k) \end{bmatrix} = \begin{bmatrix} X(k-1) \\ Y(k-1) \\ Z(k-1) \end{bmatrix} + \prod_{k=1}^{n} R(k) \begin{bmatrix} \delta_x(k) \\ \delta_y(k) \\ \delta_z(k) \end{bmatrix} \tag{3.21}$$

onde $\delta_x(k)$, $\delta_y(k)$, $\delta_z(k)$ são os incrementos de deslocamento no espaço calculados pela Equação 3.20 e $R(k)$ é a matriz de rotação calculada pela Equação 3.11, onde k é o instante discreto de computação.

No entanto, como foi visto nos modelos dos sensores da *IMU* onde os valores espúrios de *bias* e ruido estão presentes, esta recursão baseada nas estimativas incrementais de deslocamento e rotação é sujeita a propagação de erros ao longo do tempo. Isso torna desafiadora a estimação precisa de odometria de um robô utilizando unicamente os sensores proprioceptivos inerciais. Esta tese propõe uma forma alternativa de utilizar medidas de uma *IMU* complementada pelas medidas de torque dos atuadores para viabilizar a implementação de odometria inercial em um robô móvel utilizando redes neurais de baixo custo computacional.

O próximo capítulo apresenta os fundamentos para a implementação do método.

4 Fundamentos de Redes Neurais Artificiais

Um dos pilares deste trabalho é a utilização de técnicas de aprendizado de máquina para assistir à odometria baseada em *IMU* e sinais de torque de entrada em robô móvel. Dentre tais técnicas, as *RNAs* têm se tornado cada vez mais populares nos últimos anos.

Este capítulo faz uma introdução dos conceitos que fundamentam o funcionamento das redes neurais utilizadas nesta pesquisa.

4.1 Redes Neurais Artificiais

RNAs são sistemas de processamento de informação baseados no modelo de neurônios biológicos implementadas por *software* ou um circuito eletrônico, capazes de classificar dados e realizar previsões. As conexões entre os neurônios que compõe uma *RNA* são associadas a pesos sinápticos. Os pesos são parâmetros ajustados através de um processo de treinamento da rede neural para desempenhar tarefas específicas. Uma rede adequadamente ajustada, pode ser empregada para resolver problemas genéricos da mesma natureza dos dados no qual foi treinada

A representação matemática proposta por Rosenblatt em 1958 de um neurônio do tipo *perceptron* é descrita na equação 4.1.

$$y = f(\sum_{i=1}^{n} w_i x_i + b), \tag{4.1}$$

onde y é a saída do neurônio, w é o peso sináptico i, x é a entrada i e b é o *bias*. A função f é chamada de função de ativação (e.g. sigmoide, tangente hiperbólica e função linear). Alguns tipos de função de ativação são mostradas em seguida.

$$F(u) = \frac{1}{1 + e^{-u}}, \tag{4.2}$$

$$F(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}, \tag{4.3}$$

$$F(u) = u. \tag{4.4}$$

Redes neurais com várias camadas são chamadas de MLP (*Multi-layer perceptron*), no qual diversas camadas de neurônios intermediários, chamadas camadas ocultas, podem ser implementadas em cascata. A representação matemática em forma matricial de uma rede neural com duas camadas ocultas é descrita na equação 4.5.

$$\mathbf{Y} = f_2(\mathbf{W}_2 f_1(\mathbf{W}_1 \vec{x} + b_1) + b_2) \tag{4.5}$$

onde $\mathbf{W}_1$ e $\mathbf{W}_2$ são as matrizes de pesos da primeira e da segunda camada, $\vec{x}$ é o vetor de entradas na rede, f_1 e f_2. as funções de ativação das camadas 1 e 2 e b_1 e b_2, o viés de cada camada.

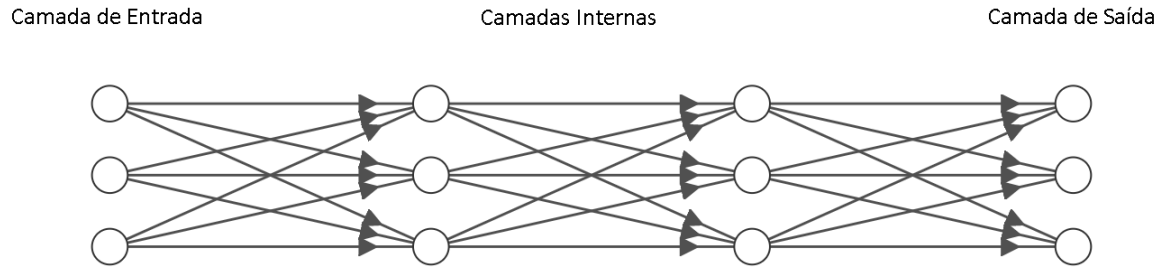

Figura 4.1: Rede Neural Perceptron Multicamadas

As redes MLP são um dos métodos mais populares na literatura para a resolução de problemas de classificação de dados (Driss et al., 2017), porém, elas não tem a mesma robustez em problemas relacionados a séries temporais como as redes neurais recorrentes (Khaldi et al., 2018). Além disso, seu desempenho de treinamento depende da qualidade do *dataset*, sendo pouco robusto para treinamento com sinais ruidosos (Kordos e Rusiecki, 2016). O próximo tópico apresentará uma arquitetura de *RNA* mais adequada para a aplicação neste trabalho.

4.2 Redes Neurais Recorrentes

A arquitetura de rede neural recorrente (RNR) é composta de laços de retroalimentação que armazenam a informação da ordem temporal das sequências de entrada, sendo portanto, conveniente para atuar em problemas com dados sequenciais e séries temporais como processamento de fala e identificação de sistemas dinâmicos (Sak et al., 2015; Richard et al., 2019).

Uma rede neural recorrente é formada de uma ou mais unidades de recorrência. Entre as unidades mais populares pode-se citar a célula de Elman, célula LSTM e GRU (Hewamalage et al., 2021). As equações 4.6 e 4.7 modelam a rede de Elman.

$$h_t = \sigma_h(\mathbf{W}_h \vec{x_t} + \mathbf{U_h} \vec{h_{t-1}} + b_h), \tag{4.6}$$

$$y_t = \sigma_y(\mathbf{W_y} \vec{h_t} + b_y), \tag{4.7}$$

onde $\vec{h_t}$ é o vetor da camada escondida, $\vec{x_t}$ é o vetor de entrada, σ_h e σ_t são as funções de ativação, $\mathbf{W}_h$, $\mathbf{U}_{h-1}$ e $\mathbf{b}_h$ são as matrizes de pesos. Redes neurais recorrentes, assim como as perceptron, também podem ter multicamadas, como apresentado na Figura 4.2.

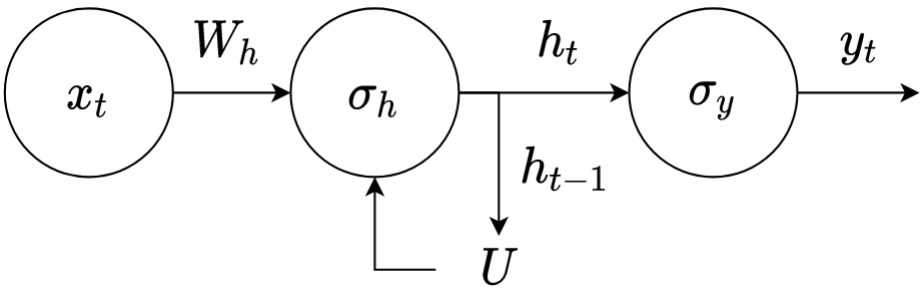

Figura 4.2: Rede de Elman

4.3 Treinamento de Redes Neurais via Retropropagação

Este tópico introduz o algoritmo de retropropagação do erro (*Backpropagation*), um dos métodos mais populares de treinamento de redes neurais. A técnica abordada aqui é conceitualmente apresentada em uma rede de uma camada; porém, extensões desse método podem ser aplicadas para n camadas, incluindo para redes neurais recorrentes, em cujo caso, se pode utilizar a variação chamada BPTT (*Backpropagation Through Time*). A Equação 4.8 descreve a saída de um neurônio n, dadas as entradas $x(n)$ e seus respectivos pesos w_{ij}, bem como a função de ativação, descrita na Equação 4.9. O erro da saída da rede é definido como $d_i - \mathbf{w}^T x$, levando à função custo, Equação 4.10, que quantifica a diferença entre a saída da rede $y(n)$ e a saída correta esperada d_i, onde $v_(n)$ é a saída de uma camada da rede a priori da aplicação da função de ativação.

$$v(n) = \sum_{i=0}^{m} w_{ij}(n).x(n), \tag{4.8}$$

$$y_j(n) = \sigma_j(v_j(n)), \tag{4.9}$$

$$\gamma = \sum_{i=1}^{n} (d_i - \mathbf{w}^{\intercal}.\mathbf{x}_i). \tag{4.10}$$

Portanto, os pesos sinápticos **w** devem ser ajustados de forma que a diferença entre a entrada e saída seja minimizado.

Diante do que foi apresentado, este capítulo teve o objetivo de compilar as principais formulações das arquiteturas de *RNAs* mais utilizadas, considerando a resolução do problema de odometria inercial proposto nessa tese. Isto posto, o capítulo seguinte detalha como ocorre a integração entre essa ferramenta computacional e a abordagem proposta.

5 Método de Odometria Inercial via Redes Neurais

Este capítulo detalha a proposta desta tese para viabilizar a utilização das medidas do *IMU* e dos atuadores de um robô, para implementar uma odometria inercial aplicando *RNAs*. Primeiramente, será mostrado o modelo cinético do robô e, em seguida, o detalhamento das arquiteturas das redes utilizadas.

5.1 Modelo Dinâmico de um Robô Diferencial

Neste trabalho, utiliza-se um robô móvel de configuração diferencial em um contexto de navegação em ambiente *indoor*. A Figura 5.1 mostra um típico robô móvel com duas rodas motorizadas e uma do tipo castor, onde:

- z é o centro de rotação.
- b é a distância entre as rodas.
- C é o centro de massa.
- r é raio das rodas.
- V_l é o vetor de velocidade linear no eixo longitudinal do robô.
- θ_z é ângulo de orientação do robô.
- d é a posição do centro de rotação robô em relação ao referencial inercial.
- ω_l é a velocidade angular da roda esquerda
- ω_r é a velocidade angular da roda direita

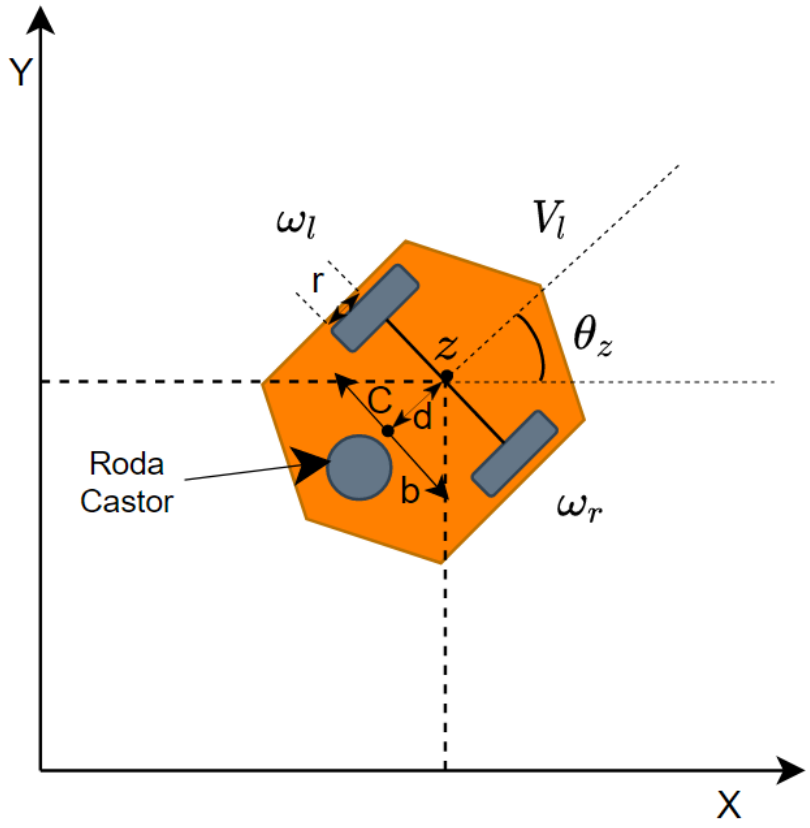

Figura 5.1: Dependências geométricas do robô diferencial

Os componentes da velocidade do ponto z podem ser expressos em termos de velocidade angular da roda esquerda (ω_l) e direita (ω_r), tais como:

$$\dot{x_a} = \frac{(\omega_r + \omega_l)cos(\theta)}{2} \tag{5.1}$$

$$\dot{y_a} = \frac{(\omega_r + \omega_l)sin(\theta)}{2} \tag{5.2}$$

$$\dot{\theta}_z = \frac{(\omega_r - \omega_l)}{b} \tag{5.3}$$

onde $\dot{x}_a$ representa a velocidade do ponto z no eixo x, $\dot{y}_a$ representa a velocidade do ponto z no eixo y, $\dot{\theta}_z$ a velocidade angular do robô em torno do eixo z. Com isso, o modelo dinâmico do robô deduzido em Ivanjko et al. (2010) pode ser expresso a partir das expressões a seguir:

$$A\ddot{\theta}_r + B\ddot{\theta}_l = T_r - K\dot{\theta}_r, \tag{5.4}$$

$$B\ddot{\theta}_r + A\ddot{\theta}_l = T_l - K\dot{\theta}_l, \tag{5.5}$$

$$A = \frac{mr^2}{4} + \frac{(I_z + md^2)r^2}{b^2} + I_o, \tag{5.6}$$

$$B = \frac{mr^2}{4} - \frac{(I_z + md^2)r^2}{b^2}, \tag{5.7}$$

onde T_l e T_r são os respectivos torques do motor esquerdo e direito e K é a constante de atrito dinâmico. A massa do robô é m, o momento de inércia do robô no eixo z é I_z, o momento de inércia resultante do rotor com a roda é I_o e a distância entre o ponto z e C é definida como d. O modelo dinâmico apresentado nas Equações 5.4 e 5.5 são utilizados para simular o robô.

Considerando os atuadores do robô como motores de corrente contínua de imã permanente, tem-se as equações que definem o modelo elétrico em 5.8 e mecânico em 5.9. As relações eletro-mecânicas são mostradas nas equações 5.10 e 5.11.

$$v_a(t) = R_a.i_a(t) + L.\frac{di_a}{dt} + v_b(t) \tag{5.8}$$

$$T(t) = J_m.\frac{d\omega_m(t)}{dt} + B_m.\omega_m(t) \tag{5.9}$$

$$V_b(t) = K_e.\omega_m(t) \tag{5.10}$$

$$T(t) = K_t.i_a \tag{5.11}$$

Onde v_a é a tensão de entrada, R_a a resistência do enrolamento, L a indutância do enrolamento, i_a a corrente que flui pelo motor, v_b a força contra-eletromotriz, K_e a constante contra-eletromotriz, ω_m a velocidade angular do eixo do motor, J_m a inercia do rotor, B_m o atrito viscoso do motor, $T(t)$ o torque gerado pelo motor. A equação 5.11 mostra K_t, a constante que relaciona a corrente do motor com o torque gerado. Aplicando a transformada de Laplace nas equações 5.8, 5.9, 5.10 e 5.11, obtém-se :

$$I_a(s) = \left[\frac{1}{Ls + R_a}\right].[V_a(s) - V_b(s)] \tag{5.12}$$

$$\Omega_m(s) = \left[\frac{1}{J_m s + B_m}\right].T(s) \tag{5.13}$$

$$V_b(s) = K_e.\omega_m(s) \tag{5.14}$$

$$T(s) = K_t.I_a(s) \tag{5.15}$$

Conectando os modelos elétricos e mecânicos no domínio da frequência mostrados de 5.12 até 5.15, forma-se o diagrama de blocos na Figura 5.2, onde $T_d(s) = r.m$ é o torque de distúrbio causado pela massa do robô no eixo rotativo. Portando para a aplicação do método proposto, é necessário medir a corrente i_a para estimar o torque dinâmico T.

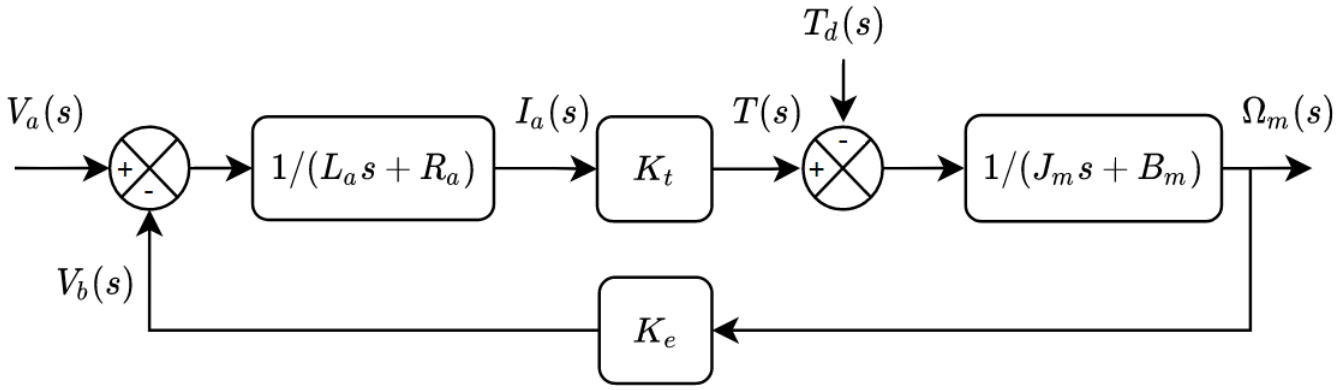

Figura 5.2: Diagrama de blocos do motor CC

5.2 Arquitetura de Redes Neurais Recorrentes

A partir do modelo cinemático do robô e das medidas de suas velocidades lineares e angulares, é possível computar a sua trajetória realizada num plano. Diante desta perspectiva, decidiu-se construir uma *RNR* independente para cada uma dessas variáveis.

Para representar melhor essa abordagem, a Figura 5.3 mostra o diagrama de blocos da arquitetura dessas redes neurais, onde T_r e T_l são os torques aplicados pelo motor direito e esquerdo, A_{xm} e A_{ym} são acelerações longitudinais e laterais, ω_{zm} é a velocidade angular medida em torno do eixo z, ω_z é a velocidade angular estimada em torno do eixo z e V_l é a velocidade linear estimada no eixo longitudinal x.

