



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DA PARAÍBA
CURSO DE BACHARELADO EM ENGENHARIA DE
COMPUTAÇÃO

DAVI JUVÊNIO GOMES DE SOUSA

CONTROLE DE NAVEGAÇÃO DE UM ROBÔ MÓVEL

CAMPINA GRANDE - PB
2022

DAVI JUVÊNIO GOMES DE SOUSA

CONTROLE DE NAVEGAÇÃO DE UM ROBÔ MÓVEL

Trabalho de Conclusão de Curso apresentado ao Programa de Graduação em Engenharia de Computação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Moacyr Pereira da Silva

CAMPINA GRANDE - PB
2022

S725c Sousa, Davi Juvêncio Gomes de.

Controle de navegação de um robô móvel / Davi Juvêncio Gomes de Sousa. - Campina Grande, 2022.

36 f. : il.

Trabalho de Conclusão de Curso (Curso de Graduação em Engenharia de Computação) - Instituto Federal da Paraíba, 2022.

Orientador: Prof. Dr. Moacy Pereira da Silva.

1. Controle de navegação 2. Controle de orientação
3. Robô móvel I. Silva, Moacy Pereira da II. Título.

CDU 681.5

Dedico este trabalho de conclusão de curso à minha bisavó Odete Francisca (em memória), que sempre fez o possível e o impossível para que eu me dedicasse aos estudos, ótimos conselhos que vou levar para toda minha vida e incentivos para realizações dos meus sonhos.

Agradecimentos

- Em primeiro lugar, gostaria de agradecer a Deus, que sempre esteve ao meu lado durante toda essa caminhada e em todos os momentos de minha vida.
- Agradeço à minha mãe Cassiana Gomes por me apoiar, escutar os meus lamentos e sempre procurar ver o lado bom das situações.
- Agradeço ao professor Moacyr Pereira, pela orientação nas etapas do projeto.
- Agradeço ao professor Fagner de Araújo pelos ensinamentos da disciplina que resultou na conclusão deste projeto.
- Ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba que além de tornar possível minha graduação em Engenharia de Computação, fez com que eu sentisse que o IFPB era minha casa.
- Agradeço à Renata da Silva por sempre estar ao meu lado me apoiando e me auxiliando nos perrengues que o curso e a vida proporcionaram.

“Eu não tenho nenhum talento especial. Apenas sou apaixonadamente curioso.” -Albert Einstein

CONTROLE DE NAVEGAÇÃO DE UM ROBÔ MÓVEL

Resumo

O problema a ser tratado neste TCC é a navegação de um robô móvel em ambientes semi-estruturados, utilizando a arquitetura de controle reativa e um sensor de varredura. A ideia principal é fazer com que o robô navegue entre dois pontos do ambiente, que é o objetivo principal de robôs de exploração, mineração, limpeza de acidentes nucleares, navegação em estradas e em tarefas agrícolas. É desenvolvido um controle de posição e de orientação fundamentado na estrutura de um robô baseado no unicycle. Ressalta-se que as informações sobre o estado do robô provêm unicamente dos sensores a bordo do robô. A estabilidade dos controladores foi comprovada pela convergência assintótica das variáveis a partir da função candidata de Lyapunov. Foi concebido uma nova abordagem de controle para que melhor atendesse a necessidade dos experimentos. O sistema foi testado utilizando o simulador CoppeliaSim junto com o ROS, caracterizando um sistema Model-in-the-Loop, e implementado em Python e MATLAB. Foi iniciado a implementação em um robô real, restando um sistema de rastreamento de movimento para testar, contribuindo para trabalhos futuros que queiram seguir a linha de pesquisa. Com os resultados dos experimentos com o controlador proposto, foi possível comprovar a convergência assintótica das variáveis controladas para seus valores desejados, resultando, assim, na navegação desejada. Foi possível constatar que o controle proposto originalmente, necessitava de melhorias, e essas melhorias reduziram drasticamente o tempo de execução da missão. A nova lei de controle proposta nesse trabalho de conclusão de curso se mostrou bastante eficaz, reduzindo drasticamente o tempo de execução da missão de navegação.

Palavras-chave: Controle de orientação. Controle de posição. Controle de navegação. ROS. Robô móvel. Modelagem.

NAVIGATION CONTROL OF A MOBILE ROBOT

Abstract

The problem to be treated in this TCC is the navigation of a mobile robot in semi-structured environments, using the reactive control architecture and a scanning sensor. The main idea is to make the robot navigate between two points in the environment, which is the main objective of robots for exploration, mining, cleaning up nuclear accidents, navigation on roads and in agricultural tasks. A position and orientation control based on the structure of a unicycle-based robot is developed. It should be noted that the information about the state of the robot comes only from the sensors on board the robot. The stability of the controllers was proved by the asymptotic convergence of the variables from the Lyapunov candidate function. A new control approach was designed to better meet the needs of the experiments. The system was tested using the CoppeliaSim simulator together with ROS, featuring a Model-in-the-Loop system, and implemented in Python and MATLAB. The implementation in a real robot was started, leaving a movement tracking system to test, contributing to future works that want to follow the line of research. With the results of the experiments with the proposed controller, it was possible to prove the asymptotic convergence of the controlled variables to their desired values, thus resulting in the desired navigation. It was possible to verify that the originally proposed control needed improvements, and these improvements drastically reduced the mission execution time. The new control law proposed in this course completion work proved to be quite effective, drastically reducing the execution time of the navigation mission.

Keywords: Orientation control. Position control. Navigation control. ROS. Mobile robot. Modeling.

Lista de Figuras

Figura 1 – Arquitetura da abordagem Reativa.	2
Figura 2 – Pioneer 3-DX.	4
Figura 3 – Robô Móvel de Acionamento Diferencial.	5
Figura 4 – Restrição de movimento de rolamento puro.	7
Figura 5 – Sistema de coordenadas para projeto do controlador de posição final.	10
Figura 6 – Pioneer 3-DX no simulador CoppeliaSim.	14
Figura 7 – Interface do simulador CoppeliaSim.	15
Figura 8 – Ícone de acesso a interface de programação em LUA.	16
Figura 9 – <i>Model-in-the-Loop</i>	17
Figura 10 – Caminho percorrido pelo robô móvel.	17
Figura 11 – Plataforma Robótica GoPiGo3.	18
Figura 12 – Coppelasim com ROS Experimento 1.	19
Figura 13 – Métricas do Experimento 1.	20
Figura 14 – Trajetória do Experimento 1.	20
Figura 15 – Coppelasim com ROS Experimento 2.	21
Figura 16 – Métricas do Experimento 2.	21
Figura 17 – Trajetória do Experimento 2.	22
Figura 18 – Coppelasim com ROS Experimento 3.	22
Figura 19 – Métricas do Experimento 3.	23
Figura 20 – Trajetória do Experimento 3.	23
Figura 21 – Métricas do Experimento 4 com $k_v = 2.4$	24
Figura 22 – Trajetória do Experimento 4 com $k_v = 2.4$	24
Figura 23 – Métricas do Experimento 4 com $k_v = 9.6$	25
Figura 24 – Trajetória do Experimento 4 com $k_v = 9.6$	25
Figura 25 – Robô Real.	26