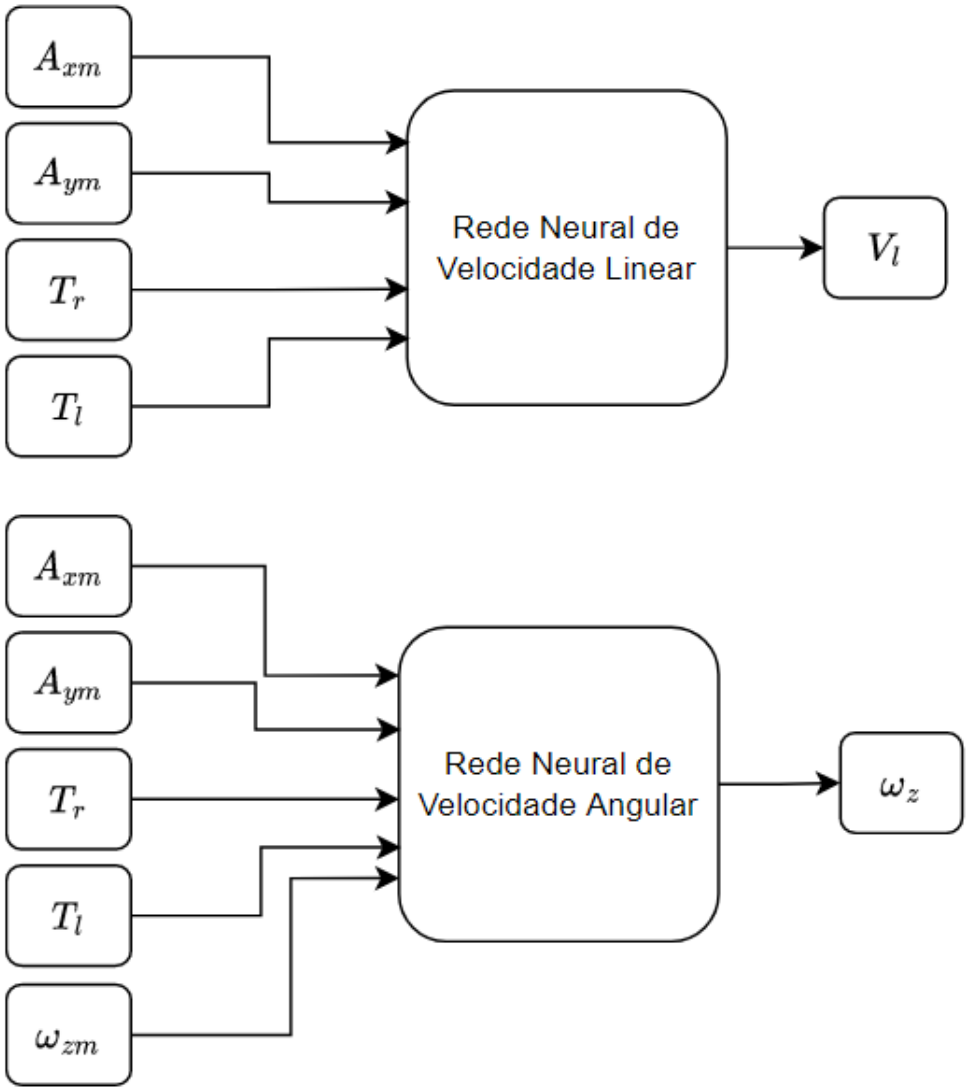

Figura 5.3: Arquitetura das *RNRs* utilizadas

A arquitetura utilizada para cada estimador é definida por uma *RNR*. Esta arquitetura é conveniente para modelar sistemas com dinâmicas não-lineares e estimar suas variáveis de saída com capacidades semelhantes de um observador de estados da teoria de controle moderno (Al Seyab e Cao, 2008). Além disso, a arquitetura é robusta a ruídos no sinal de entrada (Su e McAvoy, 1997). Essas características fazem com que as *RNRs* sejam mais adequadas para esta aplicação de robótica que envolve identificação de um sistema dinâmico, se comparadas com redes estáticas como as do tipo *feedforward* (Toha e Tokhi, 2008).

Um dos objetivos desta pesquisa é facilitar a implementação dos estimadores em sistemas embarcados com restrições de processamento e memória, como em microcontroladores. Redes recorrentes, do tipo *LSTM*, *GRU* e suas variações, são arquiteturas que têm mostrado resultados promissores na literatura em diversas aplicações; porém, o seu custo computacional é consideravelmente alto devido à sua complexidade (Rezk et al., 2020). Para que uma rede *LSTM* tenha um desempenho satisfatório em um sistema eletrônico embarcado, por exemplo, é necessário utilizar atualmente dispositivos baseados em tecnologia ASIC (Circuitos Integrados de Aplicação Específica), e FPGA (Portas Programáveis em Campo) (S. Li et al., 2015).

Para favorecer a possibilidade de implementação do método proposto em sistemas embarcados, com menores recursos computacionais e sensoriais, faz-se necessário explorar as capacidades de redes recorrentes mais simples, como as de Elman, que apresenta bons resultados na identificação de sistemas dinâmicos (Ibraheem, 2017; Zhang et al., 2010). Portanto, ambas as redes apresentadas na Figura 5.3 usam esta estrutura clássica de redes neurais, visto que a mesma atende as necessidades de processamento desejadas.

A odometria de robôs baseada na medição de velocidade angular das rodas junto com um modelo cinemático é consolidada em sistemas de localização em robótica móvel. Porém, o mesmo sofre de erros sistemáticos e não-sistemáticos, sendo que os erros sistemáticos se entendem como aqueles associados a incertezas no modelo cinemático do robô, enquanto erros não-sistemáticos são os causados tanto pelo escorregamento das rodas quanto por irregularidades no solo (Borenstein e Feng, 1995).

Como a abordagem, aqui apresentada, utiliza medidas inerciais baseadas na aceleração e velocidade angular via *IMU* solidária ao robô, e no torque dos motores, através da medida de suas correntes de acionamento, a técnica proposta se torna mais robusta a aquelas duas fontes de erro, visto que as medidas de acionamento independem do modelo cinemático e não computam falsos positivos devido à escorregamento das rodas como acontecem com sensores contadores de revolução (e.g. *encoder*).

Ao considerar que as variáveis estimadas $V_l = V_t + e_l(\mu,\sigma)$ e $\omega_z = \omega_t + e_z(\mu,\sigma)$, onde V_t e ω_t são a velocidade linear e angular de cada *target* de saída e e_l e e_z são os erros de estimação para as redes de velocidade linear e angular, respectivamente, as equações 3.20 e 3.10 podem ser reescritas da seguinte maneira:

$$\vec{P}_I(t) = \int_0^t (\vec{V}_l + \vec{e}_l(\mu,\sigma))dt, \tag{5.16}$$

$$\theta_z = \int_0^t (\omega_z + e_z(\mu,\sigma))dt. \tag{5.17}$$

As Equações 5.16 e 5.17 mostram que as estimativas de deslocamento e ângulo não são mais influenciadas diretamente pela integração do *bias*, que varia com a temperatura, e nem do ruído de medição; mas, sim, apenas pelos erros de estimação na saída de cada rede expressos por e_l e e_z. Com isso, a divergência de estimação de deslocamento, com tendência quadrática, e de ângulo, com tendência linear, é substituída por um fator dependente de uma distribuição de probabilidade do erro da rede neural, no qual tem um aspecto aproximado de uma gaussiana (Asoodeh et al., 2015; Tekin et al., 2014).

A partir dos deslocamentos, $\delta x(k)$, e ângulos incrementais, $\delta\theta_z(k)$, pode-se computar a trajetória do robô no plano X,Y, conforme a Equação 5.18, a seguir.

$$\begin{bmatrix} X(k) \\ Y(k) \end{bmatrix} = \begin{bmatrix} X(k-1) \\ Y(k-1) \end{bmatrix} + \begin{bmatrix} \delta_x(k).cos(\delta\theta_z(k)) \\ \delta_x(k).sin(\delta\theta_z(k)) \end{bmatrix} \tag{5.18}$$

Com estas fundamentações conceituais e matemáticas apresentadas, os testes e experimentos de validação da técnica proposta serão apresentados e discutidos nos próximos capítulos.

6 Verificação Numérica da Técnica Proposta

Neste capítulo, será mostrada a etapa de simulação numérica para validação da técnica proposta que associa o método de odometria inercial às *RNRs*, apresentado no capítulo anterior.

6.1 Ambiente de Simulação

As simulações do robô e a aplicação das *RNRs* para odometria foram realizadas utilizando a linguagem de *script* do MATLAB, no qual o modelo dinâmico apresentado no Capítulo 5 de um robô diferencial foi implementado. Além disso, para construção e treinamento das redes neurais, foi utilizado o *Neural Networks Toolbox*. Na Figura 6.1, apresenta-se o diagrama de blocos do modelo do robô.

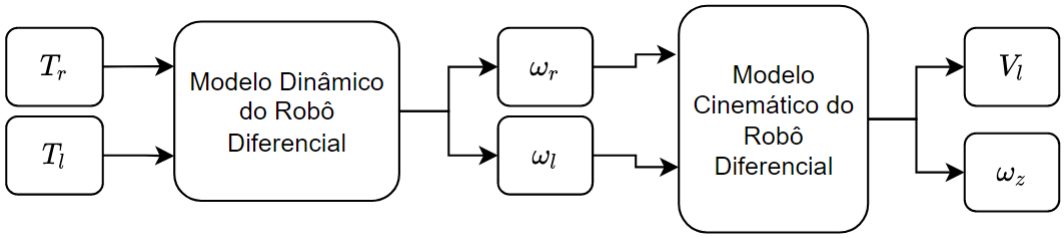

Figura 6.1: Diagrama de blocos do modelo do robô

onde T_l e T_r são os torques do motor esquerdo e do motor direito e ω_l e ω_r são a velocidade angular da roda esquerda e da roda direita.

Ao utilizar o modelo cinemático do robô diferencial mostrado nas Equações 5.1, 5.2 e 5.3, é possível computar a velocidade linear V_l do veículo em relação ao seu eixo longitudinal e a velocidade angular ω_z em torno do seu eixo de rotação z. Assim, ao derivar V_l, a aceleração linear A_l realizada pelo robô é obtida, sendo que, a partir deste ponto, é possível implementar os modelos utilizados para o acelerômetro e giroscópio, apresentados nas Equações 3.1 e 3.7, que foram aplicados para representar as medidas dos sensores inerciais do robô no ambiente simulado.

A partir destas variáveis, foram construídas duas *RNRs*, sendo que a primeira estima a velocidade linear e outra estima a velocidade angular, conforme discutido no Capítulo 5.

6.2 Treinamento das *RNRs* Aplicadas ao Robô Diferencial para Odometria Inercial

Para realizar a simulação da aplicação do método proposto, foram aplicados sinais de torque na roda direita e esquerda no modelo dinâmico do robô, obtendo como saída as velocidades linear e angular. Tais saídas do modelo cinético do robô constituem o padrão, ou *ground truth*, a partir do qual se constroem os sinais sintéticos de medições da *IMU*, atribuindo valores específicos de tendência (*bias*) e ruído gaussiano. A posteriori, esses dados são inseridos nas redes da arquitetura mostrada na Figura 6.2 para realizar o treinamento.

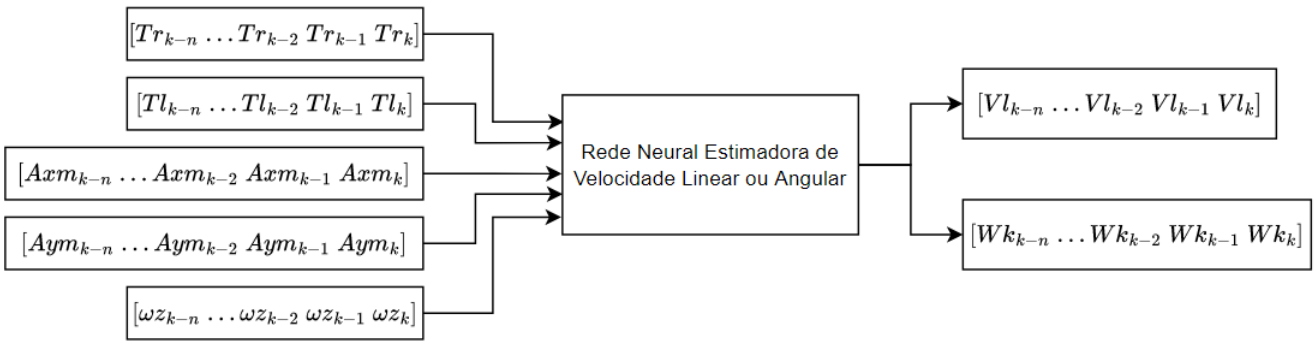

Figura 6.2: Organização dos dados para treinamento das redes

Os valores de *bias* utilizados na contaminação dos sinais tanto do acelerômetro quanto do giroscópio virtuais são, respectivamente, $0.2m/s^2$ e $0.1rad/s$. Estes valores foram considerados baseados nas medidas encontradas por Tereshkov (2013). Tais valores foram adotados por se tratarem de grandezas típicas das incertezas elétricas associadas aos ganhos dos acelerômetros e giroscópios da unidade micro-eletromecânico de sensoriamento inercial do robô utilizado no teste experimental descrito no Capítulo 7. Os seguintes parâmetros cinemáticos e dinâmicos foram adotados no modelo simulado do robô de acionamento diferencial (vide Figura 5.1): $r = 0.0325m$, $L = 0.185m$, $I_o = 0.1kgm^2$, $I_z = 0.1kgm^2$, $K = 0.4m^2/s$, $m = 0.5kg$ e $d = 0.1m$.

No treinamento de cada rede, utiliza-se, como vetores de entrada, os sinais inerciais medidos pelo modelo da *IMU* e os torques aplicados pelos atuadores. Como saída, no treinamento, utiliza-se a velocidade linear e angular do robô geradas pelo modelo dinâmico e cinemático. A Figura 6.3 mostra os torques do motor esquerdo e direito aplicados para a realização do treinamento.

No tocante à definição de alguns parâmetros da rede, sabe-se que a escolha do numero de neurônios e camadas de uma rede neural usualmente passa por um processo de teste e falha, até se encontrar uma arquitetura que cumpra os objetivos desejados (Sheela e Deepa, 2013).

Neste trabalho, utilizou-se uma abordagem baseada na estatística de desempenho e consistência de uma determinada arquitetura treinada. Assim, foi aplicado um processo de seleção para avaliar a performance das redes, com arquiteturas de 1 a 10 neurônios de uma camada oculta. Com isso, cada arquitetura foi treinada quarenta vezes com o mesmo conjunto de dados, onde após cada treinamento da rede, aplicou-se sinais de torque diferentes dos utilizados no treinamento. As saídas de cada rede (velocidade linear e angular) foram comparadas com a saída do modelo cinemático (*ground truth*) e o erro quadrático médio foi calculado. Assim, para cada arquitetura de rede treinada, obteve-se uma distribuição do valor RMS (*Root Mean Square Error*) do erro da saída.

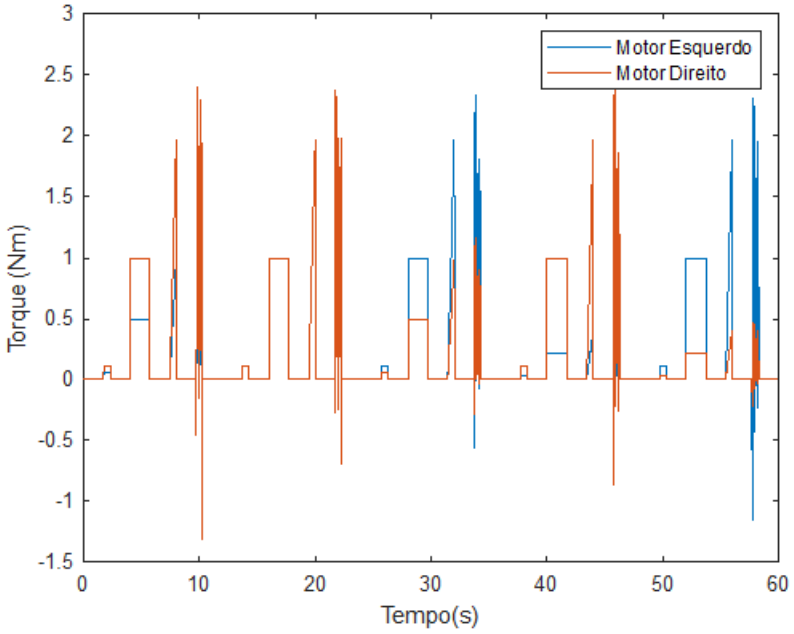

Figura 6.3: Torque nos motores para treinamento

Os sinais de torque utilizados no procedimento numérico de validação, são apresentados na Figura 6.4, sendo que nas Figuras 6.5 e 6.6 são mostrados os resultados típicos de validação, para determinadas redes de estimativa de velocidade linear e angular.

Observa-se, para cada uma das redes consideradas, as diferenças entre as velocidades reais

e estimadas. Assim, para uma mesma configuração de rede, são realizados 40 testes de validação, utilizando-se diferentes trechos dos sinais de entrada e saída, gerando um espaço amostral de onde se calculam os parâmetros estatísticos que podem aferir a qualidade de cada validação.

A Figura 6.7 ilustra os parâmetros estatísticos obtidos para validação da rede de estimação da velocidade linear com 6 neurônios, à partir de um típico espaço amostral de 40 medidas.

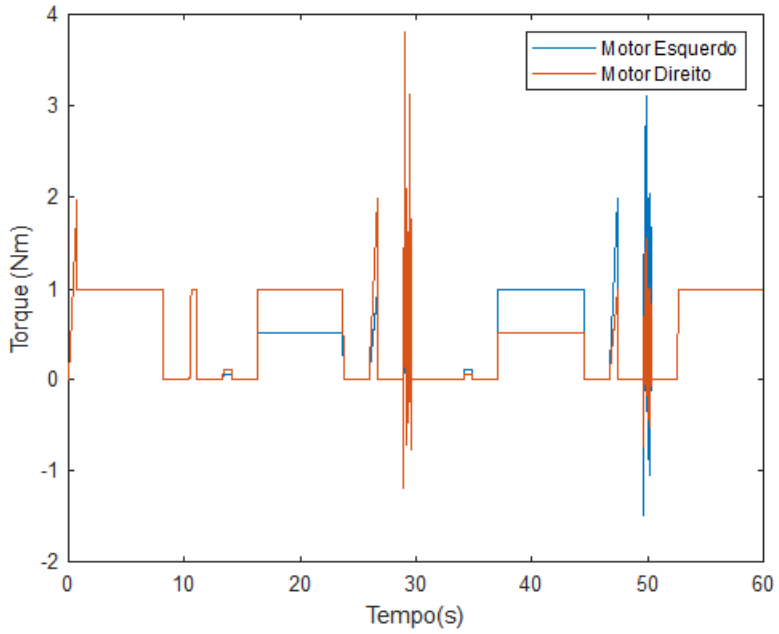

Figura 6.4: Torques utilizado para validação

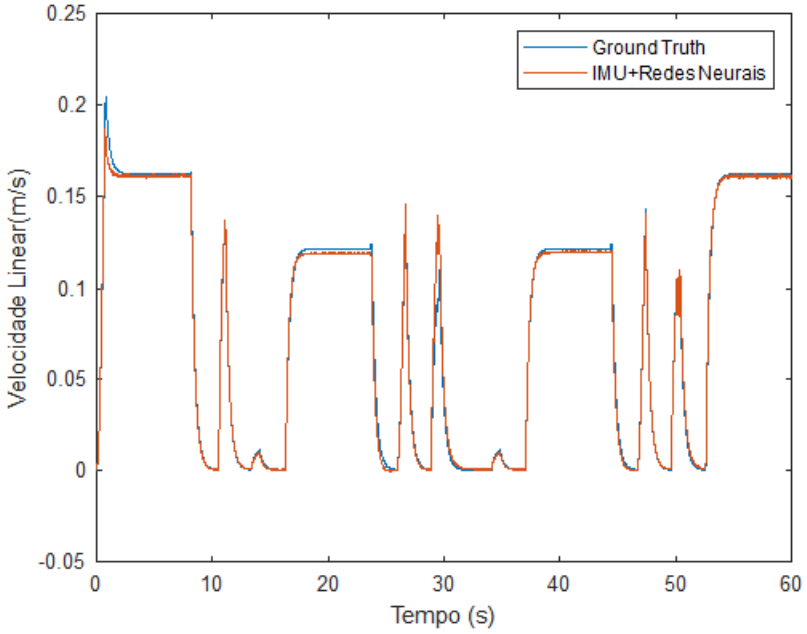

Figura 6.5: Velocidade linear real e estimada em uma iteração

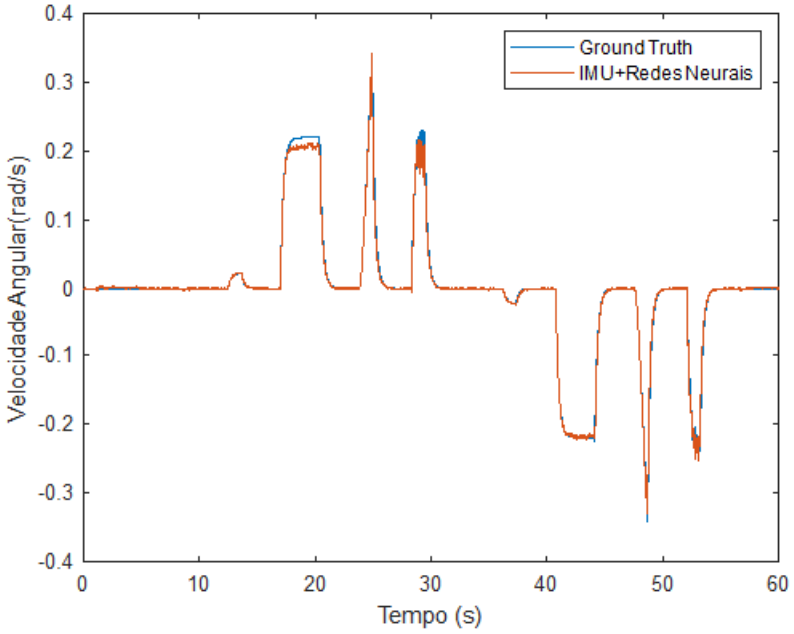

Figura 6.6: Velocidade angular real e estimada em uma iteração

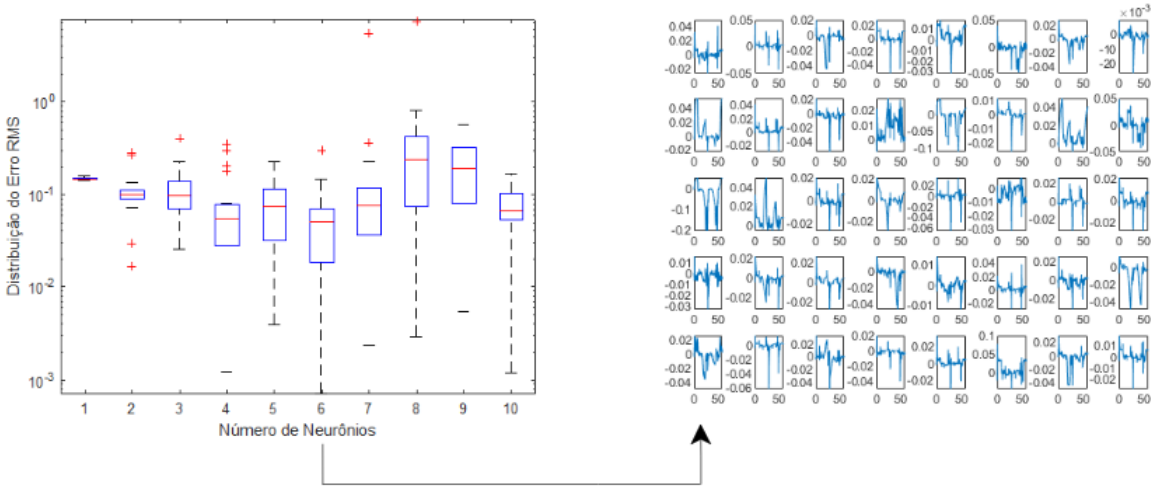

Figura 6.7: Erros de trajetória para uma rede treinada

Para definir a arquitetura ótima, a rede ideal deve possuir a menor quantidade de numero de neurônios que resulte em: menor mediana do erro quadrático médio (desempenho), menor dispersão dos pontos máximos e mínimos e menor ponto mínimo. Nesta linha, as Figuras 6.8 e 6.9 mostram os gráficos de *boxplot* com análise de performance de validação dos resultados de simulação para várias redes com diferentes números de neurônios, no qual permitem avaliar, através de inspeção visual, o desempenho e consistência de cada rede. No *boxplot*, o eixo x mostra o número de neurônios da rede e no eixo y as estatísticas relacionadas com esta rede submetida a trajetória de validação. Assim, através da inspeção da Figura 6.8, opta-se pelo uso da rede com 6 neurônios como instrumento de estimativa da velocidade linear. Contudo, a inspeção da Figura 6.9 leva a escolha da rede com 8 neurônios como instrumento de estimativa da velocidade angular do robô.

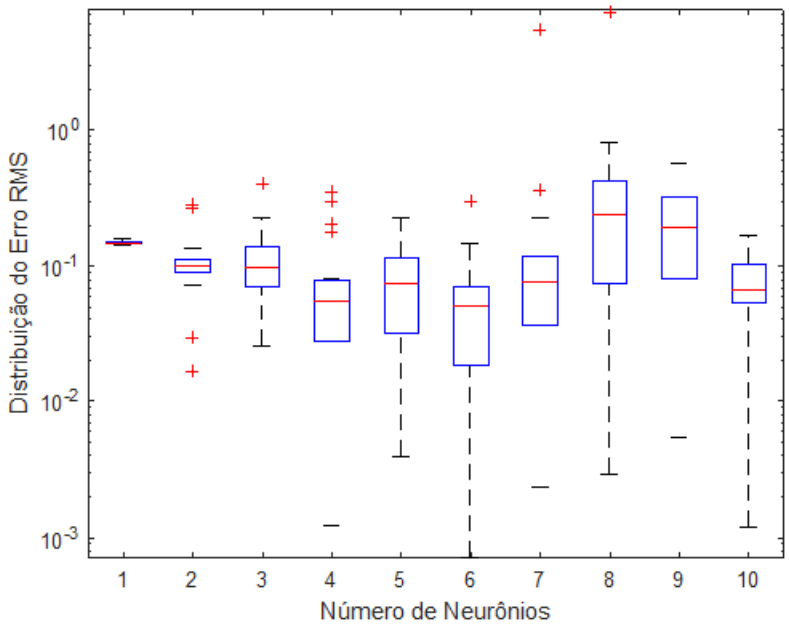

Figura 6.8: *Boxplot* para a rede de velocidade linear

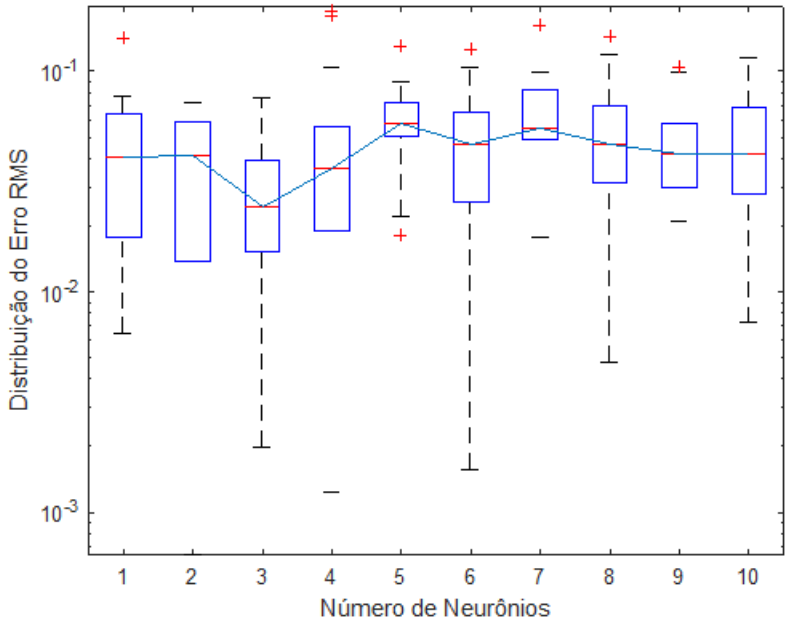

Figura 6.9: *Boxplot* para a rede de velocidade angular

Diante desta análises e definições, a seção seguinte mostra o resultado consolidado de esti-

mativa de odometria inercial e da simulação numérica de movimento robótico, utilizando as redes neurais selecionadas.

6.3 Resultados Simulados da Estimativa de Trajetória

Para a fase de teste das redes selecionadas, as entradas de torque mostradas na Figura 6.10 são aplicadas; porém, tais entradas são diferentes daquelas utilizadas nas fases de treinamento e de validação das arquiteturas de rede. O modelo cinético do robô permite o cálculo de suas saídas de velocidade, e consequentemente, a construção dos sinais sintéticos de medidas inerciais a serem utilizadas nas estimativas de movimento.

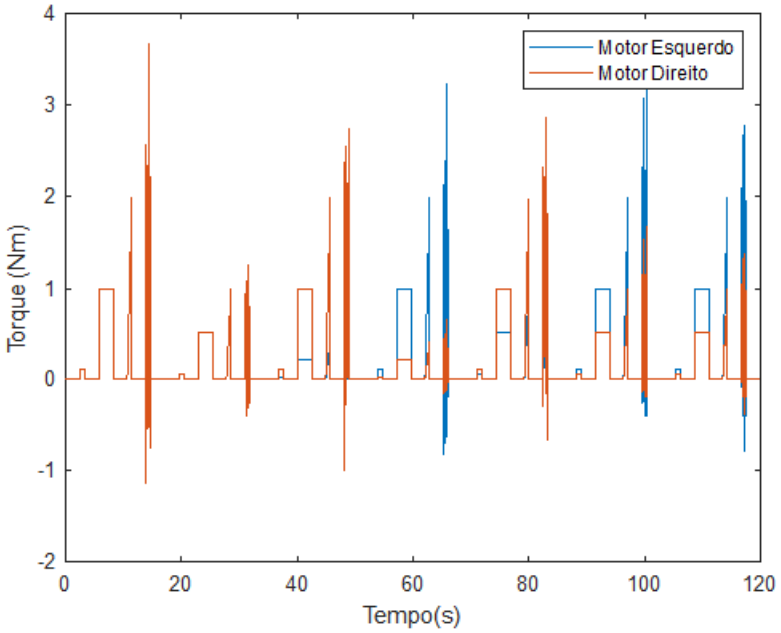

Figura 6.10: Torques de teste

Aplicando os torques de acionamento e sinais sintéticos de medidas inercias do robô, como entrada nas redes treinadas, estima-se a sua velocidade linear, que ao ser integrada fornece tanto o seu deslocamento linear (6.11) quanto a sua velocidade angular, sendo que esta ao ser integrada fornece a sua orientação (6.12). Nota-se que as estimativas de deslocamento e orientação estão sujeitos a erros persistentes cumulativos ao longo do tempo devido aos erros de estimativas dados pelas redes, conforme apresentado nas equações 5.16 e 5.17. Esses erros causam a distorção do cál-

culo da trajetória realizada pelo robô mostrado na Figura 6.13, porém, a odometria fica computada mantém consistência e a perturbação causada pelos desvios dos ruídos de medida são minimizados.

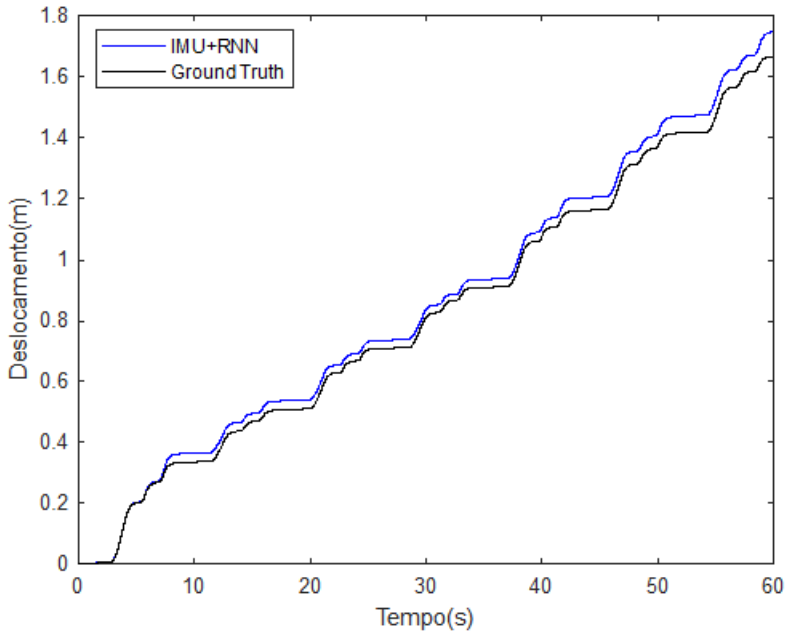

Figura 6.11: Deslocamentos estimados e reais

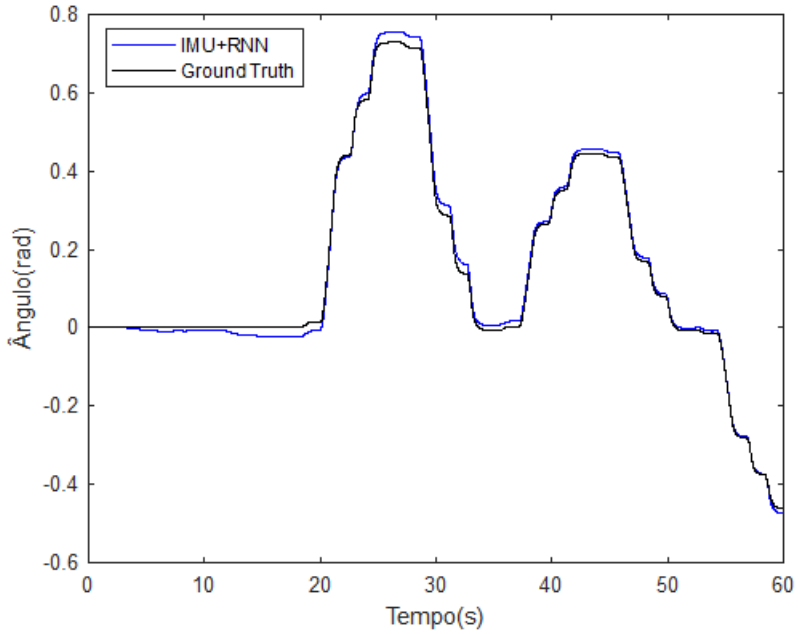

Figura 6.12: Orientações estimadas e reais

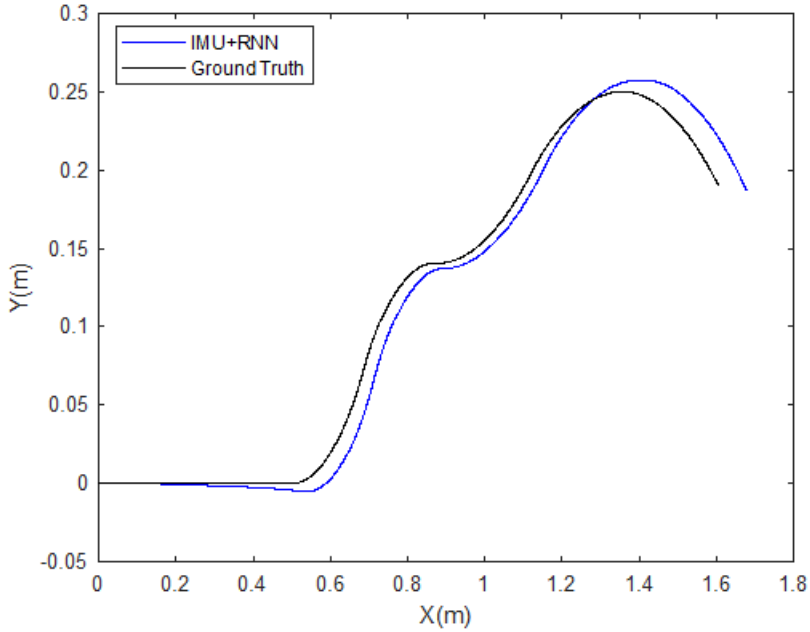

Figura 6.13: Trajetória estimada e real

Para efeito de comparação, a trajetória estimada através da técnica proposta de uso de redes neurais é colocada ao lado de outra estimativa de trajetória obtida via simples e sucessivas integrações dos sinais inerciais sintéticos, conforme mostrado na Figura 6.14. Com isso, observa-se que as incertezas presentes nos sinais inerciais sintéticos são suficientes para inviabilizar, mesmo num horizonte de curta movimentação do robô, uma correta estimativa de trajetória através de simples integrações numéricas.

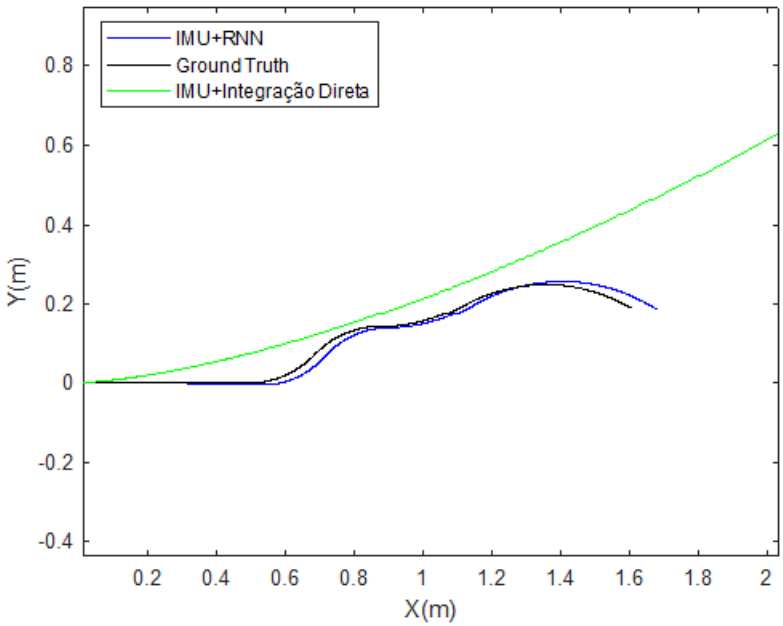

Figura 6.14: Trajetória estimada com integração direta

É oportuno salientar que, apesar do trabalho utilizar *RNRs*, foi cogitada a possibilidade de utilizar redes do tipo *feedforward*; ou seja, foram realizados testes preliminares de comparação entre ambas topologias.

Como resultado da análise, percebeu-se que a rede *feedforward*, no contexto do trabalho de predição de série temporal, é capaz de reduzir o efeito da contaminação de *bias* na medição, mas permanece sensível à contaminação de ruídos gaussianos do sinal. Contudo, a *RNR*, além de minimizar a influência de contaminação de *bias*, apresentou também um comportamento de filtro passa-baixas, respeitando o espectro de frequência esperado pelo *target* de treinamento, fazendo com que os sinais estimados melhor se aproximem do sinal real. Diante deste desempenho, ficou evidente que a *RNR* proporciona melhores resultados para a técnica proposta.

7 Aplicação Experimental

O método de odometria inercial proposto, foi aplicado experimentalmente a um robô diferencial de pequeno porte, contendo dois *encoders*, dois sensores de corrente e uma *IMU* (com acelerômetro e giroscópio). O robô foi construído especialmente para o propósito da validação do método. Devido ao desenvolvimento da parte experimental deste trabalho ter se passado dentro do período de pandemia, utilizou-se instrumentos mais modestos e de baixo custo.

Foram realizados testes de navegação do robô em um ambiente *indoor* e os dados dos sensores foram coletados para treinamento, validação e testes da rede neural que realiza as estimativas de trajetórias *offline*.

7.1 Plataforma Robótica e Configuração Experimental

A plataforma robótica utilizada para testes da odometria inercial por redes neurais é mostrada na Figura 7.1. Os principais componentes embarcados no robô são listados abaixo.