Sumário

1 – Introdução	1
1.1 Robôs Móveis	1
1.2 Definição do Problema	2
1.3 Objetivos	2
1.3.1 Objetivo geral	2
1.3.2 Objetivos específicos	2
1.4 Estrutura do TCC	3
2 – Fundamentação Teórica	4
2.1 Sistemas de Coordenadas	5
2.2 Restrições Cinemáticas do Robô de Acionamento Diferencial	6
2.3 Modelo Cinemático	8
2.4 Controladores de Posição e de Orientação	9
2.4.1 Controlador de Posição	9
2.4.2 Controlador de Orientação	11
2.4.3 Estabilidade de Sistemas Comutados	13
2.5 Simulador CoppeliaSim	13
3 – Metodologia	15
3.1 Interface de Desenvolvimento e Simulação	15
3.2 Programação	16
3.3 Desenvolvimento da Simulação	16
3.4 Nova Abordagem de Controle	18
3.5 Implementação em um Robô Real	18
4 – Resultados e Discussão	19
4.1 Experimento 1 - Quadrante Positivo	19
4.2 Experimento 2 - Quadrante Negativo	19
4.3 Experimento 3 - Quadrante Misto	21
4.4 Experimento 4 - Implementação da nova lei de controle	22
4.5 Experimento 5 - Implementação em um Robô Real	24
5 – Conclusão	27
5.1 Trabalhos Futuros	27
Referências	28

1 Introdução

A robótica é uma área do saber que conjuga conhecimentos de engenharia mecânica, com o estudo da estática e a dinâmica, de matemática, com a descrição dos modelos dos robôs e do movimento no espaço, de engenharia eletrônica, com o projeto de sensores e interfaces para os robôs, de teoria de controle, com o projeto de algoritmos para que o robô realize os movimentos desejados, e, ainda, da computação, com a programação de todos os algoritmos desenvolvidos para que a tarefa designada para o robô possa ser realizada. Em robótica, a navegação é um problema prático que muitas vezes envolve o deslocamento de agentes na presença de objetos, tais como paredes, obstáculos diversos ou mesmo outros robôs (([MOHANAN; SALGOANKAR, 2018](#)), ([VILLANI et al., 2018](#)), ([ZACHARAKI et al., 2020](#))).

Em um ambiente semi-estruturado, admite-se um certo grau de mobilidade dos objetos presentes no meio. Escritórios, oficinas, laboratórios e residências são exemplos típicos de ambientes semi-estruturados, os quais sugerem a adoção de abordagem reativa. Devido a esse tipo de raciocínio a ser implementado em um robô móvel, a abordagem que será utilizada neste TCC será a reativa, que não utiliza nenhum conhecimento prévio do meio, a não ser alguma informação muito geral como o conhecimento de que o ambiente é plano e fechado. Por este motivo não é possível planejar uma trajetória para o robô, pois não existe um mapa do ambiente. A navegação baseada na abordagem reativa foca o fato de que as percepções estão intimamente relacionadas com as ações, isto é, o robô reage ao que percebe. Isto resulta em simplicidade e em um baixo custo computacional, duas qualidades importantes em robótica móvel. Assim, cada ação tomada por um robô que navega com algoritmos reativos é produzida com base nas informações sensoriais. Essa característica permite que um robô navegue em ambientes onde os obstáculos podem mudar de posição, já que o robô pode percebê-los e reagir. O esquemático da ideia geral da abordagem reativa está representado na Figura 1.

É importante ressaltar que é desejado a implementação de algoritmos de desvio de obstáculos baseados na arquitetura de controle reativa, ou seja, o robô deve “sentir” o ambiente e reagir de forma que a tarefa que lhe foi designada possa ser realizada.

1.1 Robôs Móveis

Os robôs móveis, por definição, são dispositivos mecânicos montados sobre uma base não fixa, capazes de interagir com o ambiente à sua volta através de seus sensores e atuadores, os quais são controlados por um sistema computacional normalmente a bordo ([ALATISE; HANCKE, 2020](#)). Os robôs móveis são classificados quanto à sua anatomia,

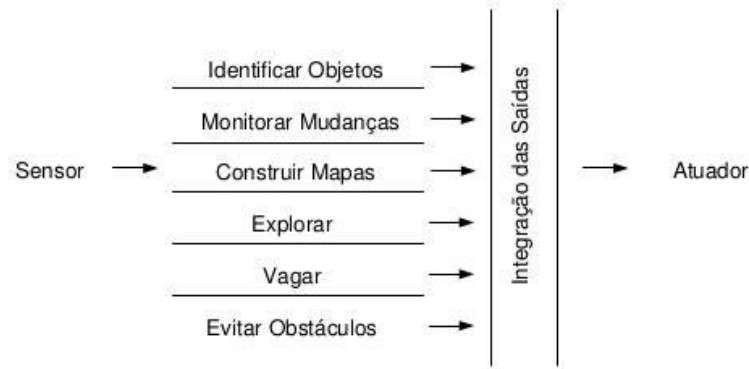


Figura 1 – Arquitetura da abordagem Reativa.

ou seja, em função do ambiente no qual eles foram projetados para navegar. Neste sentido, eles podem ser terrestres, aquáticos ou aéreos.

1.2 Definição do Problema

O principal problema a ser tratado neste TCC é a navegação de um robô móvel em ambientes semi-estruturados, utilizando a arquitetura de controle reativa e um sensor de varredura. A ideia principal é fazer com que o robô navegue entre dois pontos do ambiente, que é o objetivo principal de robôs de exploração, mineração, limpeza de acidentes nucleares, navegação em estradas e em tarefas agrícolas.

1.3 Objetivos

O presente trabalho visa desenvolver um controle de posição e de orientação baseado na estrutura de um robô baseado no uniciclo. Ressalta-se que as informações sobre o estado do robô provêm unicamente dos sensores a bordo do robô.

1.3.1 Objetivo geral

O objetivo geral deste trabalho é fazer com que um robô móvel dotado de sensores locais seja capaz de identificar a posição e seguir para o ponto desejado, mantendo-se no ponto desejado uma orientação relativa pré-definidas.

1.3.2 Objetivos específicos

- (i) Estudar o modelo cinemático do robô móvel tipo uniciclo;
- (ii) Estudar o controle de posição e de orientação robô móvel;
- (iii) Validar a metodologia proposta através de simulações;

- (iv) Propor uma nova lei de controle;
- (v) Implementar a metodologia proposta em um robô real.

1.4 Estrutura do TCC

Este TCC está organizada da seguinte forma:

- Capítulo 1: Introdução

Este Capítulo descreve, em linhas gerais, o tema e o estado da arte deste TCC. São apresentados os problemas que devem ser resolvidos, assim como os objetivos a serem alcançados com este trabalho.

- Capítulo 2: Fundamentação Teórica

Este Capítulo apresenta algumas características do robô, especificamente, do sensor de varredura, o modelo cinemático do robô e um algoritmo para navegação de robôs. Além disso, é apresentado o simulador utilizado.

- Capítulo 3: Metodologia

Este Capítulo apresenta a metodologia utilizada para a execução dos experimentos e a implementação do controle utilizado.

- Capítulo 4: Resultados e Discussão

Este Capítulo apresenta os resultados dos experimentos e algumas discussões sobre os resultados obtidos.

- Capítulo 5: Conclusão

Este Capítulo apresenta a conclusão do trabalho e também discute alguns trabalhos futuros.

2 Fundamentação Teórica

Os robôs móveis autônomos se caracterizam por sua capacidade de se movimentar em ambientes estruturados ou semi-estruturados sem a intervenção de um operador humano. Para isso, o robô deve possuir sensores que adquiram informações sobre o ambiente onde o robô está navegando, para que o mesmo possa tomar as decisões necessárias para obter sucesso na realização da tarefa que lhe foi designada.

É muito frequente a utilização de sensores laser e de ultra-som (sonares) e câmaras de vídeo cuja informação permite, entre outras funções, obter informação para a criação de mapas do ambiente, localizar objetos e detectar obstáculos. Estes sensores estão presentes no robô móvel utilizado para a realização de simulações e experimentos deste TCC, o Pioneer 3-DX, da MobileRobots, apresentado na Figura 2. Neste capítulo são descritas as principais características, o modelo e o controlador do robô.