1. Um microcontrolador ESP32, *clock* de 160 MHz, *Dual Core*, 520Kb SRAM, Wi-Fi: 802.11 b/g/n.
2. Dois sensores de corrente INA219.
3. Dois *encoders* de rodas com resolução de 341PPR.
4. Dois motores CC com tensão nominal de 6V e rotação máxima de 210RPM.
5. Uma *IMU* MPU6050 (contém um acelerômetro e um giroscópio, ambos de 3 eixos).
6. *Driver* de potência L298N (Ponte H para dois motores).
7. Um conversor CC-CC do tipo *Buck* para abaixar a tensão da bateria de lítio-ion (em torno de $7.4V$) para $5V$.

O chassi do robô foi projetado no *software* de CAD Fusion 360 e confeccionado em impressora 3D com material PLA. O *firmware* embarcado no microcontrolador ESP32 foi desenvolvido

utilizando a IDE Arduino, e o diagrama de blocos da Figura 7.2 mostra a arquitetura do sistema.

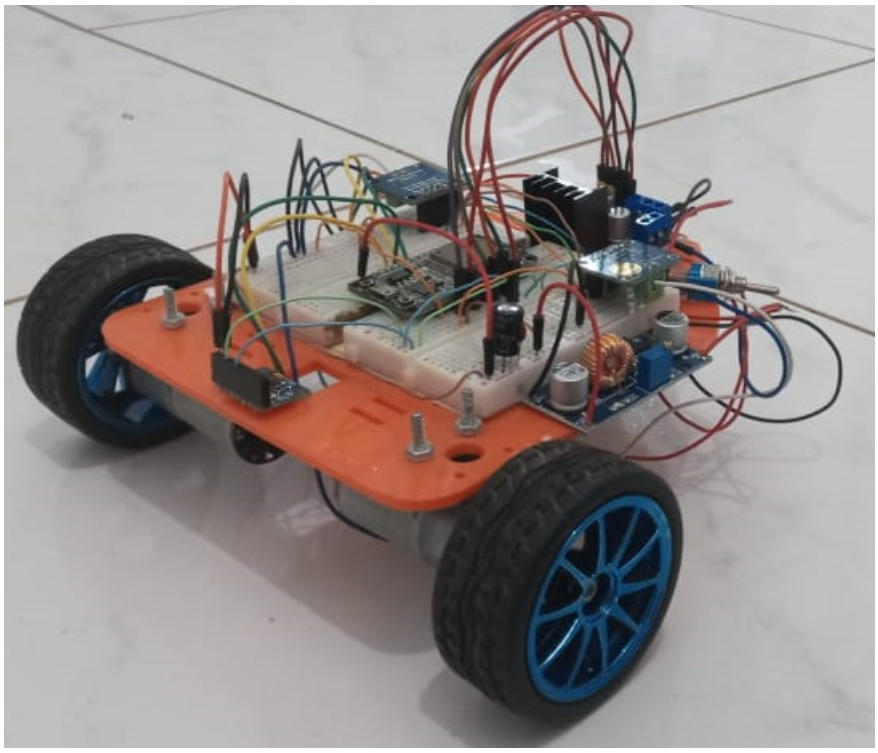

Figura 7.1: Plataforma de Teste.

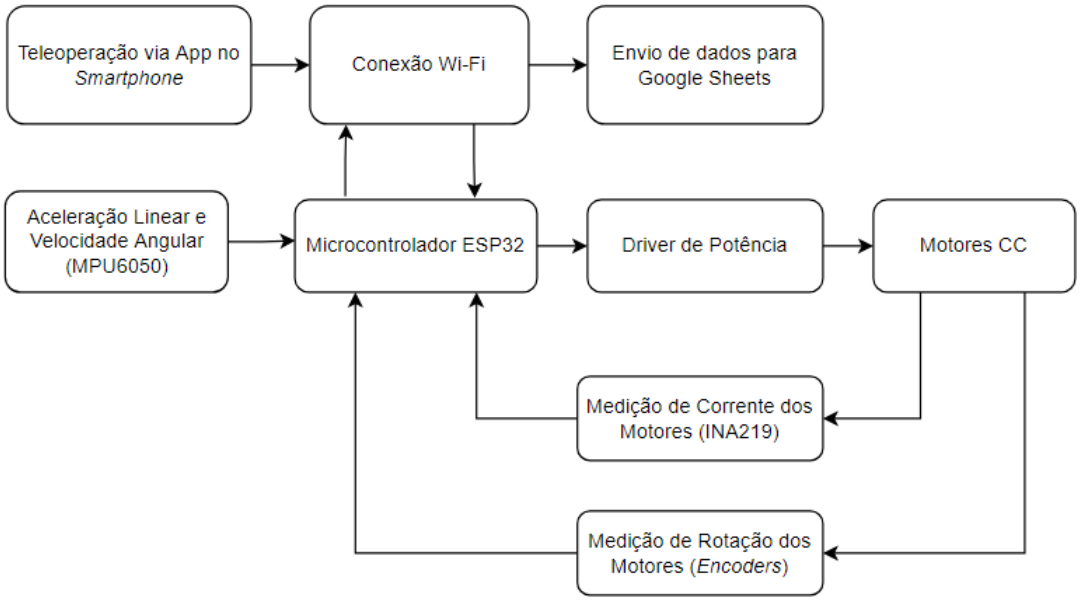

Figura 7.2: Diagrama de Blocos de Funcionamento do Sistema Robótico

Conforme apresentado na Figura 7.2, o robô é teleoperado através de um aplicativo em um *smartphone* desenvolvido utilizando o app Blynk que envia instruções de controle do robô pela internet para um servidor *broker*. O microcontrolador ESP32 conectado a rede local, por sua vez, faz a consulta dessas instruções no *broker* e envia os sinais correspondentes para o *driver* de potência comandar os motores. Além disso, o microcontrolador também é responsável por realizar a leitura dos sensores de corrente, o cômputo da rotação dos motores dado pelos *encoders* e leitura dos dados de aceleração nos eixos x e y e velocidade angular em torno do eixo z, medidos pela *IMU*.

A frequência de amostragem de leitura de todos os dados é aproximadamente $8Hz$. Após os dados preencherem as pilhas de memória, os mesmos são enviados no formato JSON para uma planilha no Google *Sheets* via HTTP Post. Em sequência, o MATLAB importa os dados da planilha para a realização do treinamento e validação das redes neurais. Os dados de velocidade dados pelo *encoder* são utilizados para gerar dados de velocidade linear e angular via modelo cinemático do robô, sendo essa uma abordagem consolidada na literatura para fazer estimativas de movimento o tipo *dead reckoning* de um veículo. Assim, é possível fazer comparação de desempenho da odometria inercial assistida por redes neurais com odometria de rodas via modelo cinemático.

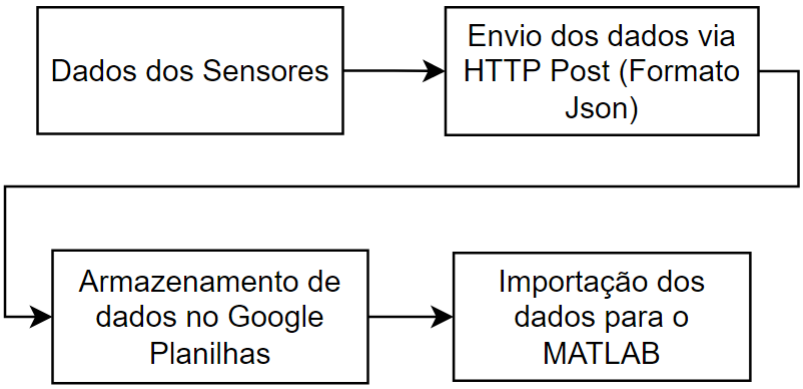

Figura 7.3: Diagrama de fluxo de aquisição do *dataset*

Dois marcadores visuais são fixados no robô, assim como uma câmera superior, posicionada de forma a cobrir toda a área de trajeto possível a ser realizado pelo mesmo. Esse conjunto óptico é usado para fornecer a trajetória referência (*Ground Truth* de movimentação do robô). Um exemplo de imagem capturada pela câmera, e trajetória dos marcadores do robô, são mostrados na Figura 7.4. A câmera utilizada foi de um *smartphone* com uma taxa de 30 quadros por segundo.

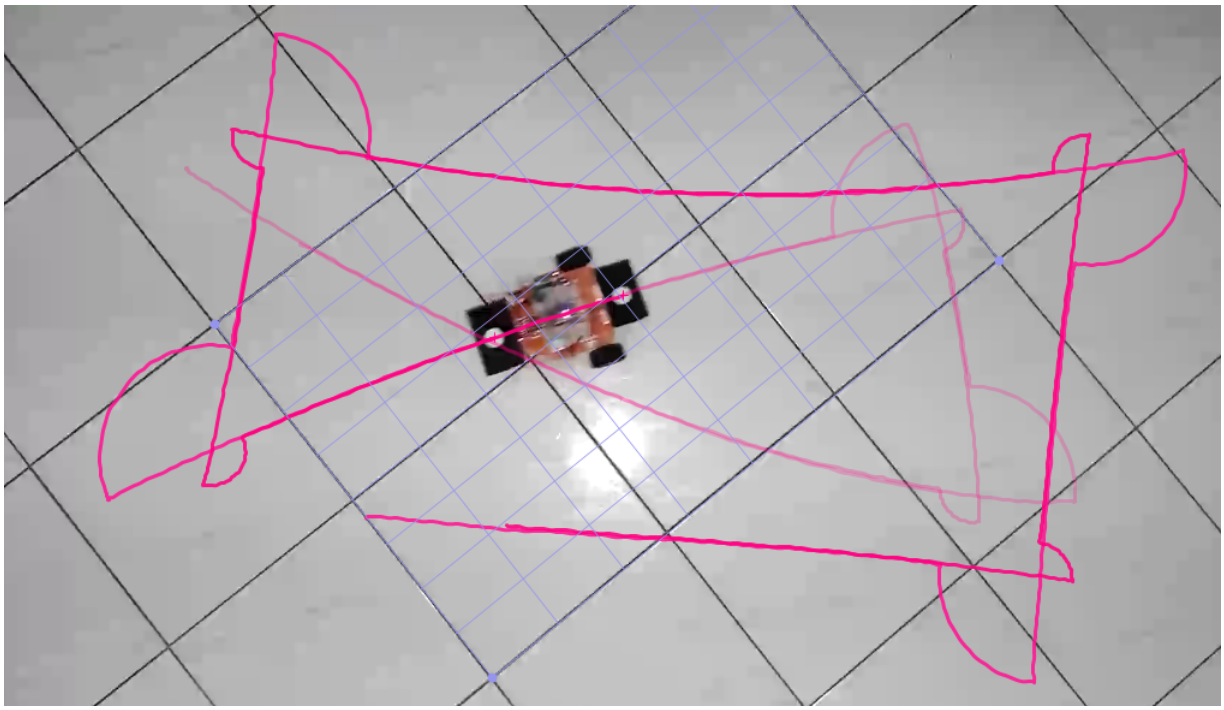

Figura 7.4: Kinovea rastreando o movimento do robô capturado por câmera

A câmera grava todo o movimento do robô enquanto este está sendo operado remotamente, para posteriormente ser realizado o tratamento das imagens na plataforma Kinovea - *software* de código aberto utilizado em aplicações de biomecânica. Dessa forma, é possível rastrear os marcadores posicionados no robô enquanto o mesmo se desloca pelo ambiente a fim de registrar a velocidade linear e angular do mesmo para serem utilizadas na fase de treinamento, validação e teste da metodologia proposta.

A câmera utilizada filma com uma taxa de 30 quadros por segundo. O Kinovea rastreia os dois marcadores posicionados no robô, armazenando suas posições X e Y ao longo da trajetória no referencial fixo da imagem. A partir das posições obtidas dos dois marcadores, foi computado o ângulo de orientação e o deslocamento longitudinal do robô no seu referencial móvel, e assim, estimado a velocidade linear e angular do mesmo para serem utilizados como alvo para o treinamento das redes neurais.

7.2 Treinamento e Validação das Redes Neurais para Odometria Inercial

A aplicação do método proposto de odometria inercial, assistida por redes neurais, é implementado na plataforma MATLAB. Assim como no ambiente simulado, são construídas duas redes, uma para estimar velocidade linear longitudinal e outra para estimar velocidade angular, conforme o esquema mostrado na Figura 7.5. As redes neurais são treinadas com um *dataset* com o tamanho de $1.4MB$, ou cerca de $200s$ de registros da movimentação do robô.

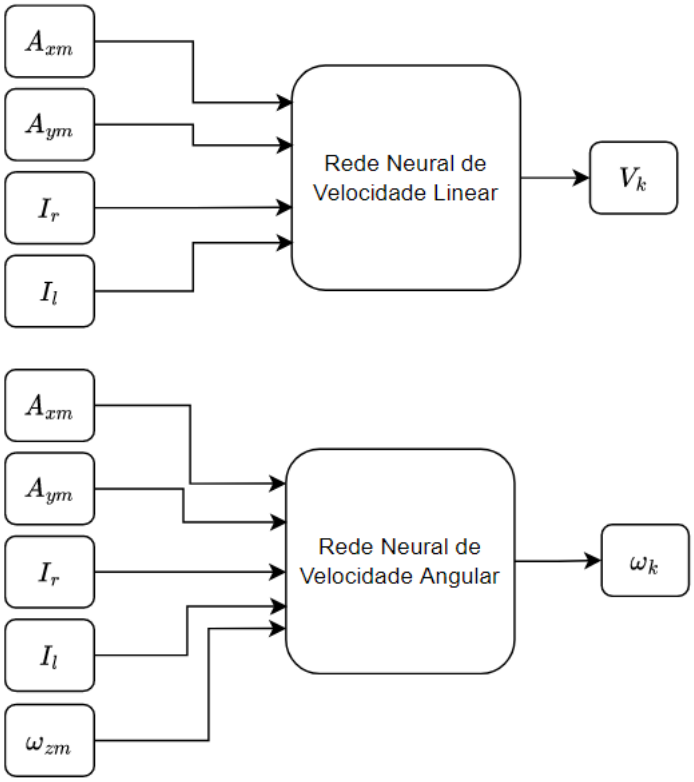

Figura 7.5: Arquitetura da Rede Neural para Velocidade Linear e Angular

Têm-se, da Figura 7.5, que I_r é a corrente no motor direito, I_l é a corrente no motor esquerdo, A_x a aceleração linear longitudinal e A_y a aceleração linear lateral. Visto que os atuadores utilizados são motores de corrente contínua, a relação entre torque T e corrente elétrica i_m é dada pela Equação 5.11. Portanto, as medidas das correntes dos motores foram utilizadas no método como as entradas de torque das redes, visto que a relação entre essas grandezas é linear.

As velocidades V_k e ω_k, utilizadas como o alvo dos treinamentos, são a velocidade linear e angular no referencial móvel estimadas a partir dos dados x e y coletados pelo Kinovea no referencial inercial dos marcadores rastreados pelo *software*.

Os dados de corrente dos motores, aceleração linear e velocidade angular coletados no experimento de teleoperação para treinamento das redes são mostrados nas Figuras 7.6 à 7.8.

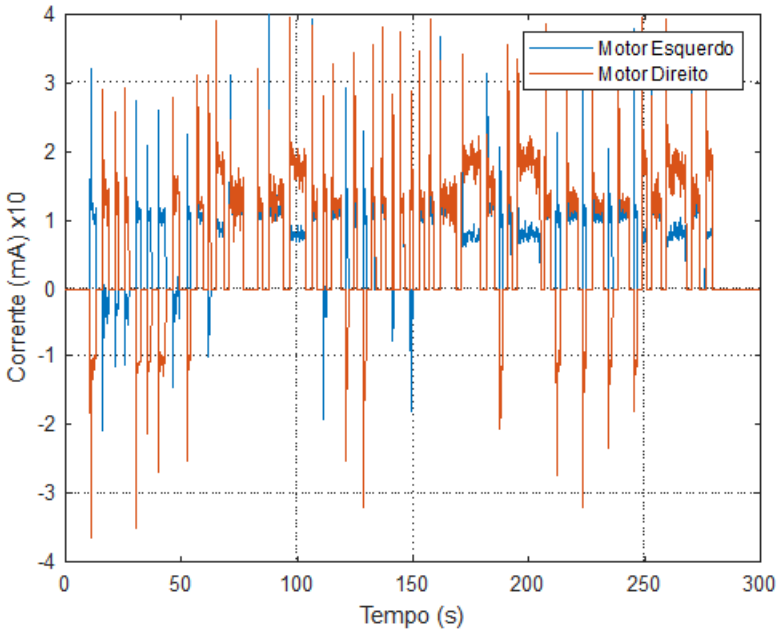

Figura 7.6: Correntes do motor esquerdo e direito para treinamento das redes

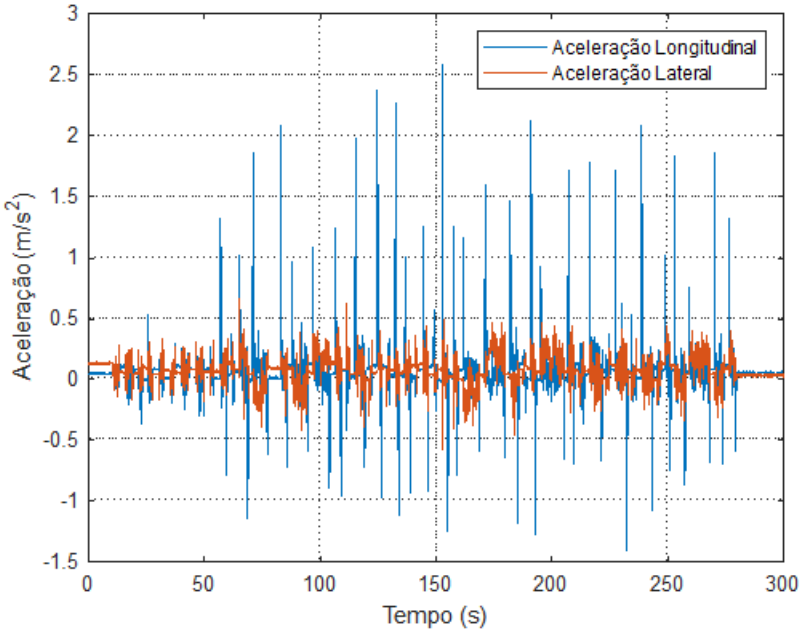

Figura 7.7: Aceleração longitudinal e lateral para treinamento das redes

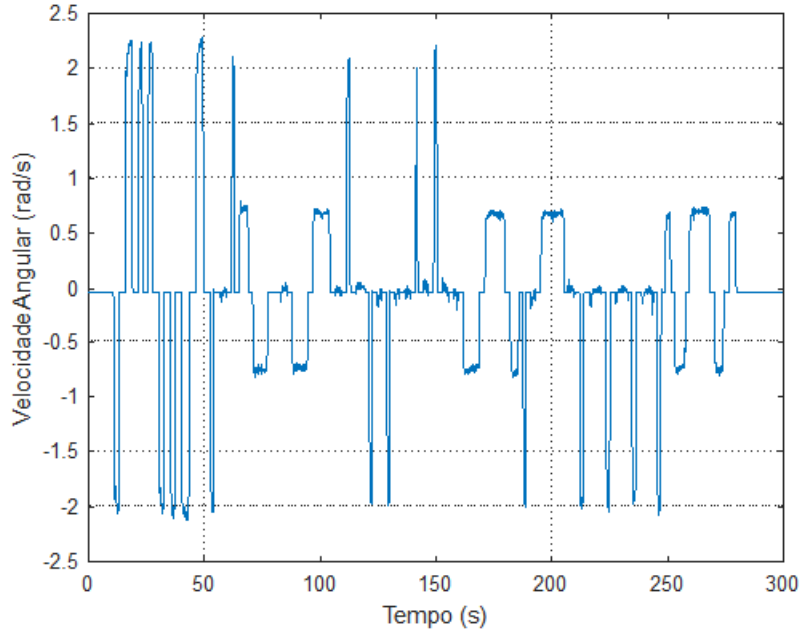

Figura 7.8: Velocidade angular para treinamento das redes

A velocidade linear longitudinal e velocidade angular do *Ground Truth* são mostradas nas

Figuras 7.9 e 7.10, sendo possível observar que a velocidade linear se mostra ruidosa devido as variações de rastreamento dos marcadores do robô pelo *software* Kinovea.

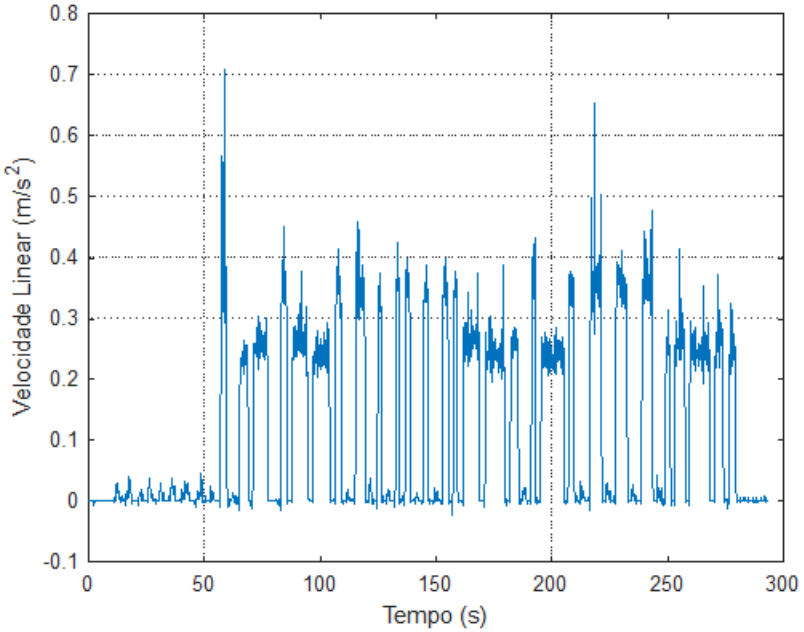

Figura 7.9: Velocidade Linear do *Ground Truth*

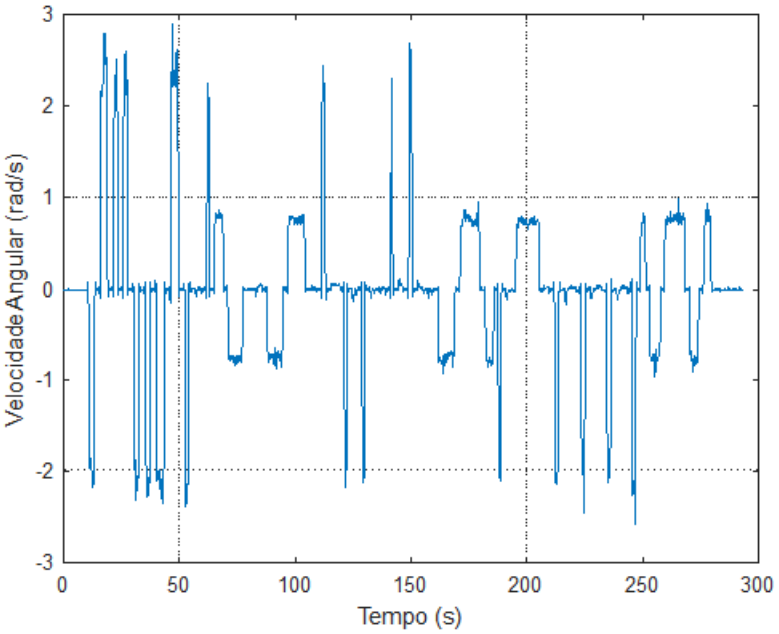

Figura 7.10: Velocidade angular do *Ground Truth*

Após o treinamento de cada rede, a mesma é validada aplicando sinais de entrada coletados em outro experimento; ou seja, a partir dos dados de corrente dos motores, aceleração linear e velocidade angular mostrados nas Figuras 7.11 à 7.13.

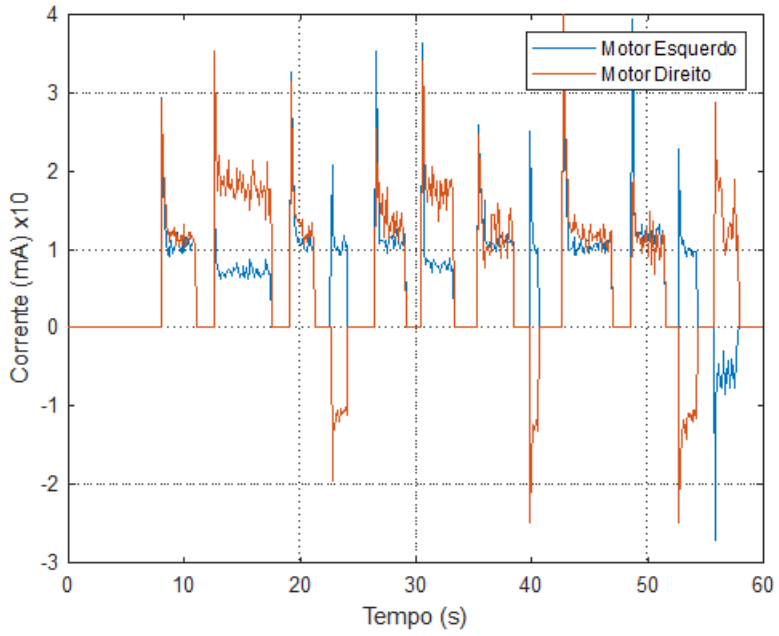

Figura 7.11: Correntes nos motores para validação

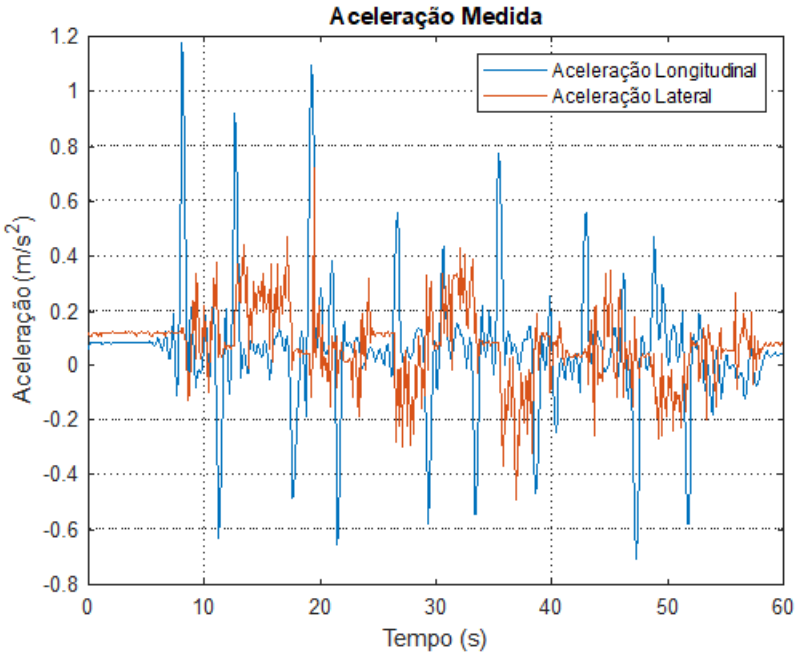

Figura 7.12: Aceleração linear e lateral para validação

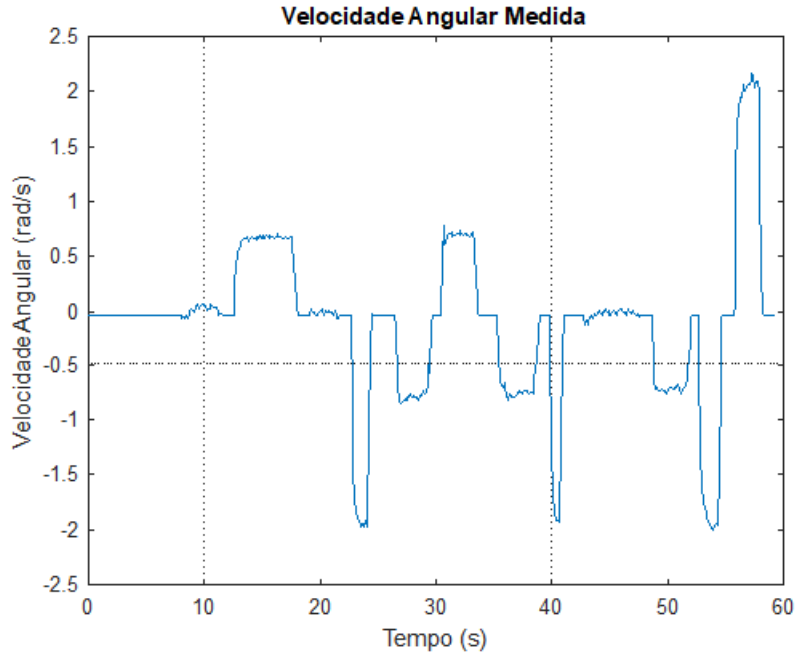

Figura 7.13: Velocidade angular para validação

Na Figura 7.14, apresenta-se o comparativo entre a velocidade linear estimada pela rede neu-

ral, a velocidade linear dado pelo modelo cinemático via *encoder* e o *Ground Truth* dado pelo Kinovea.

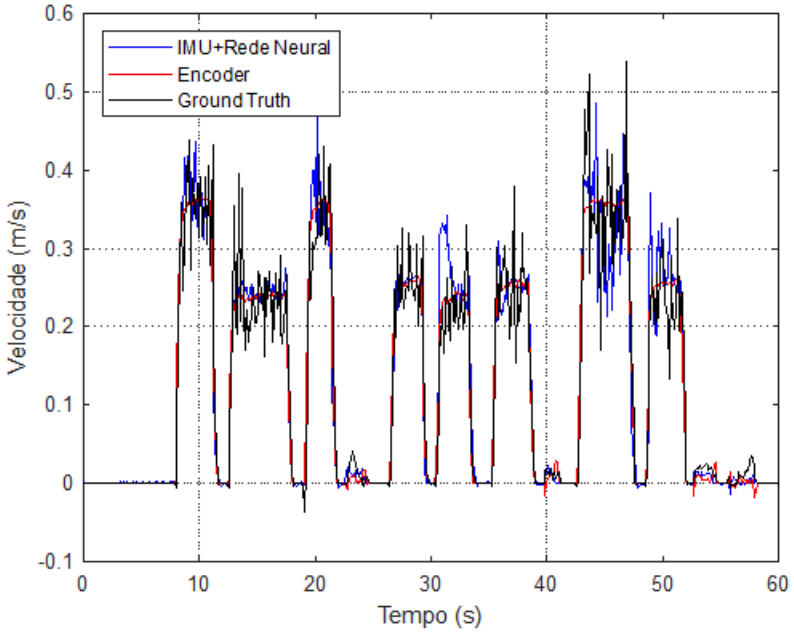

Figura 7.14: Comparativo entre velocidades lineares

A Figura 7.15 apresenta o comparativo entre a velocidade angular estimada pela rede neural, a velocidade angular dado pelo modelo cinemático via *encoder* e o *Ground Truth* dado pelo Kinovea.

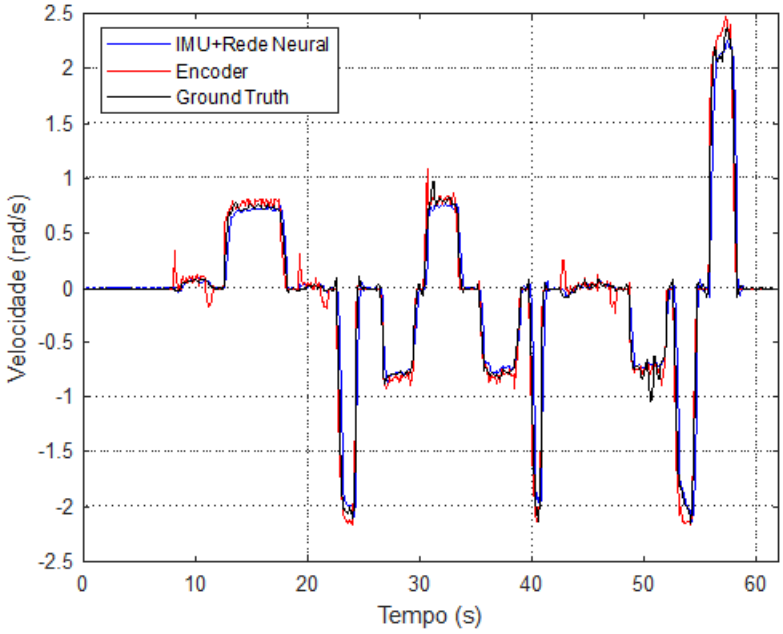

Figura 7.15: Comparativo entre velocidades angulares

Ao integrar as velocidades lineares, para obter os deslocamentos estimados (Figura 7.16), e as velocidades angulares, para obter as orientações estimadas (Figura 7.17), calcula-se os erros quadráticos médios das trajetórias de cada variável estimada, conforme se encontra na Tabela 7.2. Na Tabela 7.1 é mostrado o tempo de execução para treinar 40 vezes as redes neurais de 1 a 10 neurônios e o tempo de treinamento apenas para a redes de 10 neurônios.

Tabela 7.1: Erros de deslocamento e angulo

Rede Estimadora	10 neurônios	1 a 10 neurônios
Velocidade Linear (m)	9.8096 min	36 min
Velocidade Angular (rad)	6.6642 min	32 min

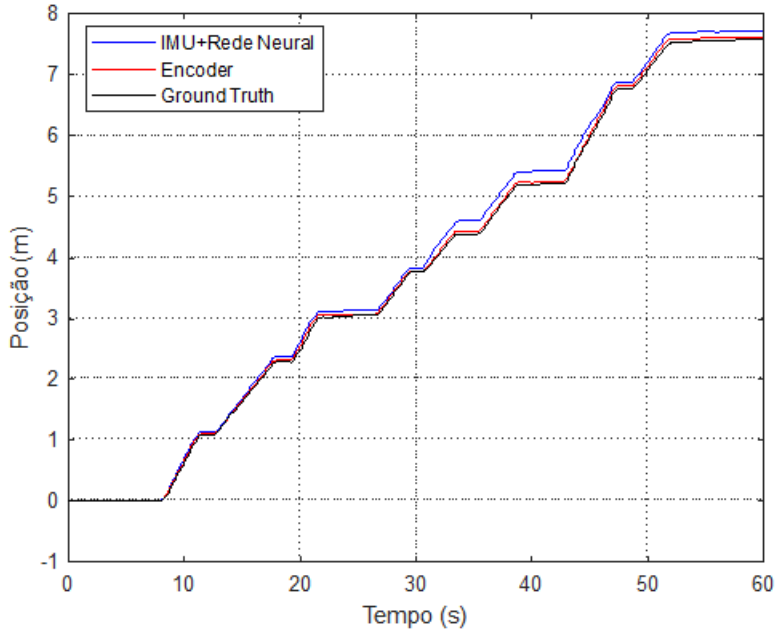

Figura 7.16: Comparativo entre deslocamentos estimados

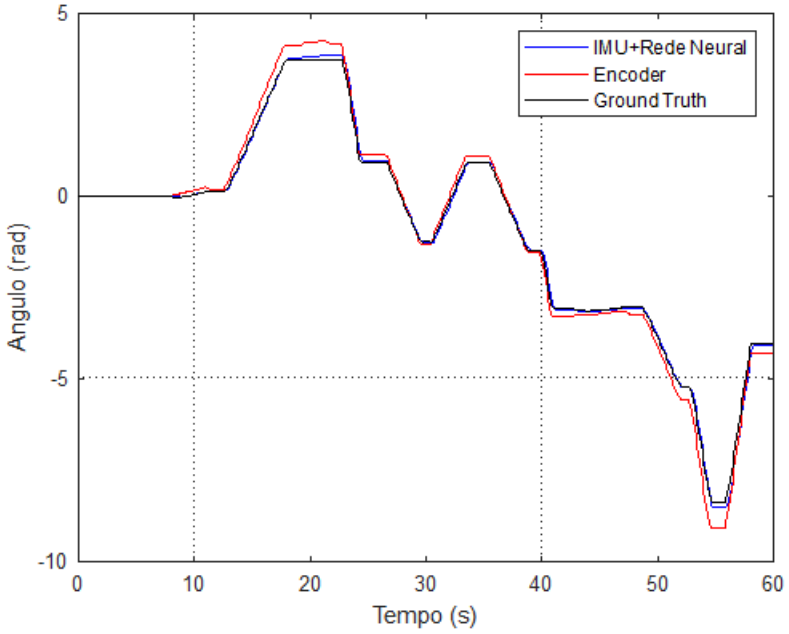

Figura 7.17: Comparativo entre orientações estimadas

Tabela 7.2: Erros de deslocamento e angulo

Variável Estimada	Erro RMS (IMU+Redes Neurais)	Erro RMS (Encoder+Modelo Cinemático)
Deslocamento (m)	0.1339	0.0394
Ângulo (rad)	0.1062	0.2776

Pode-se portanto aplicar a Equação 3.21 para computar a trajetória (Figura 7.18) do robô estimada pelo método de odometria inercial, odometria de rodas e Kinovea (*Ground Truth*).

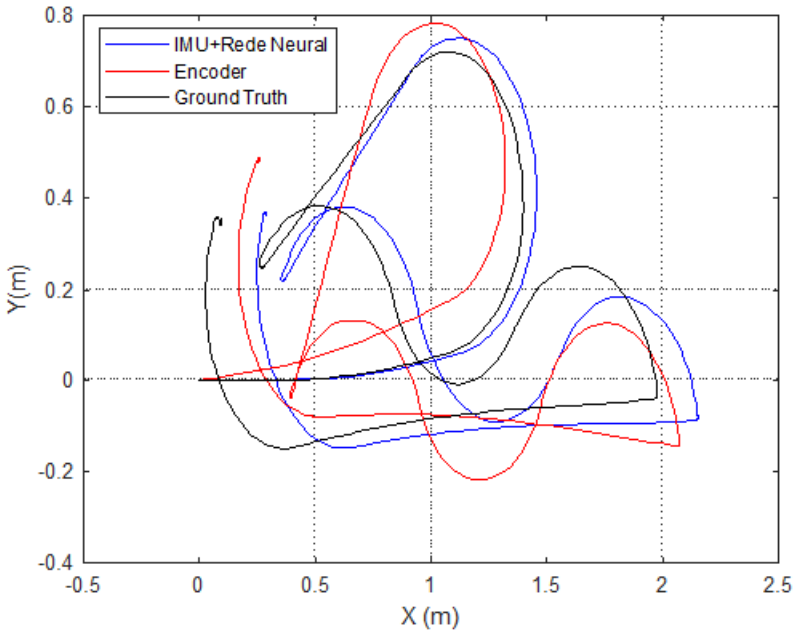

Figura 7.18: Comparativo entre trajetórias estimadas

Apesar do erro de deslocamento estimado pelo modelo cinemático com *encoder* ser menor, o erro de trajetória estimada pelo método de odometria inercial é menor, visto que o estimador de orientação do mesmo teve um desempenho mais próximo do valor real. Os dados de erro de posição ao longo da trajetória nos eixos x e y são mostrados na Tabela 7.3. Além disso, a rede neural corrige o potencial erro de divergência linear ao integrar o sinal do giroscópio para obter orientação e a divergência quadrática ao integrar duas vezes o sinal do acelerômetro para obter deslocamento, como é apresentado nas Figuras 7.19 e 7.20. Apesar dos valores de *bias* $0.08m/s^2$, para aceleração longitudinal medida, e $0.03rad/s$, para velocidade angular medida, o processo

de integração direta dos dados brutos da *IMU* gera um rápido acumulo de erro nas estimativas de deslocamento e orientação, mas que ao serem aplicados na rede neural junto às entradas de corrente elétrica dos motores, faz com que a estimativa de estados sejam coerentes em comparação a estimativa via *encoder* e *Ground Truth*.