Figura 2 – Pioneer 3-DX.
Fonte: MobileRobots

O Pioneer 3-DX é um robô terrestre do tipo uniciclo, ou seja, possui duas rodas de tração diferencial independentes e uma roda castor para apoio da estrutura. Esta configuração permite que o robô gire em torno do próprio eixo, conferindo três graus de liberdade¹ ao robô. Também possuindo sensores que geram informações como posição (x,y), ângulo de guinada e velocidades linear e angular atuais (JAIEM et al., 2021). Sua comunicação com um computador externo é através da porta serial RS-232, e os sinais de controle enviados ao robô são expressos pelo vetor

$$u_d = [v_d \ \omega_d]$$

onde u_d e ω_d são respectivamente, a velocidade linear e angular desejadas.

¹ Quantidade de movimentos que o robô pode realizar no espaço bidimensional ou tridimensional.

2.1 Sistemas de Coordenadas

Para descrever a posição dos robôs móveis com rodas em seu ambiente, dois sistemas de coordenadas diferentes (*frames*) precisam ser definidos.

1. Sistema de Coordenadas Inerciais: Este sistema de coordenadas é um quadro global que é fixo no ambiente ou plano em que os robôs móveis de rodas se movem. Além disso, este quadro é considerado como o quadro de referência e é denotado como $\{X_I, Y_I\}$.
2. Sistema de Coordenadas do Robô: Este sistema de coordenadas é um quadro local conectado aos robôs móveis com rodas e, portanto, movendo-se com ele. Este quadro é denotado como $\{X_r, Y_r\}$.

As duas estruturas definidas são mostradas na Figura 3. A origem da estrutura do robô é definida como o ponto médio A no eixo entre as rodas. O centro de massa C do robô é assumido como estando no eixo de simetria, a uma distância d da origem A .

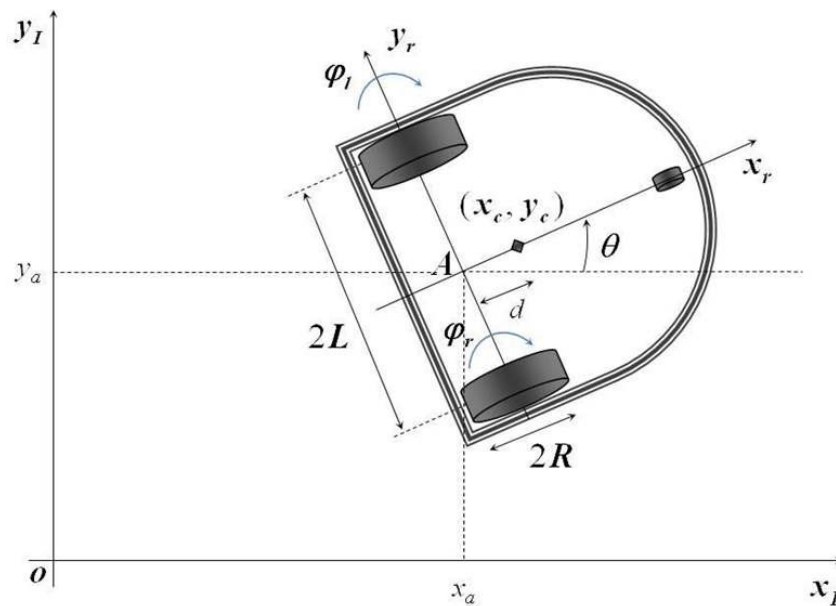


Figura 3 – Robô Móvel de Acionamento Diferencial.

Conforme mostrado na Figura 3, a posição e orientação do robô no quadro inercial podem ser definidas como

$$q^I = \begin{bmatrix} x_a \\ y_a \\ \theta \end{bmatrix} \quad (1)$$

A questão importante que precisa ser explicada nesta etapa é o mapeamento entre esses dois quadros. A posição de qualquer ponto no robô pode ser definida no quadro do robô e no quadro inercial da seguinte forma:

Sejam $X^r = \begin{bmatrix} x^r \\ y^r \\ \theta^r \end{bmatrix}$ e $X^I = \begin{bmatrix} x^I \\ y^I \\ \theta^I \end{bmatrix}$ as coordenadas do ponto dado no referencial do robô e no referencial inercial, respectivamente.

Então, as duas coordenadas são relacionadas pela seguinte transformação:

$$X^I = R(\theta)X^r \quad (2)$$

Onde $R(\theta)$ é a matriz de rotação ortogonal

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Essa transformação permitirá também a manipulação de movimento entre quadros.

$$\dot{X}^I = R(\theta)\dot{X}^r \quad (4)$$

Logo a equação (4) é muito importante na derivação dos modelos cinemáticos e dinâmicos de acionamento diferencial do robô móvel, pois descreve a relação entre as velocidades no referencial inercial e no chassis do robô.

2.2 Restrições Cinemáticas do Robô de Acionamento Diferencial

O movimento de um robô móvel com acionamento diferencial é caracterizado por duas equações de restrição não-holonômicas² que são obtidas por duas premissas principais:

- Sem movimento de deslizamento lateral: Esta restrição significa simplesmente que o robô pode se mover apenas em um movimento curvo (para frente e para trás), mas não para o lado. No quadro do robô, esta condição significa que a velocidade do ponto central A é zero ao longo do eixo lateral:

$$\dot{y}_a^r = 0 \quad (5)$$

Usando a matriz de rotação ortogonal $R(\theta)$, a velocidade no referencial inercial dá

$$-\dot{x}_a \sin \theta + \dot{y}_a \cos \theta = 0 \quad (6)$$

² Definem-se como não-holonômicos sistemas com dimensão finita onde algum tipo de restrição é imposta a um ou mais estados do sistema.

- Restrição de rolamento puro:

A restrição de rolamento puro representa o fato de que cada roda mantém um ponto de contato P com o solo conforme mostrado na Figura 4. Não há deslizamento da roda em seu eixo longitudinal (x_r) e nenhuma derrapagem em seu eixo ortogonal (y_r). As velocidades dos pontos de contato na estrutura do robô estão relacionadas às velocidades das rodas por:

$$\begin{cases} v_{pR} = R\dot{\varphi}_R \\ v_{pL} = R\dot{\varphi}_L \end{cases} \quad (7)$$

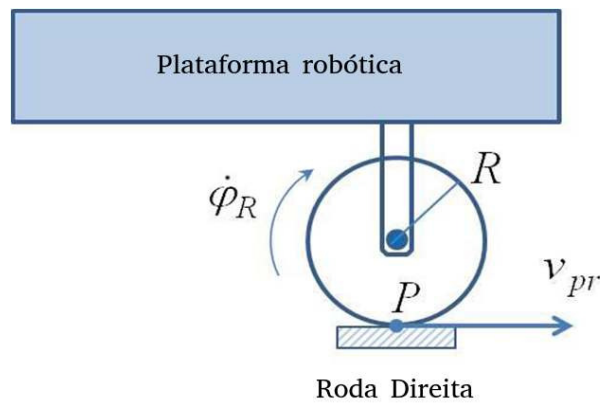


Figura 4 – Restrição de movimento de rolamento puro.

No referencial inercial, essas velocidades podem ser calculadas em função das velocidades do ponto central A do robô:

$$\begin{cases} \dot{x}_{pR} = \dot{x}_a + L\dot{\theta} \cos \theta \\ \dot{y}_{pR} = \dot{y}_a + L\dot{\theta} \sin \theta \end{cases} \quad (8)$$

$$\begin{cases} \dot{x}_{pL} = \dot{x}_a + L\dot{\theta} \cos \theta \\ \dot{y}_{pL} = \dot{y}_a + L\dot{\theta} \sin \theta \end{cases} \quad (9)$$

Usando a matriz de rotação $R(\theta)$, as equações de restrição de rolamento são formuladas da seguinte forma:

$$\begin{aligned} \dot{x}_{pR} \cos \theta + \dot{y}_{pR} \sin \theta &= R\dot{\varphi}_R \\ \dot{x}_{pL} \cos \theta + \dot{y}_{pL} \sin \theta &= R\dot{\varphi}_L \end{aligned} \quad (10)$$

Usando as velocidades dos pontos de contato da equação (x, y) e substituindo em (x, y) , as três equações de restrição podem ser escritas na seguinte forma de matriz:

$$\Lambda(q)\dot{q} = 0 \quad (11)$$

Onde

$$\Lambda(q) = \begin{bmatrix} -\sin \theta & \cos \theta & 0 & 0 & 0 \\ \cos \theta & \sin \theta & L & -R & 0 \\ \cos \theta & \sin \theta & -L & 0 & R \end{bmatrix} \quad (12)$$

e Onde

$$\dot{q} = \begin{bmatrix} \dot{x}_a & \dot{y}_a & \dot{\theta} & \dot{\varphi}_R & \dot{\varphi}_L \end{bmatrix}^T \quad (13)$$

Onde

$$\begin{cases} v_R = R\dot{\varphi}_R \\ v_L = R\dot{\varphi}_L \end{cases} \quad (14)$$

A matriz de restrições acima $\Lambda(q)$ será usada para a modelagem dinâmica do robô móvel de acionamento diferencial.

2.3 Modelo Cinemático

A modelagem cinemática é o estudo do movimento de sistemas mecânicos sem considerar as forças que afetam o movimento. Para o robô móvel de acionamento diferencial, o principal objetivo da modelagem cinemática é representar as velocidades do robô em função das velocidades das rodas motrizes juntamente com os parâmetros geométricos do robô.

A velocidade linear de cada roda motriz na estrutura do robô é, portanto, a velocidade linear do robô móvel de acionamento diferencial na Estrutura do Robô é a média das velocidades lineares das duas rodas

$$v = \frac{v_R + v_L}{2} = R \frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \quad (15)$$

e a velocidade angular do robô móvel de acionamento diferencial é

$$\omega = \frac{v_R - v_L}{2L} = R \frac{(\dot{\varphi}_R - \dot{\varphi}_L)}{2} \quad (16)$$

As velocidades dos robôs móveis de acionamento diferencial na estrutura do robô agora podem ser representadas em termos das velocidades do ponto central A na estrutura do robô da seguinte forma:

$$\begin{cases} \dot{x}_a^r = R \frac{(\dot{\varphi}_R + \dot{\varphi}_L)}{2} \\ \dot{y}_a^r = 0 \\ \dot{\theta} = \omega = R \frac{(\dot{\varphi}_R - \dot{\varphi}_L)}{2L} \end{cases} \quad (17)$$

Por isso

$$\begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ 0 & 0 \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \quad (18)$$

As velocidades do robô móvel de acionamento diferencial podem ser obtidas também no referencial inercial da seguinte forma:

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} \cos \theta & \frac{R}{2} \cos \theta \\ \frac{R}{2} \sin \theta & \frac{R}{2} \sin \theta \\ \frac{R}{2L} & -\frac{R}{2L} \end{bmatrix} \begin{bmatrix} \dot{\varphi}_R \\ \dot{\varphi}_L \end{bmatrix} \quad (19)$$

A equação (19) representa o modelo cinemático direto do robô móvel de acionamento diferencial. Outra forma alternativa para o modelo cinemático pode ser obtida pela representação das velocidades do robô móvel de acionamento diferencial em termos das velocidades lineares e angulares do robô móvel de acionamento diferencial no quadro do Robô.

$$\dot{q}^I = \begin{bmatrix} \dot{x}_a^r \\ \dot{y}_a^r \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (20)$$

2.4 Controladores de Posição e de Orientação

Inicialmente, este TCC tem o objetivo de propiciar a um robô móvel tipo uniclo uma navegação segura em ambientes semi-estruturados, utilizando a abordagem de controle reativo e um sensor de varredura laser. A proposta, portanto, é fazer com que o robô navegue entre dois pontos no ambiente. A navegação, por sua vez, deve ser realizada de forma estável, isto é, o robô deve sair das coordenadas de origem e atingir assintoticamente o alvo desejado, e nele permanecer sem oscilações. A um controlador projetado segundo este objetivo dá-se o nome de Controlador de Posição. Entretanto, em algumas aplicações é desejável que o robô atinja o alvo com uma orientação pré-definida. A estes controladores dá-se o nome de Controlador de Postura, ou ainda Controlador de Posição e Orientação Final.

Durante o projeto deste controlador são levadas em consideração todas as variáveis de estado que definem a postura do robô no ambiente, ou seja, $[x \ y \ z]^T \rightarrow [x_d \ y_d \ z_d]^T$ para $t \rightarrow t_c$, e outro de orientação final, que faz $\psi \rightarrow \psi_d$ para $t \rightarrow \infty$, os quais comutam entre si para cumprir a tarefa proposta. O instante de tempo t_c representa o momento do comutação dos controladores.

Este trabalho adotou a abordagem de controle de posição e de orientação, a qual é apresentado em (TOIBERO et al., 2007).

2.4.1 Controlador de Posição

Na Figura 5 são ilustrados os referenciais que irão descrever o controle de posição. A origem do plano absoluto, ou referencial inercial, é representado por $\langle o \rangle$. O sistema de

coordenadas do robô móvel é dado por $\langle a \rangle$ e o alvo desejado se encontra referenciado por $\langle g \rangle$. Assim, o objetivo do controle de posição é levar o robô móvel, $\langle a \rangle$, inicialmente localizado em $\langle o \rangle$, até $\langle g \rangle$, de forma estável.

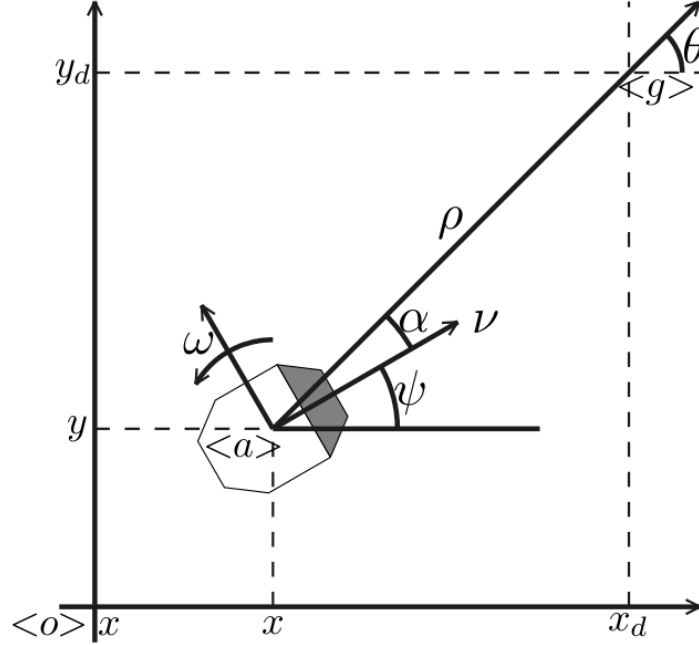


Figura 5 – Sistema de coordenadas para projeto do controlador de posição final.

Visualizando a Figura 5, os erros de posição podem ser definidos como

$$\begin{cases} \tilde{x} = x_d - x \\ \tilde{y} = y_d - y \end{cases} \quad (21)$$

onde x_d e y_d representam as coordenadas do ponto desejado. Logo, levar os erros de posição a zero equivale dizer atingir o alvo desejado.