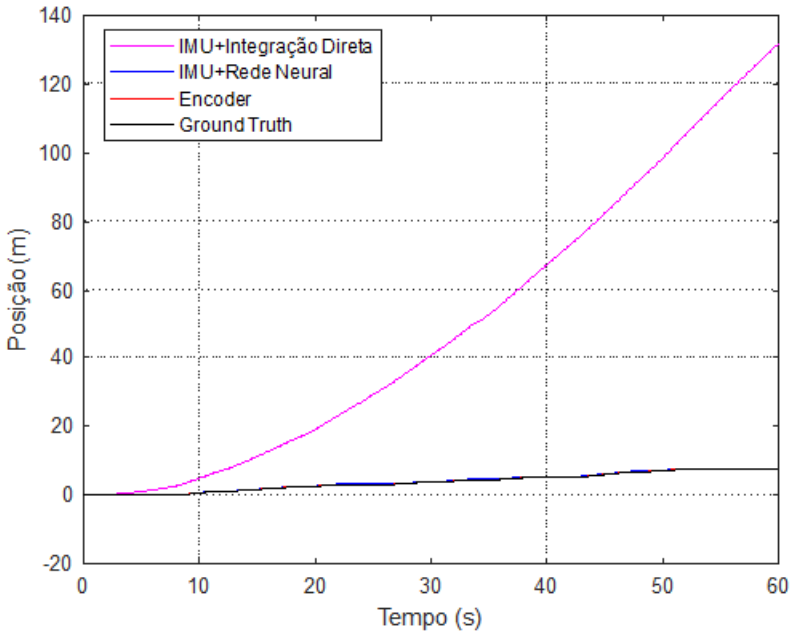

Figura 7.19: Comparativo entre deslocamentos estimados com integração direta

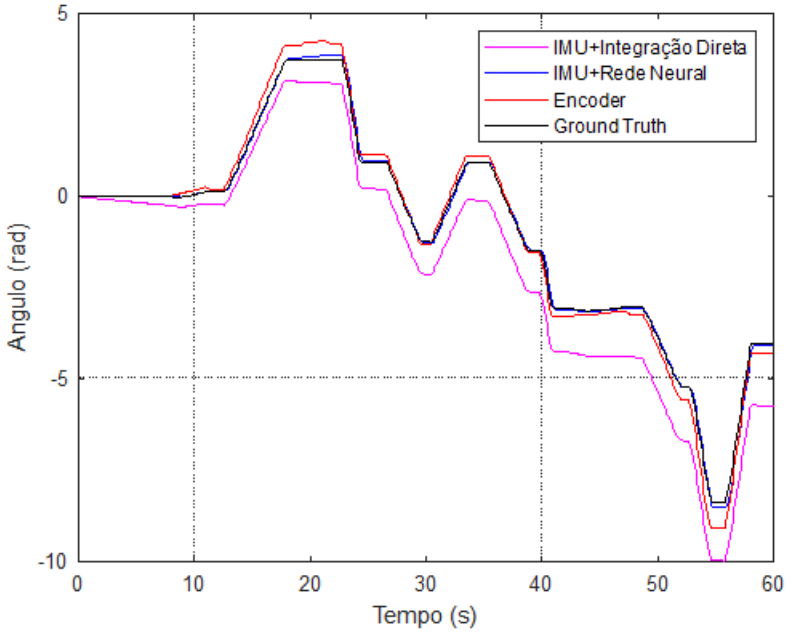

Figura 7.20: Comparativo entre orientações estimadas com integração direta

Tabela 7.3: Erros de posição X,Y

Variável Estimada	Erro RMS (IMU+Redes Neurais)	Erro RMS (Encoder+Modelo Cinemático)
Posição em X (m)	0.1442	0.1112
Posição em Y (m)	0.0358	0.1515

Dessa forma, a abordagem de odometria inercial via redes neurais utilizando dados de corrente dos motores e das medições da IMU pôde ser validada, considerando que a mesma teve uma performance equivalente à abordagem via modelo cinemático e *encoder* e ao mesmo tempo viabilizou a utilização dos dados brutos da IMU para estimativa dos estados de deslocamento e orientação. Ambas as redes estimadoras para velocidade linear e angular são compostas por 1 camada oculta com 10 neurônios. O método utilizado na Seção 6.2 foi utilizado apenas como uma referência de partida, mas a posteriori, o número de neurônios foram escolhidos por ajustes finos de tentativa e erro. O método de treinamento utilizado foi Levenberg-Marquardt. O critério de parada para os treinamentos foram considerando o erro quadrático médio menor que 10^{-3} para a rede de velocidade linear e angular. Na próxima seção será mostrada as rede neurais previamente treinadas, sendo submetidas a outras trajetórias para realização de testes.

7.3 Testes das Redes Neurais para Odometria Inercial

A seguir será mostrado e discutido os resultados referentes à duas trajetórias realizadas para testar as redes neurais treinadas na seção anterior, sendo comparada com os resultados obtidos por modelo cinemático com *encoder*. As trajetórias realizadas para teste foram executadas para avaliar o robô em condições mistas de movimento seu movimento, como deslocamento linear, curvas longas e giros em torno do próprio eixo. Além dos resultados obtidos pelas redes neurais e pelo *encoder*, também são mostrados as estimativas referentes a integração direta dos dados da *IMU*, enfatizando a potencialidade do método proposto em mitigar os erros propagados pela ultima abordagem.

7.3.1 Trajetória de Teste 1

A trajetória deste teste é composto por percursos com trechos retos e curvas longas. O deslocamento é mostrado na Figura 7.21, a orientação na Figura 7.22 e a trajetória estimada na Figura 7.23. A Tabela 7.4 mostra os erros *RMS* de cada variável estimada.A tabela também mostra a forte correlação das variáveis estimadas pelas redes neurais com o *ground truth*. Neste teste a odometria inercial teve melhor desempenho na estimativa de deslocamento, orientação e a posição no eixo X da trajetória realizada. A trajetória neste teste estimada pelas redes neurais teve um resultado mais coerente em relação à trajetória alvo, principalmente referente à orientação da trajetória total.

Tabela 7.4: Erros RMS do teste 1

Variável	Erro RMS (IMU+RN)	Erro RMS (Encoder+Cinemática)	Correlação (IMU+RN)
Deslocamento (m)	0.0935	0.2116	0.9991
Orientação (rad)	0.1625	0.3769	0.9968
Posição em X (m)	0.0690	0.1112	0.9910
Posição em Y (m)	0.2153	0.1484	0.9693

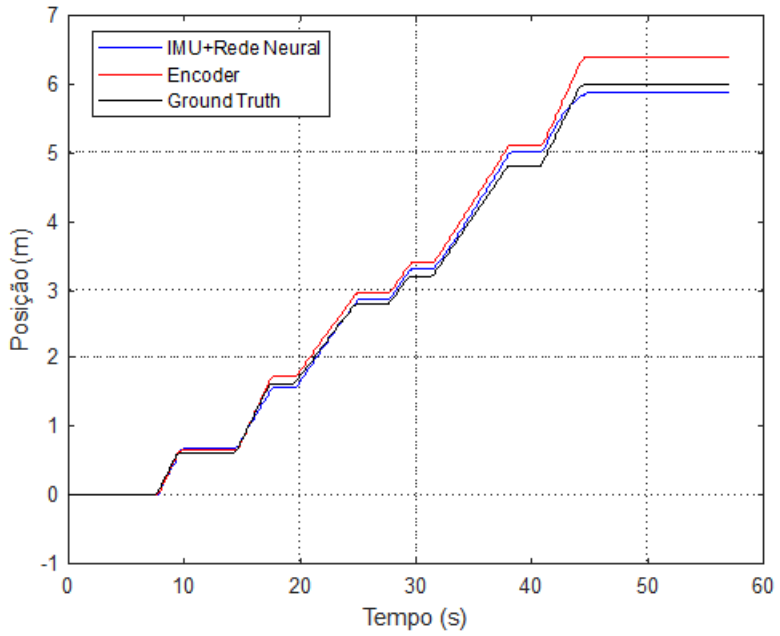

Figura 7.21: Deslocamento estimado no teste 1

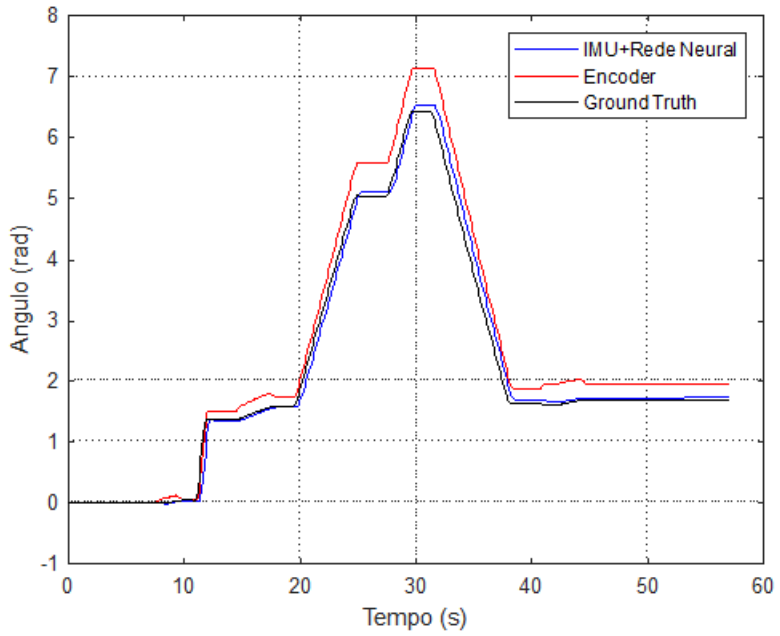

Figura 7.22: Orientação estimada no teste 1

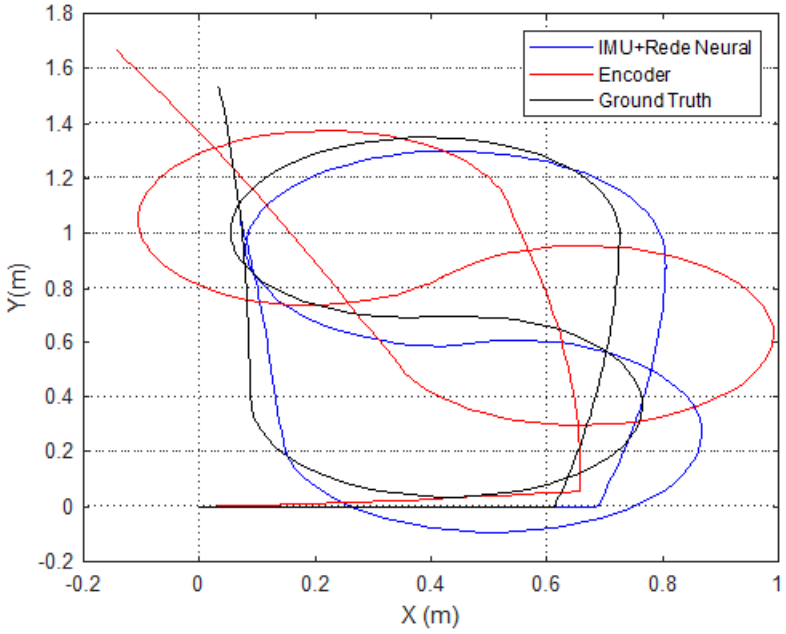

Figura 7.23: Trajetória estimada no teste 1

Nas Figuras 7.24 e 7.25, pode-se observar as estimativas de deslocamento e orientação, com os diferentes métodos considerados. É notável que as estimativas dadas pela integração dos dados brutos de aceleração e velocidade angular divergem rapidamente com o tempo, como fundamentado pelas Equações 3.10 e 3.19. O erro de integração ocorre mesmo quando o robô está completamente em repouso, como nos segundos iniciais ($0s$ a $10s$ aproximadamente) e nos segundos finais após $45s$ devido ao *bias* presente nas medições da IMU. Esses resultados mostram uma das grandes dificuldades de se implementar odometria inercial em veículos com rodas, especialmente em relação à estimativa de deslocamento que acumula erros quadraticamente com o tempo. Apesar da estimativa de orientação dado pela integração direta tenha um desempenho relativo melhor do que a estimativa de deslocamento, os erros são suficientes para causar grandes desvios na reconstrução da pose do robô ao longo do tempo, visto que os erros de orientação na estimativa de trajetória realizada traz maiores consequências do que erros na estimativa de deslocamento.

O método proposto portanto, foi capaz de manter o deslocamento e orientação estimadas confinadas em uma região próxima ao *ground truth* e a odometria de rodas dado pelo *encoder*.

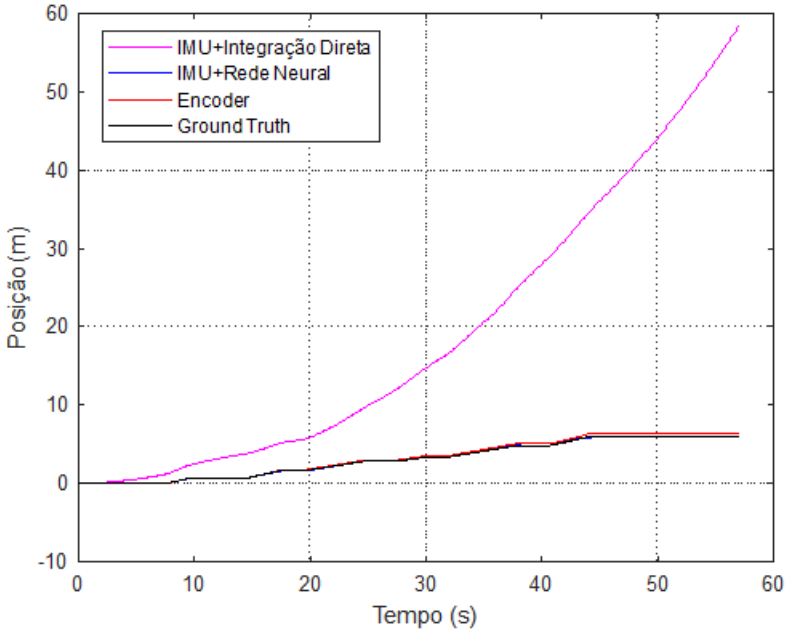

Figura 7.24: Deslocamento estimado com integração direta do sinal do acelerômetro

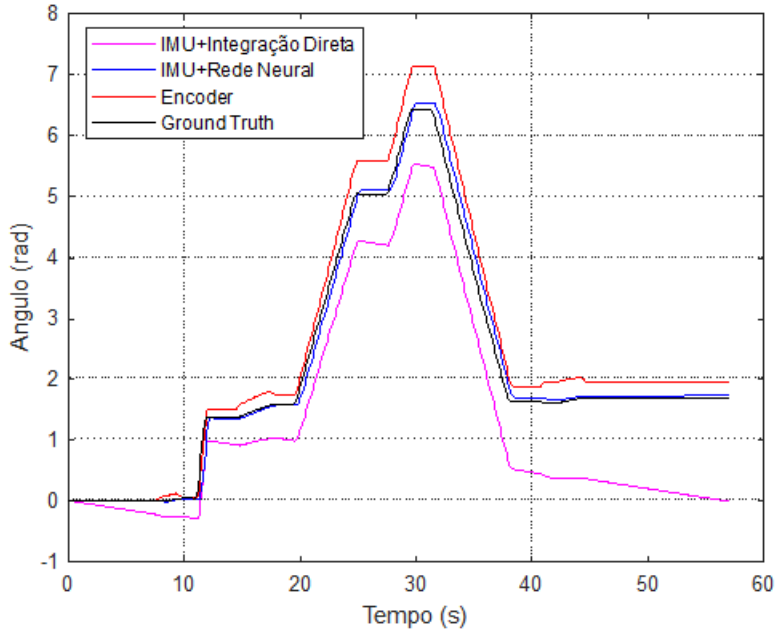

Figura 7.25: Orientação estimada com integração direta do sinal do giroscópio

Os histogramas da densidade de probabilidade dos erros das redes, expressos nas Equa-

ções 5.16 e 5.17 para velocidade linear V_l e angular ω_z, são mostrados nas Figuras 7.26 e 7.27. Para o estimador via *encoder* com modelo cinemático, os histogramas são apresentados nas Figuras 7.28 e 7.29.