Ainda visualizando a Figura 5, o erro de posição ρ , a orientação do alvo θ em função da posição corrente do robô e o erro de orientação α do robô em relação ao alvo $\langle g \rangle$ são dados por

$$\begin{cases} \rho = \sqrt{\tilde{x}^2 + \tilde{y}^2} \\ \theta = \arctan \frac{\tilde{y}}{\tilde{x}} \\ \alpha = \theta - \psi \end{cases} \quad (22)$$

Utilizando a Equação (22), o modelo cinemático do robô móvel apresentado na Equação 20 pode agora ser escrito em coordenadas polares, como

$$\begin{cases} \dot{\rho} = -v \cos \alpha \\ \dot{\theta} = v \frac{\sin \alpha}{\rho} \\ \dot{\alpha} = v \frac{\sin \alpha}{\rho} - \omega \end{cases} \quad (23)$$

Desta forma, o controle de posição consiste em levar os erros de posição e de orientação a zero, ou seja, levar $[\rho \ \alpha]^T \rightarrow [0 \ 0]^T$, e, conseqüentemente, obter que $[x \ y]^T \rightarrow [x_d \ y_d]^T$.

O controlador aqui apresentado é projetado com base na Teoria de Lyapunov. Assim, é proposta a função candidata de Lyapunov radialmente ilimitada, dada por

$$V(\rho, \alpha) = \frac{1}{2}\rho^2 + \frac{1}{2}\alpha^2 > 0 \quad (24)$$

Tomando sua derivada temporal, tem-se, após substituir a Equação 23, que

$$\dot{V}(\rho, \alpha) = -\rho v \cos \alpha + \alpha \left(v \frac{\sin \alpha}{\rho} - \omega \right) \quad (25)$$

Adotando os sinais de controle

$$\begin{aligned} v &= v_{max} \tanh \rho \cos \alpha \\ \omega &= k_\omega \alpha + v_{max} \frac{\tanh \rho}{\rho} \sin \alpha \cos \alpha \end{aligned} \quad (26)$$

onde v_{max} e k_ω são constantes reais positivas. Para evitar a saturação dos controladores, os sinais de controle gerados são limitados pela função de saturação tangente hiperbólico ($\tanh(\cdot)$).

Inserindo tais sinais de controle da Equação (26) na Equação (25), obtém-se

$$\dot{V}(\rho, \alpha) = -v_{max}\rho \tanh \rho \cos^2 \alpha - k_\omega \alpha^2 < 0 \quad (27)$$

ou seja, a derivada da função candidata de Lyapunov é definida negativa. Logo $[\rho \ \alpha]^T \in L_\infty^n \cap L_2^n$, e, portanto, garante-se que $[\rho \ \alpha]^T \rightarrow [0 \ 0]^T$ para $t \rightarrow \infty$, assintoticamente. É importante ressaltar que ρ deve ser ligeiramente diferente de zero, a fim de evitar uma singularidade na Equação (23). Assim, considera-se que o robô atingiu a posição desejada quando $\rho < \delta$, sendo $\rho > 0$ arbitrariamente pequeno, o qual irá sanar, portanto, uma possível situação de singularidade.

2.4.2 Controlador de Orientação

Em situações nas quais o robô móvel é dotado de uma garra, este extremo operativo deve estar orientado em relação ao objeto a ser agarrado. Neste contexto, é proposto a seguir um controlador de orientação, o qual deve rotacionar o robô até um ângulo desejado e mantê-lo nesta postura.

Pensando na situação em que o robô já atingiu o alvo desejado, utilizando, por exemplo, o controlador apresentado na subseção anterior, verifica-se que $\alpha \rightarrow 0$ para $t \rightarrow \infty$. Assumindo que o sistema se encontra em estado estacionário, tem-se que $\psi = \theta$.

O objetivo do controlador de orientação é fazer com que $\psi \rightarrow \psi_d$. O erro de orientação é dado por

$$\tilde{\psi} = \psi_d - \psi \quad (28)$$

Do modelo cinemático do robô móvel apresentado na Equação (20), tem-se que

$$\dot{\psi} = \omega \quad (29)$$

Tomando a derivada temporal da Equação (28) e assumindo que ψ_d é uma constante, a Equação (29) é agora dada por

$$\dot{\tilde{\psi}} = -\omega \quad (30)$$

Seja uma função candidata de Lyapunov que garanta a estabilidade assintótica do ponto de equilíbrio $\varphi = 0$ da Equação (30), ou seja,

$$V(\tilde{\psi}) = \frac{1}{2}\tilde{\psi}^2 \quad (31)$$

Tomando sua derivada e utilizando a Equação (30), resulta

$$\dot{V}(\tilde{\psi}) = -\tilde{\psi}\omega \quad (32)$$

Assumindo os sinais de controle dados por

$$\begin{aligned} v &= 0 \\ \omega &= \omega_{max} \tanh \tilde{\psi} \end{aligned} \quad (33)$$

para $\omega_{max} > 0$, obtém-se

$$\dot{V}(\tilde{\psi}) - \omega_{max}\tilde{\psi} \tanh \tilde{\psi} < 0 \quad (34)$$

ou seja, a derivada da função candidata de Lyapunov é definida negativa e, por conseguinte, o equilíbrio $\tilde{\psi} = 0$ é assintoticamente estável. Assim, $\psi \rightarrow \psi_d$ para $t \rightarrow \infty$.

2.4.3 Estabilidade de Sistemas Comutados

Em um sistema de controle de postura utilizando os controladores apresentados nas subseções anteriores, quando o robô atinge o ponto alvo o sistema deve comutar do controlador de posição para o de orientação. Porém, deve-se ter em mente que a estabilidade dos sistemas deve estar assegurada durante o chaveamento de controladores. Para isto, utiliza-se a extensão direta do teorema de estabilidade de Lyapunov, que garante a estabilidade de sistemas comutados desde que os controladores de cada sistema em separado sejam projetados utilizando a mesma função candidata de Lyapunov (TOIBERO et al., 2007).

Quando o robô atinge a posição desejada, um valor de orientação desejada deve ser atingido. Pela Figura 5, isto equivale a dizer que $\theta = \psi_d$. Assim, o objetivo agora é fazer com que $\psi \rightarrow \theta$. Para verificar a estabilidade na comutação, considera-se o desenvolvimento do controlador de posição aplicado ao controle de orientação. Tomando os sinais de controle da Equação (26), porém assumindo $\rho \approx 0$, tais sinais podem ser escritos como

$$\begin{aligned} v &= 0 \\ \omega &= k_\omega \alpha \end{aligned} \tag{35}$$

ou seja, eles são similares à Equação (33). Pela estabilidade assintótica, já demonstrada na Seção anterior, conclui-se, portanto, que $\alpha \in L_\infty^n \cap L_\infty^2$, ou seja, $\alpha \rightarrow 0$ para $t \rightarrow \infty$, e, por sua vez, $\psi \rightarrow \theta = \psi_d$, para $t \rightarrow \infty$, efetuando a re-orientação do robô para estabelecer a postura desejada. Adicionalmente, pode-se dizer que a função candidata de Lyapunov da Equação (31) está contida na da Equação (24), e, portanto afirmar que ambos os controladores foram projetados segundo a mesma função candidata de Lyapunov, o que garante a estabilidade de sistemas comutados.

2.5 Simulador CoppeliaSim

O simulador CoppeliaSim tem uma vasta gama de recursos para serem utilizados nos mais variados tipos de aplicações, esses recursos permitem que possamos realizar testes com uma maior precisão e eficiência dos estudos desenvolvidos. Nem todos os recursos são utilizados neste trabalho, porém o conhecimento é importante para a realização de trabalhos futuros. Alguns dos recursos mais importantes do CoppeliaSim são:

- Suporta API para 6 tipos de linguagens de programação: C, Java, Python, Matlab, Octave e Lua;
- API's remotas para que seja possível controlar os objetos através de programas externos;

- Cinemática direta e inversa para realização dos cálculos automaticamente;
- Sensores de proximidade conseguindo calcular com exatidão os objetos no simulador;

Além dos recursos citados anteriormente, o CoppeliaSim disponibiliza em seu software o robô Pioneer 3-DX como visto na Figura 6, o qual tem todas as características de um robô real, tanto nos sensores, motores entre outros aspectos.