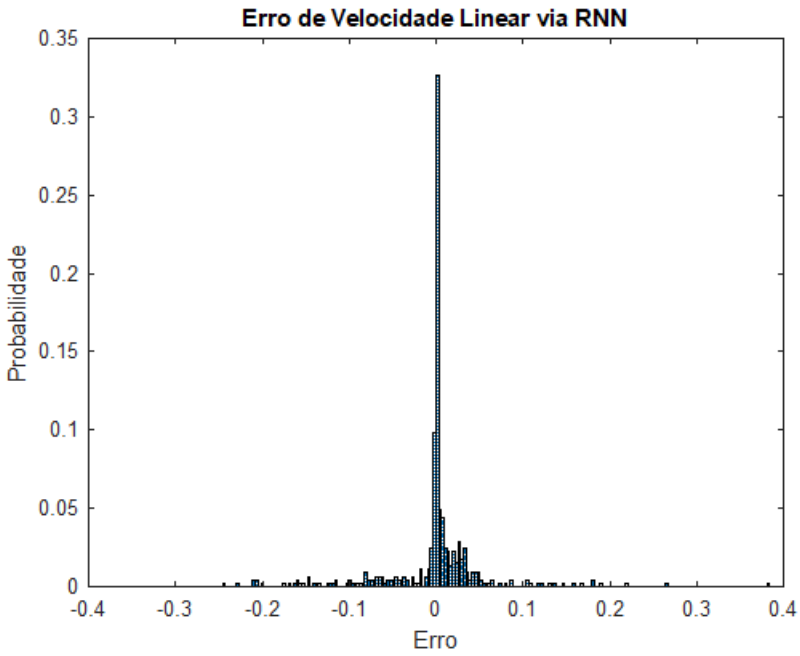

Figura 7.26: Distribuição de probabilidade do erro para V_l

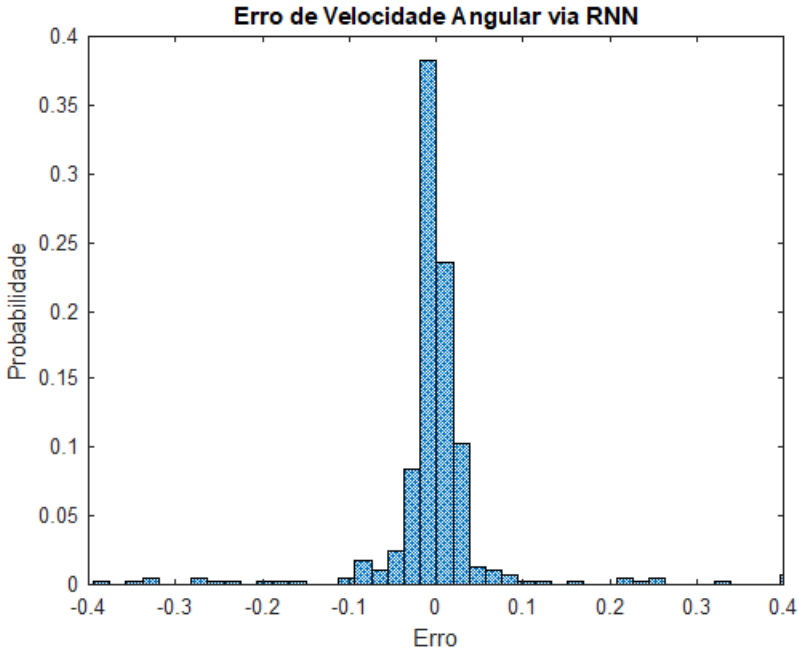

Figura 7.27: Distribuição de probabilidade do erro para ω_z

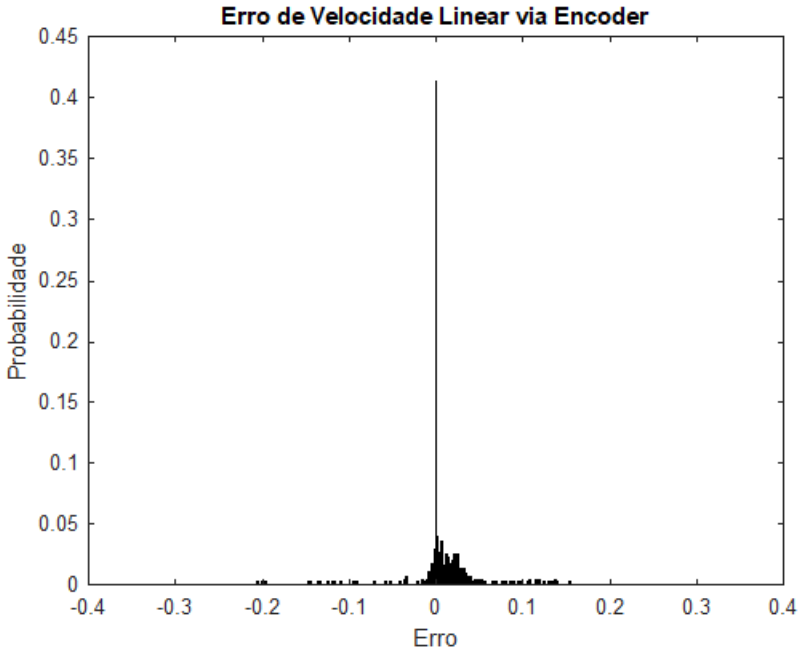

Figura 7.28: Distribuição de probabilidade do erro para V_l

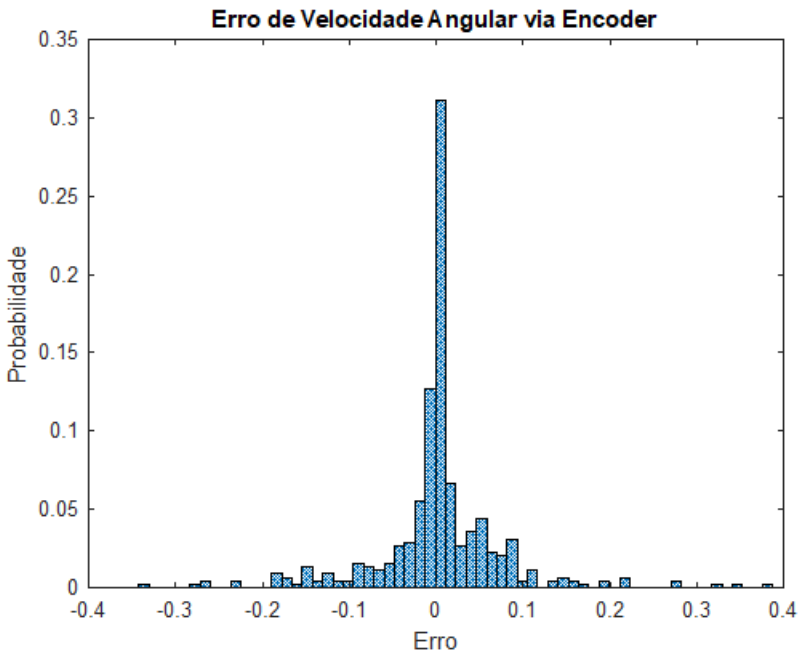

Figura 7.29: Distribuição de probabilidade do erro para ω_z

É notável, na observação dos histogramas acima, que para os erros de velocidade linear via redes neurais e *encoder*, ambas as distribuições são assimétricas, ainda que a estimativa pelo *encoder* tenha um pico mais alto próximo de zero. Para velocidade angular, as distribuições ficaram mais semelhantes a uma gaussiana, onde os maiores picos próximos de zero são dados pela estimativa via RNR.

7.3.2 Trajetória de Teste 2

A trajetória deste teste é composta por segmentos retos, com curvas longas em movimento e giros em torno do próprio eixo. O deslocamento é mostrado na Figura 7.30, a orientação na Figura 7.31 e a trajetória na Figura 7.32.

É interessante observar que, para o deslocamento, a odometria inercial se manteve concordante com as estimativas do *encoder* e do *Ground Truth* até aproximadamente $25s$, porém, um erro de estimativa neste ponto propagou-se de forma persistente para os próximos intervalos de tempo, ainda que as demais estimativas de deslocamento tenham sido coerentes.

A Tabela 7.5 mostra os erros *RMS* de cada variável estimada, sendo possível observar que o modelo cinemático com *encoder* teve melhor performance na estimativa de deslocamento, enquanto a odometria inercial teve melhor estimativa de orientação, posição x e posição y da trajetória. A tabela também mostra a forte correlação das variáveis estimadas pelas redes neurais com o *ground truth*.

Nessa trajetória, ainda que o erro *RMS* de deslocamento dado pela odometria inercial tenha sido consideravelmente maior que o erro dado pela odometria de rodas, o erro de trajetória dado pela abordagem proposta continua menor devido o erro da orientação ser menor.

Tabela 7.5: Erros RMS do teste 2

Variável	Erro RMS (IMU+RN)	Erro RMS (Encoder+Cinemática)	Correlação (IMU+RN)
Deslocamento (m)	0.3283	0.0322	0.9989
Orientação (rad)	0.1472	0.1864	0.9988
Posição em X (m)	0.1055	0.2178	0.9912
Posição em Y (m)	0.0722	0.2360	0.9879

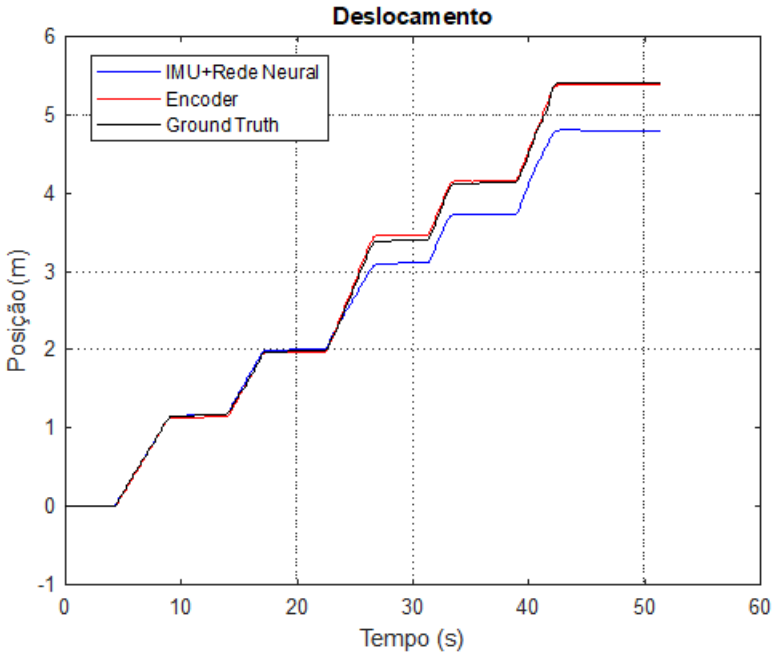

Figura 7.30: Deslocamento estimado no teste 2

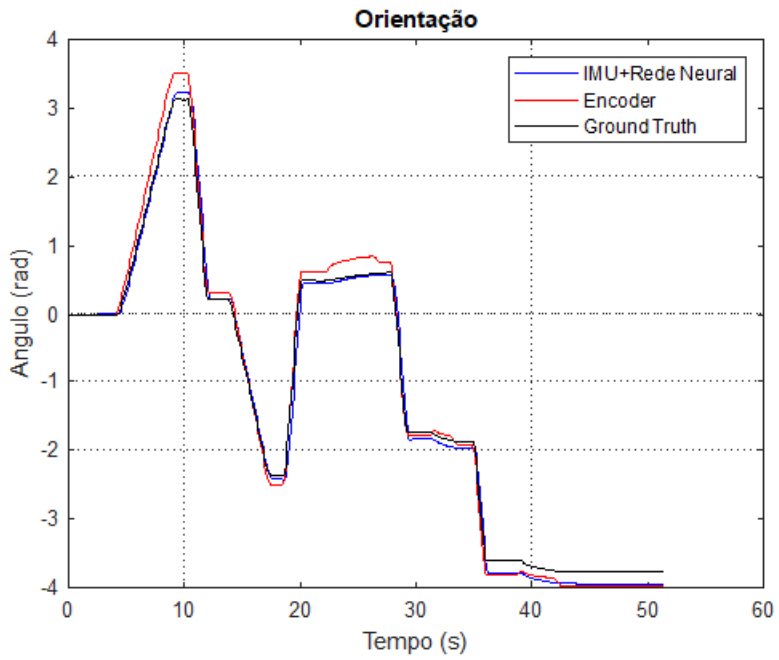

Figura 7.31: Orientação estimada no teste 2

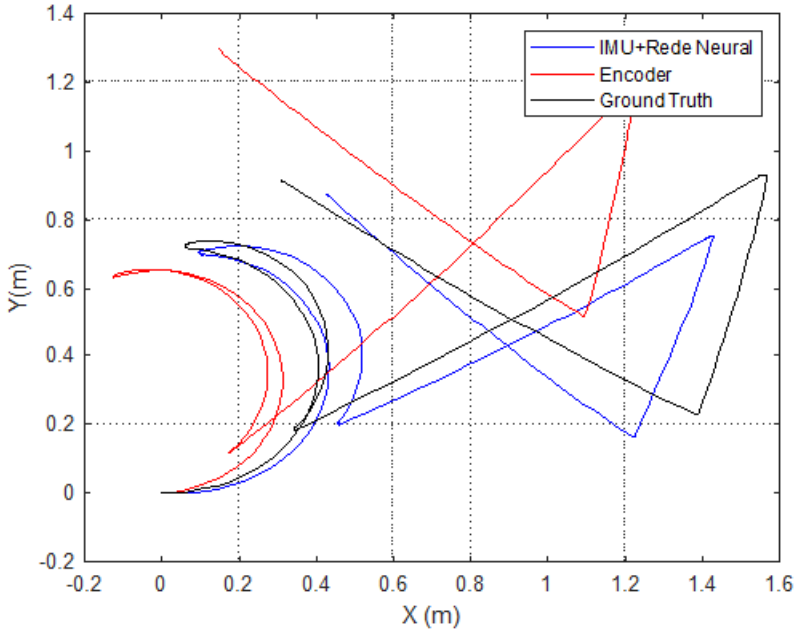

Figura 7.32: Trajetória estimada no teste 2

Nas Figuras 7.33 e 7.34, pode-se observar as estimativas de deslocamento e orientação, com

os diferentes métodos considerados para o teste 2. Assim como no experimento anterior, é possível ver a rápida divergência tanto para deslocamento quanto para orientação via integração dos dados brutos da IMU. Novamente, as estimativas de velocidade linear e angular dadas pela rede puderam ser integradas para estimar deslocamento e orientação, que se mantiveram confinadas nas proximidades dos seus *ground truths*.

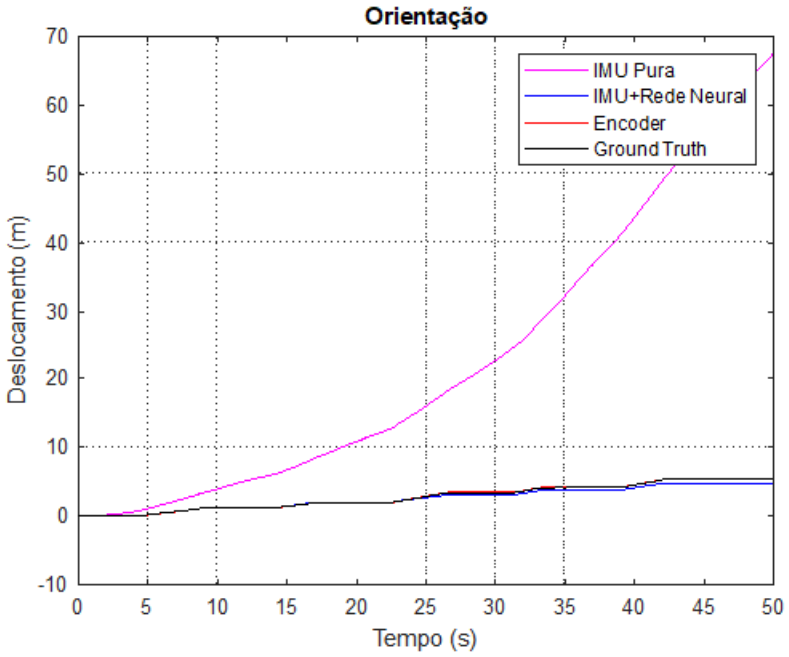

Figura 7.33: Deslocamento estimado com integração direta do sinal do acelerômetro

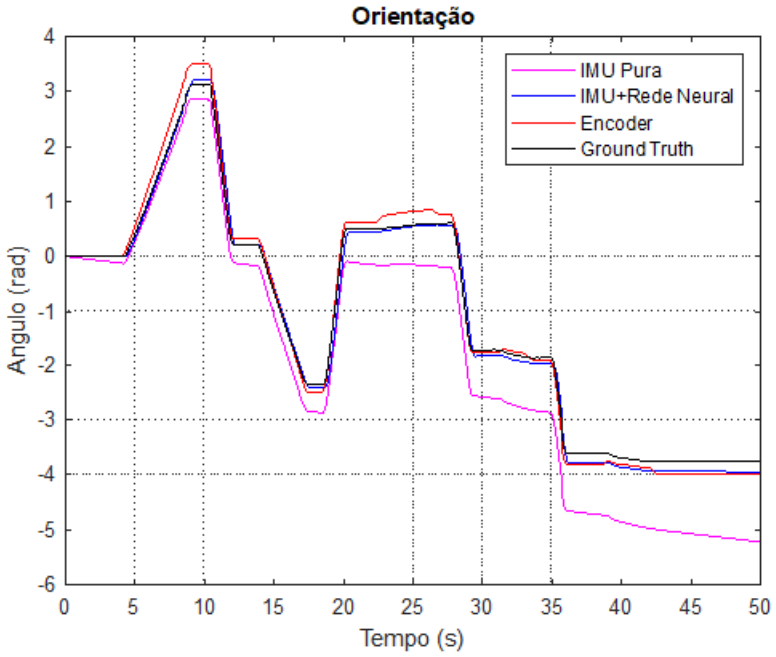

Figura 7.34: Orientação estimada com integração direta do sinal do giroscópio

O histograma da densidade de probabilidade dos erros das redes, expressos nas Equações 5.16 e 5.17 para velocidade linear V_l e angular ω_z é mostrada nas Figuras 7.35 e 7.36 e para o estimador via *encoder* com modelo cinemático nas Figuras 7.37 e 7.38, respectivamente.

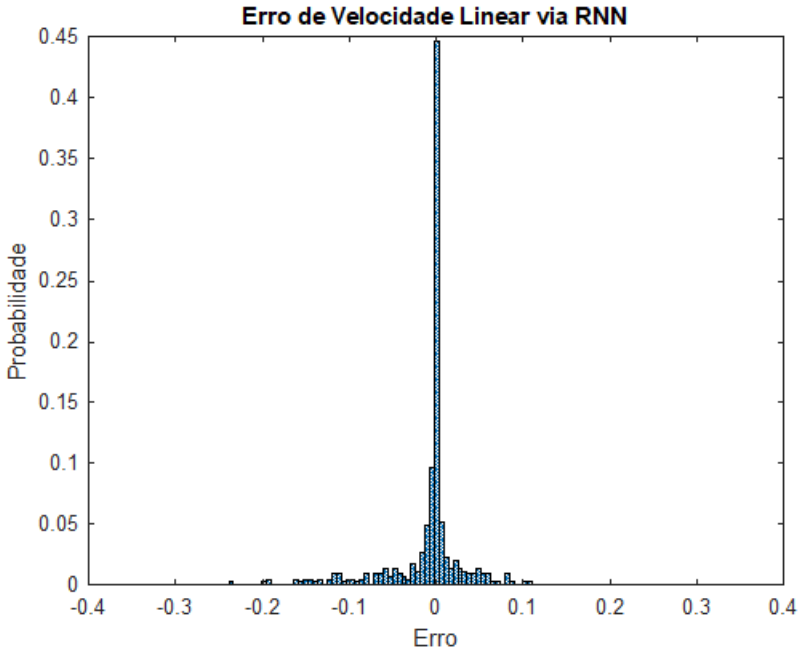

Figura 7.35: Distribuição de probabilidade do erro para V_l

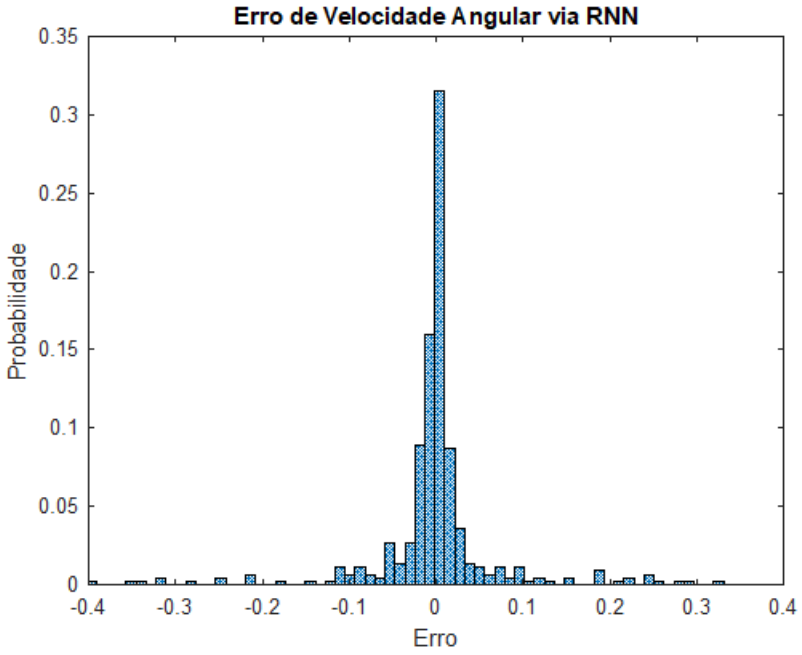

Figura 7.36: Distribuição de probabilidade do erro para ω_z

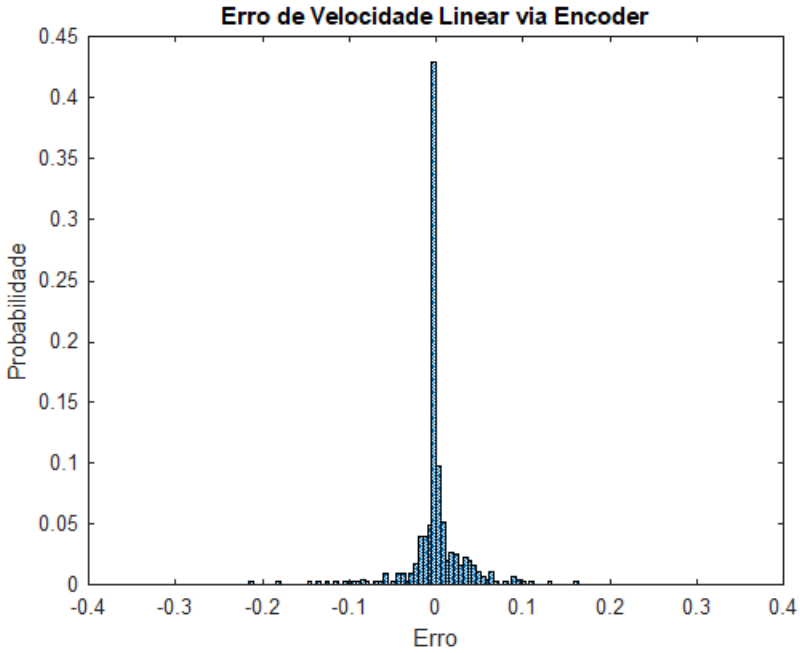

Figura 7.37: Distribuição de probabilidade do erro para V_l

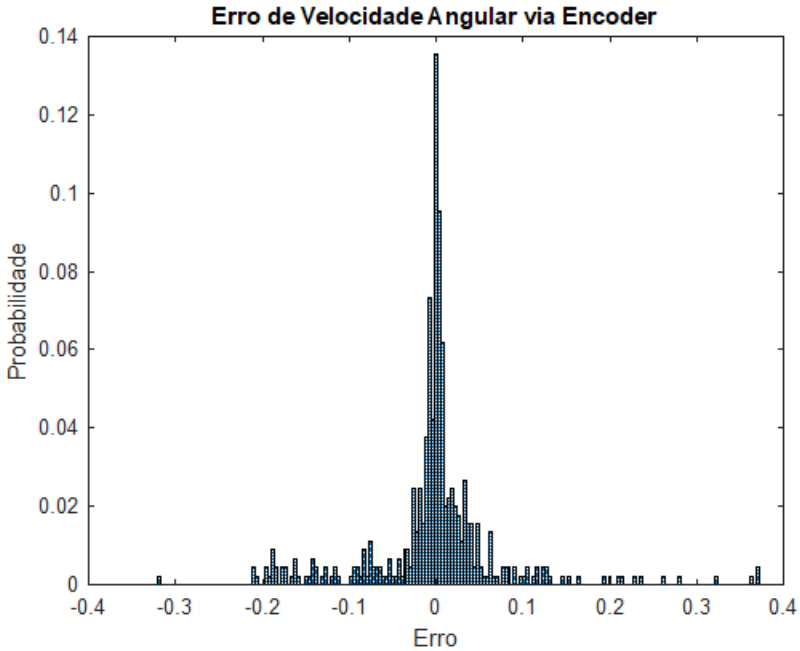

Figura 7.38: Distribuição de probabilidade do erro para ω_z

Para esta trajetória, a distribuição dos erros para velocidades lineares se mantiveram mais

simétricos, assim como a distribuição dos erros para orientação. Apesar do erro de deslocamento apontado anteriormente referente ao instante próximo de $25s$ na Figura 7.30, a distribuição de velocidade dada pela rede neural teve maiores picos próximos de zero do que a velocidade dado pelo *encoder*, o que sugere que neste ponto específico houve um erro de estimativa de velocidade, porém, ao longo da trajetória este erro foi menor do que a odometria de rodas.