Figura 6 – Pioneer 3-DX no simulador CoppeliaSim.
Fonte: Coppelia Robotics

3 Metodologia

Neste capítulo é apresentado o processo desenvolvido para controle de navegação, mostrando a plataforma de simulação, o caminho e cálculos propostos para atingir o objetivo.

3.1 Interface de Desenvolvimento e Simulação

Como comentado no capítulo anterior, a ferramenta de simulação utilizada é o CoppeliaSim, então nesta sessão será caracterizada sua interface e suas funcionalidades. Na Figura 7 as principais seções do simulador são exibidas, como, barra de menu, barra de ferramentas, tela de modelos, tela de hierarquia e a tela de projeto.



Figura 7 – Interface do simulador CoppeliaSim.

A barra de menu contém a maioria das funcionalidades e ferramentas do simulador, desde a criação de novas cenas e a adição de componentes até o acesso ao manual do usuário.

A barra de ferramentas permite ao usuário realizar mudanças na posição ou na orientação de um objeto, robô ou outro componente quando o mesmo está na tela de projeto.

A barra de ferramentas ainda permite ao usuário inicializar a simulação, pausar, parar e até mesmo acelerar ou reduzir a velocidade de simulação.

A tela de modelos mostra todos os robôs, mesas, pessoas e acessórios que podem ser utilizados para realizar uma simulação. Para utilizar o objeto escolhido, o usuário

precisa apenas segurar o objeto e arrastar para a tela de projeto com o mouse, soltando logo em seguida.

A tela de hierarquia apresenta todos os elementos que compõem uma cena na tela de projeto. Os objetos que são construídos na cena apresentam uma estrutura hierárquica, podendo assim expandir ou contrair a quantidade de itens.

Na tela de projeto, todos os componentes são colocados para gerar a simulação e visualização do que está sendo executado.

Na Figura 8 é apresentado o ícone que dá acesso a interface de programação do CoppeliaSim, o qual é programado em linguagem LUA, entretanto, através dessa interface e também de API's é possível realizar a comunicação com linguagens de programação alternativas, como python, matlab, C++, entre outras.

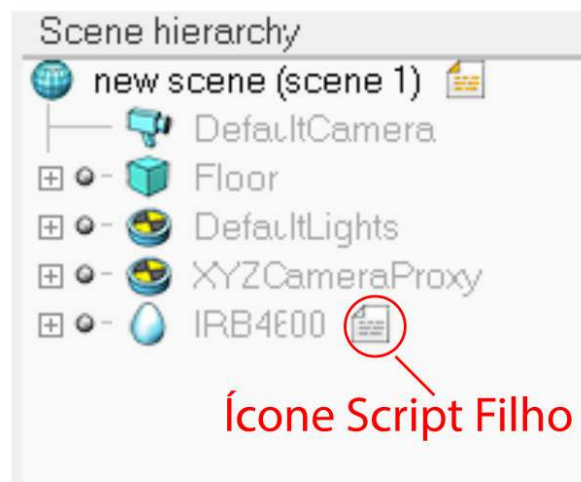


Figura 8 – Ícone de acesso a interface de programação em LUA.

3.2 Programação

No primeiro momento foram criados códigos em Python e em MATLAB que se conectam com o Coppelasim por API(Interface de programação de aplicações), códigos criados com as especificações de controle do projeto. O código caracteriza o modelo de *Model-in-the-Loop* onde o controlador atua e um modelo simulado, conforme ilustrado esquematicamente na Figura 9. Também foi utilizado a configuração de programação em MATLAB para a sintonia dos controladores. O algoritmo de navegação é desenvolvido em uma estrutura baseada em ROS que permite a execução de seus algoritmos de controle.

3.3 Desenvolvimento da Simulação

Ao compreender as ferramentas do CoppeliaSim, agora é possível conhecer mais sobre o desenvolvimento e aplicabilidade empregada nesse trabalho. Com referência à

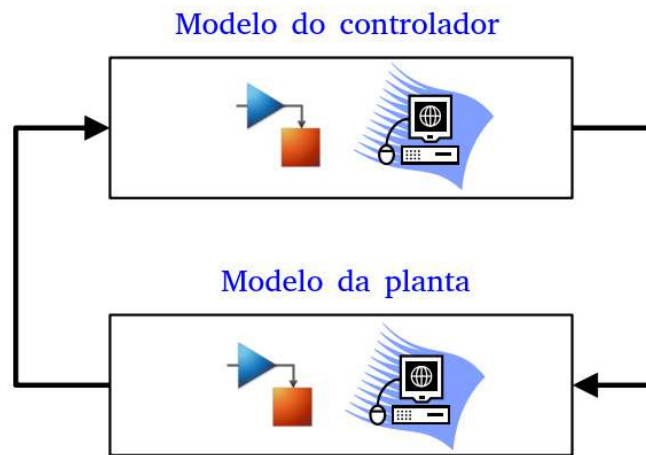
Figura 9 – *Model-in-the-Loop*.

Figura 5 algumas hipóteses podem ser geradas, o estado de navegação pode ser definido como o erro entre a pose inicial $[x, y, \theta]$ e o ponto alvo $[x_d, y_d, \theta_d]$. O controlador de navegação permite movimentos para frente e para trás, com referência ao centro de massa do robô, e o robô pode girar em torno de seu próprio eixo.

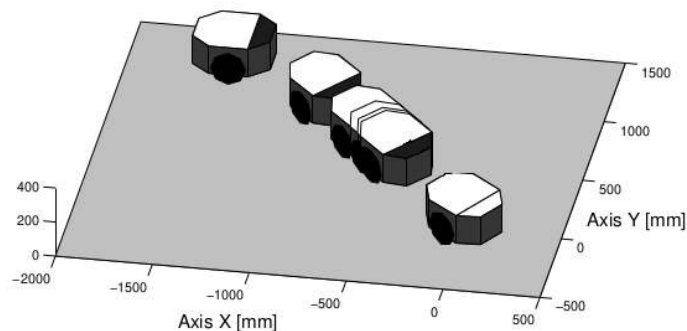


Figura 10 – Caminho percorrido pelo robô móvel.

A simulação utilizando o Pioneer 3DX tem como o objetivo o rastreamento do alvo. A Figura 10 é possível verificar a trajetória do robô para lograr a postura desejada. No experimento, o robô está situado na posição $[-150cm, 120cm]$, com uma orientação de 0° em relação ao eixo $[x, y]$. Por sua vez, o robô deverá ir para o alvo que está localizado nas coordenadas $[0cm, 0cm]$, com uma orientação de -45° . O robô é plotado a cada intervalo de 3 segundos e, é possível verificar que o sistema é estável e o movimento é suave.

O ROS(Robot Operating System) foi implementado no sistema, pois através de uma interface de comunicação em alto nível podemos controlar o robô, comandos de controle para atuação dos motores e resposta dos sensores de odometria do robô administrada pelo ROS. Além disso, o sistema é projetado para que o controle de alto nível do veículo, como planejamento de missão, possa ocorrer inteiramente via ROS, inteiramente por meio da interface de planejamento de missão padrão do veículo ou usando uma mistura dos dois. Isso permite que os controle de baixo nível, como atuação dos motores, sejam abstraídos

pelos recursos implementados no ROS, isso facilita o teste de novas funcionalidades, a mudança do controle de navegação e a implementação em um robô real.