7.3.3 Considerações para Aplicação em Fusão Sensorial

Como apresentado no Estado da Arte, algumas pesquisas envolvem a fusão de dados entre uma IMU com outros sensores para realizar odometria ou localização, portanto, vale considerar o método de odometria inercial proposto nesta tese como candidata para fusão de dados utilizando filtro de Kalman.

Nas seções 7.3.1 e 7.3.2, onde foram mostradas as trajetórias de teste, foram apresentadas as funções densidade de probabilidade dos erros de velocidade linear e angular, tanto para os estimadores via redes neurais quanto para os *encoders* com modelo cinemático. Esses sinais de erro foram submetidos ao teste de Shapiro-Wilk; porém, todas as hipóteses nulas de normalidade foram rejeitadas, o que implica que as medições não são, a rigor, gaussianas.

Sobre a possibilidade de fusão sensorial, existem literaturas que aplicam inferência bayesiana para deduzir o filtro de Kalman, o que podem sugerir que as variáveis envolvidas precisam ser gaussianas (Cao e Schwartz, 2004), porém, a aplicação prática do filtro no contexto de fusão sensorial é adequada ainda que os sinais das medidas envolvidas não sejam gaussianas (Pei et al., 2019). Dessa forma, as estimativas dadas pela odometria inercial pelo método proposto são potenciais candidatas para aplicações em robótica que envolvam a fusão de sensores via filtro de Kalman para realizar localização ou odometria.

Para efeito de demonstração, a Figura 7.39 mostra a trajetória referente ao teste 1 realizado na Seção 7.3 aplicando filtro de Kalman estendido para fazer fusão entre a velocidade linear estimadas pelo *encoder* e pelas redes neurais. O erro RMS no eixo x e y calculado foi 0.0740 m e 0.0697 m. Apesar do erro em x ser um pouco maior do que pela abordagem dada pela rede neural, o erro em y foi consideravelmente menor, gerando em um erro de trajetória resultante pelas componentes x e y menor que as duas abordagens individuais.

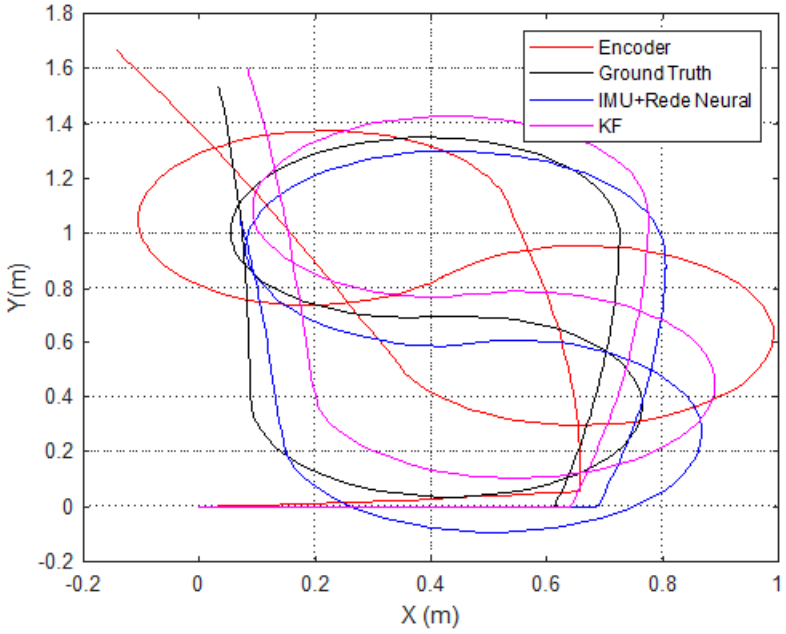

Figura 7.39: Trajetória do teste 1 com filtro de kalman

7.4 Link para dataset

O seguinte endereço dá acesso ao *dataset* utilizado na realização deste trabalho : `https://github.com/engpedroramon/legacyrobot`

8 Conclusões

Esta pesquisa é um dos estudos em odometria inercial desenvolvido no grupo de automação e navegação robótica do Laboratório de Processamento de Sinais e Análise de Sistemas Dinâmicos do Departamento de Sistemas Integrados da Faculdade de Engenharia Mecânica da UNICAMP.

Com base no levantamento bibliográfico, observou-se que diversas pesquisas foram realizadas sobre os desafios das técnicas de odometria inercial, em geral, e especialmente aplicada na robótica móvel. Com isso, houve a necessidade de levantar os principais problemas de se utilizar uma *IMU* para realizar estimativa de deslocamento e orientação de um robô. Nesta pesquisa, constatou-se a problemática relacionada às perturbações nas medidas do acelerômetro e giroscópio que causam divergência em curto prazo na estimativa da pose de um robô.

Como diferencial das demais pesquisas, nesta tese foi apresentada uma abordagem baseada nas medições de uma *IMU* em complemento com sinais de atuação sendo processadas em uma rede neural de Elman para se obter a pose do robô ao longo de uma trajetória. Esta técnica permitiu resultados equivalentes ou superiores a trajetória estimada por *encoders* com o modelo cinemático, em especial para estimativa de orientação. A técnica proposta se apresentou robusta aos possíveis erros não-sistemáticos presentes no método de odometria de rodas.

É oportuno mencionar que a metodologia proposta foi validada utilizando sensores de baixa resolução, baixa taxa de amostragem dos sinais medidos e redes recorrentes de Elman com um *dataset* modesto de $1.4MB$ para treinamento da mesma. Além disso, importa destacar que nenhum trabalho na literatura, de acordo com a pesquisa desenvolvida, implementou odometria inercial em um robô móvel com trajetórias mais complexas em ambiente *indoor* utilizando poucos recursos tanto instrumentais quanto computacionais. Foi constatado que a maioria das abordagens de odometria inercial utiliza: *IMU* em fusão com outros sensores exteroceptivos (câmera, LIDAR, etc); redes neurais profundas para rastreamento de pedestres; ou, utiliza-se um instrumental de alto custo, quando mostrado em veículo com rodas.

Apesar das estimativas de pose do robô tenham sido feitas *offline* no MATLAB, as mesmas são factíveis de serem implementadas em um sistema embarcado, pois ambas as redes utilizadas nos experimentos têm somente 10 neurônios com apenas uma camada. Redes neurais artificiais são implementadas matematicamente como matrizes que por sua vez ocupam espaço de armaze-

namento, assim, quanto menor a rede neural mais conveniente a mesma se torna para ser implementada em sistemas com pouca memória de programa como em microcontroladores, portanto, as redes treinadas demostradas no Capítulo 7 contemplam esta vantagem. Além disso, a complexidade matemática da rede recorrentes de Elman é consideravelmente menor do que redes recorrentes modernas do tipo LSTM e GRU.

Os experimentos foram realizados com um robô de baixo custo construído especificamente para esta pesquisa, visto que não foi encontrado um *dataset* disponível com dados compostos por IMU e corrente ou torque dos motores em um robô. Portanto, a plataforma construída permitiu a construção de um *dataset*, como contribuição para pesquisas posteriores na área.

8.1 Propostas para Trabalhos Futuros

- Implementar as redes neurais treinadas *offline* em um sistema embarcado para realizar a odometria inercial em tempo real.
- Robôs aspiradores domésticos de baixo custo realizam mapeamento binário do ambiente utilizando *bumpers* quando acontecem colisões em algum obstáculo. De modo análogo com o robô utilizado, pode-se realizar o mapeamento binário baseado nos sinais de acelerômetro em vez das chaves mecânicas utilizadas nos *bumpers*; assim, será possível implementar localização e mapeamento pelo método apresentado nesta tese com um robô de baixo custo.
- Utilizar a rede neural para realizar detecção de escorregamento em odometria com rodas.

Referências

Aggarwal, P., Syed, Z., e El-Sheimy, N. (2008). Thermal calibration of low cost mems sensors for land vehicle navigation system. In *Vtc spring 2008-ieee vehicular technology conference* (pp. 2859–2863).

Alajlan, N. N., e Ibrahim, D. M. (2022). Tinyml: Enabling of inference deep learning models on ultra-low-power iot edge devices for ai applications. *Micromachines*, *13*(6), 851.

Al Seyab, R., e Cao, Y. (2008). Nonlinear system identification for predictive control using continuous time recurrent neural networks and automatic differentiation. *Journal of Process Control*, *18*(6), 568–581.

Aqel, M. O., Marhaban, M. H., Saripan, M. I., e Ismail, N. B. (2016). Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, *5*(1), 1–26.

Asoodeh, M., Bagheripour, P., e Gholami, A. (2015). Nmr parameters determination through ace committee machine with genetic implanted fuzzy logic and genetic implanted neural network. *Acta Geophysica*, *63*(3), 735–760.

Bastos, V. B., Tavares, A. F. P., Quiroz, C. H. C., Silva, P. R. M., Lima, M. V. P., e Kurka, P. R. G. (2018, April). Monocular visual odometry for robotic wheelchair in a virtual environment. In *2018 annual ieee international systems conference (syscon)* (p. 1-8). doi: 10.1109/SYSCON.2018.8369569

Borenstein, J., e Feng, L. (1995). Correction of systematic odometry errors in mobile robots. In *Proceedings 1995 ieee/rsj international conference on intelligent robots and systems. human robot interaction and cooperative robots* (Vol. 3, pp. 569–574).

Brossard, M., Barrau, A., e Bonnabel, S. (2020). *Rins-w: Robust inertial navigation system on wheels.*

Cai, J., Yu, R., e Cheng, L. (2012). Autonomous navigation research for mobile robot. In *Proceedings of the 10th world congress on intelligent control and automation* (pp. 331–335).

Cao, L., e Schwartz, H. M. (2004). Analysis of the kalman filter based estimation algorithm: an orthogonal decomposition approach. *Automatica*, *40*(1), 5–19.

Debeunne, C., e Vivet, D. (2020). A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors*, *20*(7), 2068.

Delgado, J., Silva, P. R., Quiroz, C., e Kurka, P. R. G. (2016). Automatic calibration of low cost inertial gyroscopes with a ptu. In *2016 ieee 19th international conference on intelligent transportation systems (itsc)* (pp. 121–125).

Driss, S. B., Soua, M., Kachouri, R., e Akil, M. (2017). A comparison study between mlp and convolutional neural network models for character recognition. In *Real-time image and video processing 2017* (Vol. 10223, p. 1022306).

Ferrero, R., Gandino, F., Hemmatpour, M., Montrucchio, B., e Rebaudengo, M. (2016). Exploiting accelerometers to estimate displacement. In *2016 5th mediterranean conference on embedded computing (meco)* (pp. 206–210).

Fourati, H. (2015). Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter. *IEEE Transactions on Instrumentation and Measurement*, *64*(1), 221–229.

Gui, P., Tang, L., e Mukhopadhyay, S. (2015). Mems based imu for tilting measurement: Comparison of complementary and kalman filter based data fusion. In *2015 ieee 10th conference on industrial electronics and applications (iciea)* (pp. 2004–2009).

Gutmann, J.-S., e Fox, D. (2002). An experimental comparison of localization methods continued. In *Intelligent robots and systems, 2002. ieee/rsj international conference on* (Vol. 1, pp. 454–459).

Hazry, D., e Mohd Sofian, M. R. (2009). Study of inertial measurement unit sensor.

Hewamalage, H., Bergmeir, C., e Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427.

Ibraheem, K. (2017). System identification of thermal process using elman neural networks with no prior knowledge of system dynamics. *Int. J. Comput. Appl*, *161*, 38–46.

Ivanjko, E., Petrinic, T., e Petrovic, I. (2010). Modelling of mobile robot dynamics. In *7th eurosim congress on modelling and simulation* (Vol. 2).

Khaldi, R., Chiheb, R., e El Afia, A. (2018). Feedforward and recurrent neural networks for time series forecasting: comparative study. In *Proceedings of the international conference on learning and optimization algorithms: Theory and applications* (pp. 1–6).

Kim, W.-Y., Seo, H.-I., e Seo, D.-H. (2021). Nine-axis imu-based extended inertial odometry neural network. *Expert Systems with Applications*, *178*, 115075.

Kordos, M., e Rusiecki, A. (2016). Reducing noise impact on mlp training. *Soft Computing*, *20*(1), 49–65.

Lee, S., Lee, D., Choi, P., e Park, D. (2020). Accuracy–power controllable lidar sensor system with 3d object recognition for autonomous vehicle. *Sensors*, *20*(19), 5706.

Li, M., e Mourikis, A. I. (2013). High-precision, consistent ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, *32*(6), 690–711.

Li, S., Wu, C., Li, H., Li, B., Wang, Y., e Qiu, Q. (2015). Fpga acceleration of recurrent neural network based language model. In *2015 ieee 23rd annual international symposium on field-programmable custom computing machines* (pp. 111–118).

Lima, M. V., Silva, P. R., Quiroz, C. H., e Kurka, P. R. (2017). Neural network regularization of an inertial odometry estimation for position control of a mobile robot. In *2017 chilean conference on electrical, electronics engineering, information and communication technologies (chilecon)* (pp. 1–6).

Lynen, S., Achtelik, M. W., Weiss, S., Chli, M., e Siegwart, R. (2013). A robust and modular multi-sensor fusion approach applied to mav navigation. In *Intelligent robots and systems (iros), 2013 ieee/rsj international conference on* (pp. 3923–3929).

Mahrami, M., Islam, M. N., e Karimi, R. (2013). Simultaneous localization and mapping: issues and approaches. *International Journal of Computer Science and Telecommunications*, *4*(1).

Pei, Y., Biswas, S., Fussell, D. S., e Pingali, K. (2019). An elementary introduction to kalman filtering. *Communications of the ACM*, *62*(11), 122–133.

Rezk, N. M., Purnaprajna, M., Nordström, T., e Ul-Abdin, Z. (2020). Recurrent neural networks: An embedded computing perspective. *IEEE Access*, *8*, 57967–57996.

Richard, A., Mahé, A., Pradalier, C., Rozenstein, O., e Geist, M. (2019). A comprehensive benchmark of neural networks for system identification.

Rollinson, D., Choset, H., e Tully, S. (2013). Robust state estimation with redundant proprioceptive sensors. In *Asme 2013 dynamic systems and control conference* (pp. V003T40A005–V003T40A005).

Sak, H., Senior, A., Rao, K., e Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.

Samatas, G. G., e Pachidis, T. P. (2022). Inertial measurement units (imus) in mobile robots over the last five years: A review. *Designs*, *6*(1), 17.

Sariff, N., e Buniyamin, N. (2006). An overview of autonomous mobile robot path planning algorithms. In *Research and development, 2006. scored 2006. 4th student conference on* (pp. 183–188).

Seo, W., e Baek, K.-R. (2017). Indoor dead reckoning localization using ultrasonic anemometer with imu. *Journal of Sensors*, *2017*.

Shabbir, J., e Anwer, T. (2018). A survey of deep learning techniques for mobile robot applications. *arXiv preprint arXiv:1803.07608*.

Sheela, K. G., e Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical problems in engineering*, *2013*.

Silva, P. R. d. M., et al. (2015). Localização de robô em ambiente interno utilizando um dispositivo móvel baseado no sistema operacional android para navegação inercial.

Slabaugh, G. G. (1999). Computing euler angles from a rotation matrix. *Retrieved on August*, *6*(2000), 39–63.

Solin, A., Cortes, S., Rahtu, E., e Kannala, J. (2018). Inertial odometry on handheld smartphones. In *2018 21st international conference on information fusion (fusion)* (pp. 1–5).

Su, H. T., e McAvoy, T. J. (1997). Artificial neural network for nonlinear process identification and control. In *Nonlinear process control* (pp. 371–428).

Tekin, Y., Tümsavas, Z., e Mouazen, A. M. (2014). Comparing the artificial neural network with parcial least squares for prediction of soil organic carbon and ph at different moisture content levels using visible and near-infrared spectroscopy. *Revista Brasileira de Ciência do Solo*, *38*(6), 1794–1804.

Tereshkov, V. M. (2013). An intuitive approach to inertial sensor bias estimation. *International Journal of Navigation and Observation*, *2013*.

Toha, S. F., e Tokhi, M. O. (2008). Mlp and elman recurrent neural network modelling for the trms. In *2008 7th ieee international conference on cybernetic intelligent systems* (pp. 1–6).

Ullah, I., Shen, Y., Su, X., Esposito, C., e Choi, C. (2019). A localization based on unscented kalman filter and particle filter localization algorithms. *IEEE Access*, *8*, 2233–2246.

Yuan, W., Li, Z., e Su, C.-Y. (2016). Rgb-d sensor-based visual slam for localization and navigation of indoor mobile robot. In *Advanced robotics and mechatronics (icarm), international conference on* (pp. 82–87).

Zhang, Z., Tang, Z., Gao, S., e Yang, G. (2010). Training elman neural network for dynamic system identification using an adaptive local search algorithm. *International Journal of Innovative Computing, Information and Control*, *6*(5), 2233–2243.

Zotov, S. A., Rivers, M. C., Trusov, A. A., e Shkel, A. M. (2010). Chip-scale imu using folded-mems approach. In *Sensors, 2010 ieee* (pp. 1043–1046).