3.4 Nova Abordagem de Controle

Com a utilização da lei de controle prevista, foi notado uma necessidade de modificação na lei para que melhor atendesse a necessidade dos experimentos. Como a lei de controle é diretamente proporcional à distancia ao alvo, quanto mais próximo do alvo o robô se movimenta lentamente. Então foi necessário adicionar um ganho proporcional k_v à variável de controle da velocidade do robô, logo a Equação (26) é substituída por:

$$\begin{aligned}v &= v_{max} \tanh(k_v \rho) \cos \alpha \\ \omega &= k_\omega \alpha + v_{max} \frac{\tanh \rho}{\rho} \sin \alpha \cos \alpha\end{aligned}\tag{36}$$

3.5 Implementação em um Robô Real

A implementação em uma plataforma robótica real foi através de um kit robótico chamado GoPiGo3, que é uma plataforma robótica móvel para o Raspberry Pi desenvolvida pela Dexter Industries e adquirida pela Modular Robotics. Sua escolha foi devido a fácil integração dos sensores de odometria com a Raspberry pi e também pelo fato de ser um sistema embarcado que pode embarcar o ROS.

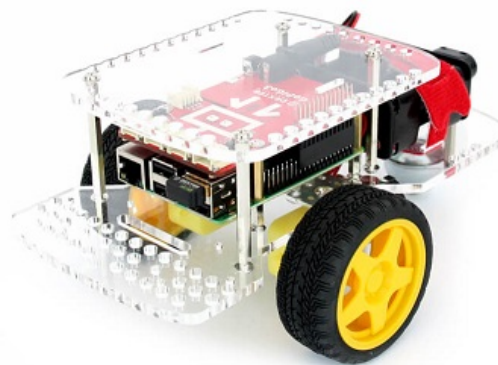


Figura 11 – Plataforma Robótica GoPiGo3.
Fonte: Dexter Industries

Foi instalado o sistema operacional Raspbian, que é uma variante do Debian baseada no ARM, embarcado o ROS Noetic Ninjemys que junto com a plataforma Dexter para controle e aquisição dos valores dos sensores.

4 Resultados e Discussão

Nesta seção são apresentados experimentos e simulações utilizando a proposta de controle de navegação. Um robô Pioneer 3DX (tipo uniciclo) foi utilizado. Foram assumidos os seguintes valores para a constante envolvida no controlador de navegação: $k\omega = 0.5$ e a tolerância de comutação entre os controladores de posição e orientação: 10cm.

4.1 Experimento 1 - Quadrante Positivo

O primeiro experimento tem como objetivo principal destacar a estratégia no quadrante positivo de referência. Neste caso, o robô está localizado na posição $(0m, 0m)$, com orientação igual a 0.5 radianos com respeito ao eixo x. Seu objetivo é chegar à posição $(2m, 3m)$, com orientação também igual a zero com respeito ao eixo x.

A Figura 12 ilustra a trajetória do robô seguidor visto no Rviz e a simulação no Coppeliassim. A Figura 13 ilustra a evolução das variáveis do experimentos e a Figura 14 ilustra a trajetória do robô P3dx durante o experimento.

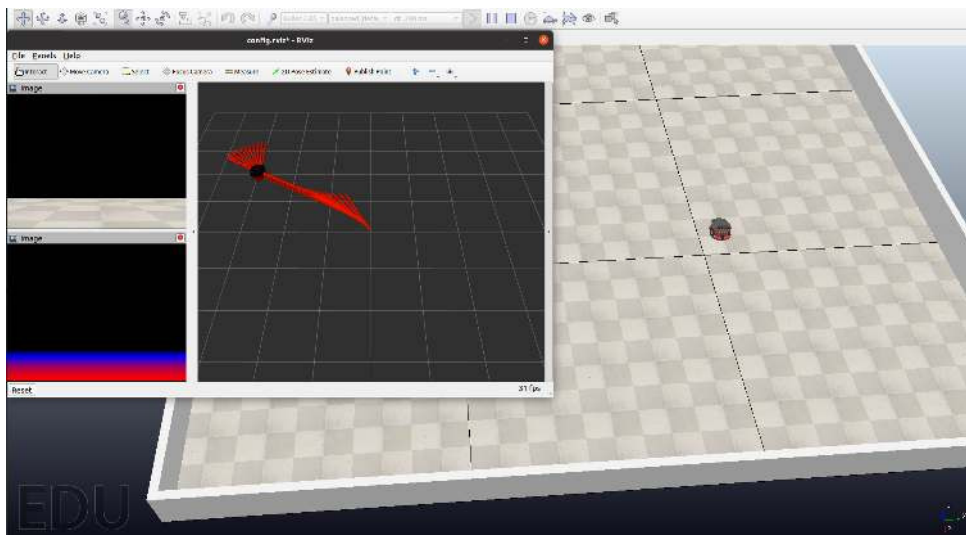


Figura 12 – Coppeliassim com ROS Experimento 1.

O objetivo da missão é realizado, o robô chega ao destino controlado pela lei de controle prevista. É importante mencionar que a missão foi realizada em torno de 60s.

4.2 Experimento 2 - Quadrante Negativo

O segundo experimento tem como objetivo principal destacar a estratégia no quadrante negativo de referência. Neste caso, o robô está localizado na posição $(0m, 0m)$, com

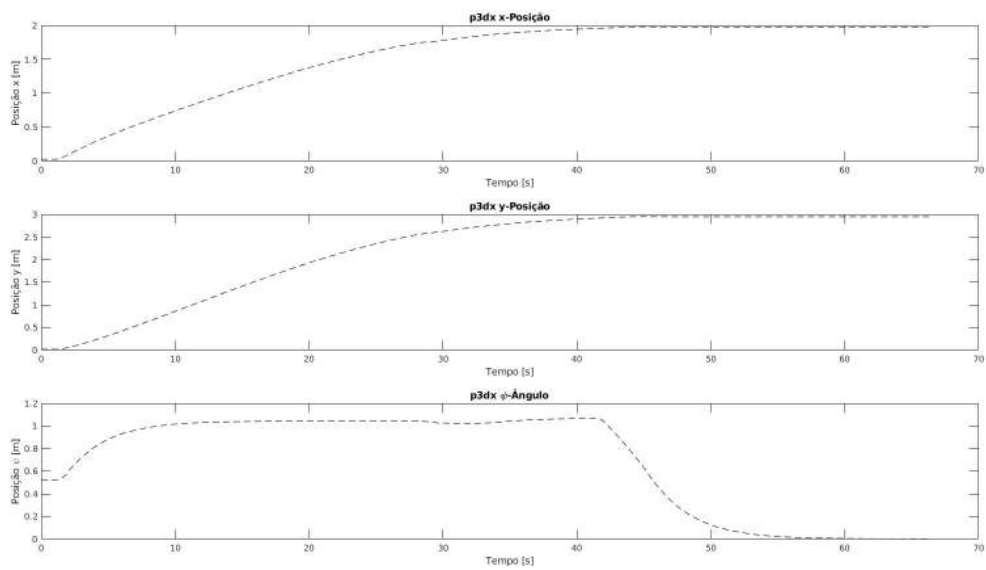


Figura 13 – Métricas do Experimento 1.

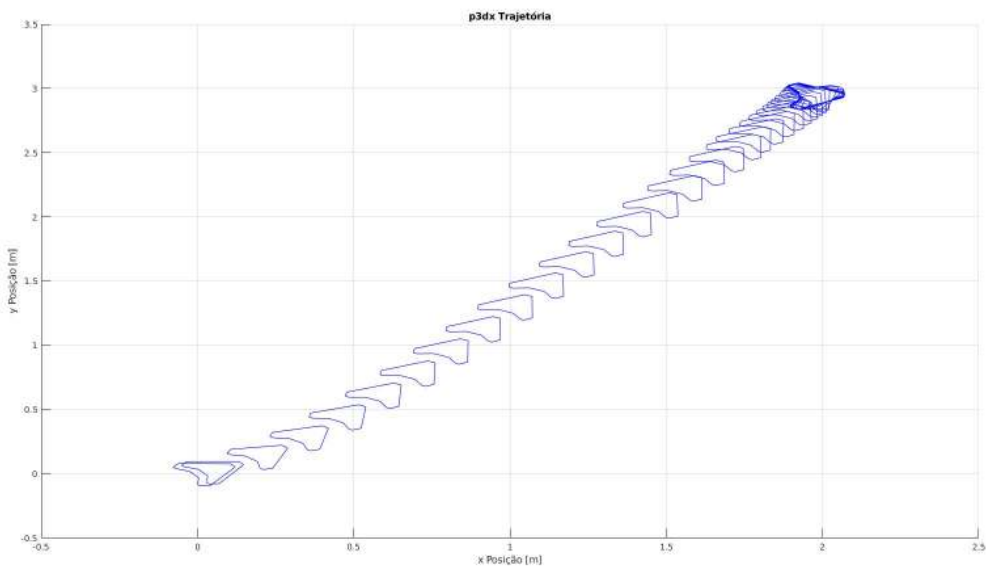


Figura 14 – Trajetória do Experimento 1.

orientação igual a 0.5 radianos com respeito ao eixo x. Seu objetivo é chegar à posição $(-1m, -2m)$, com orientação também igual a zero com respeito ao eixo x.

A Figura 15 ilustra a trajetória do robô seguidor visto no Rviz e a simulação no Coppeliassim. A Figura 16 ilustra a evolução das variáveis do experimentos e a Figura 17 ilustra a trajetória do robô P3dx durante o experimento.

O objetivo da missão é realizado, o robô chega ao destino controlado pela lei de controle prevista. É importante mencionar que a missão foi realizada em torno de 45s.

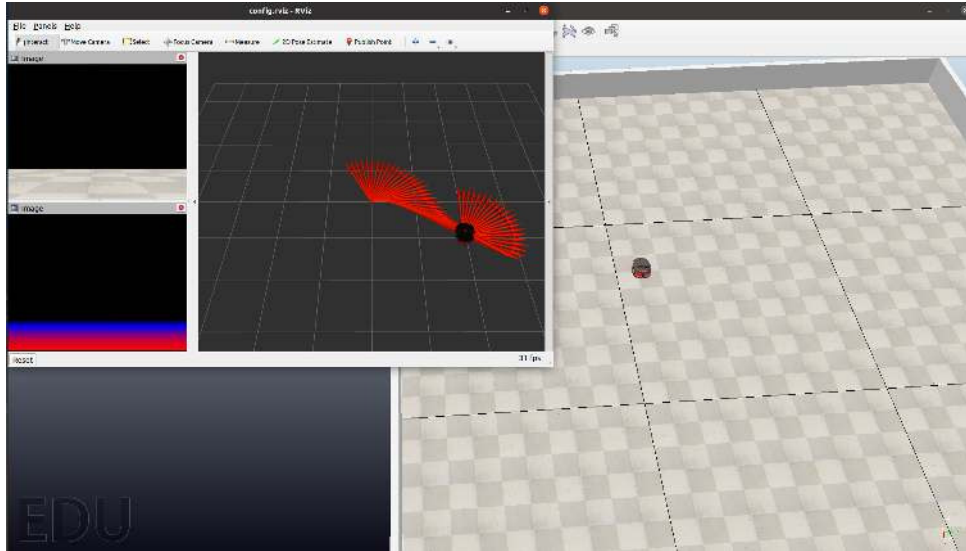


Figura 15 – Coppeliasim com ROS Experimento 2.

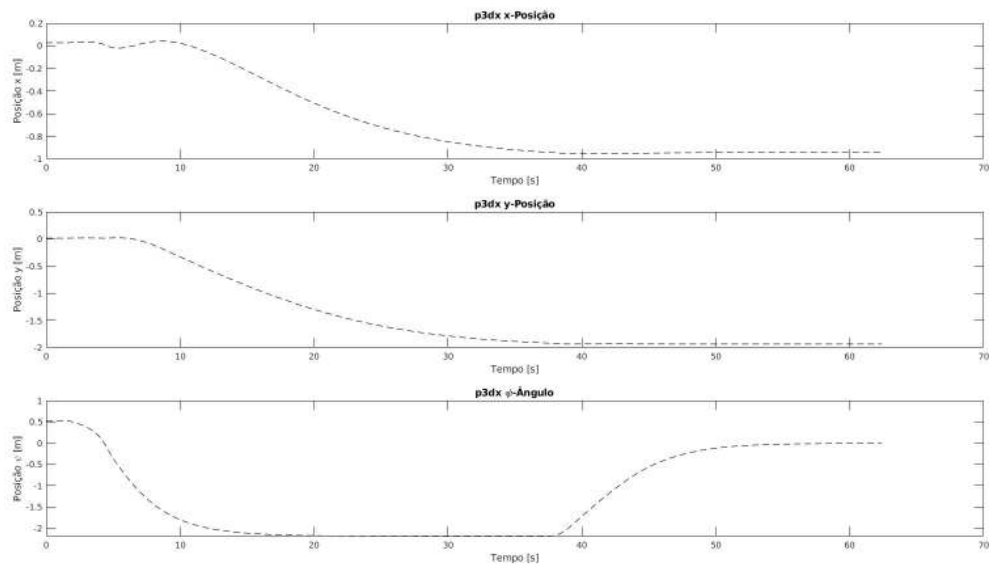


Figura 16 – Métricas do Experimento 2.

4.3 Experimento 3 - Quadrante Misto

O terceiro experimento tem como objetivo principal destacar a estratégia no quadrante misto de referência. Neste caso, o robô está localizado na posição $(0m, 0m)$, com orientação igual a 0.5 radianos com respeito ao eixo x . Seu objetivo é chegar à posição $(-1m, 4m)$, com orientação também igual a zero com respeito ao eixo x .

A Figura 18 ilustra a trajetória do robô seguidor visto no Rviz e a simulação no Coppeliasim. A Figura 19 ilustra a evolução das variáveis do experimentos e a Figura 20 ilustra a trajetória do robô P3dx durante o experimento.

O objetivo da missão é realizado, o robô chega ao destino controlado pela lei de controle prevista. É importante mencionar que a missão foi realizada em torno de 70s.

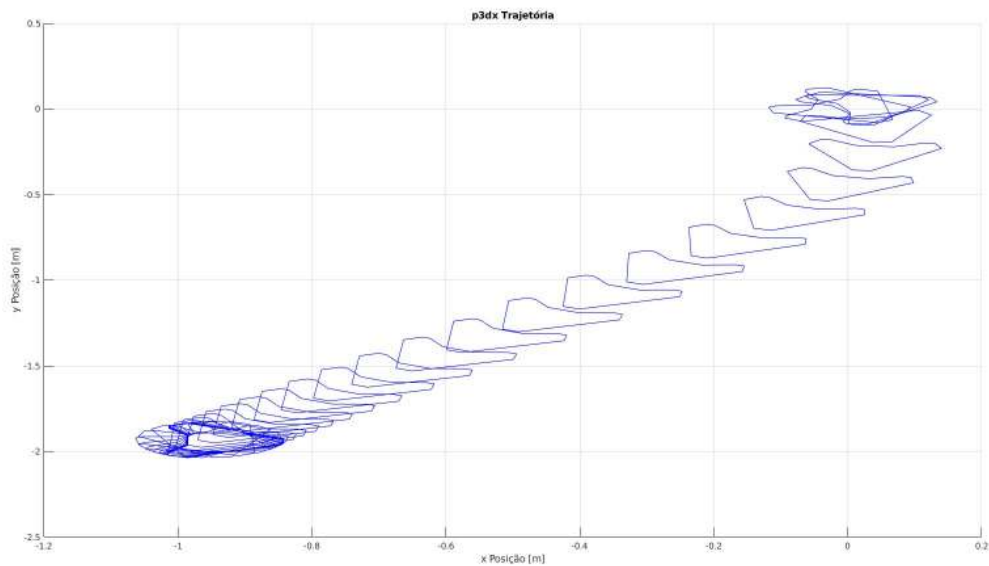


Figura 17 – Trajetória do Experimento 2.

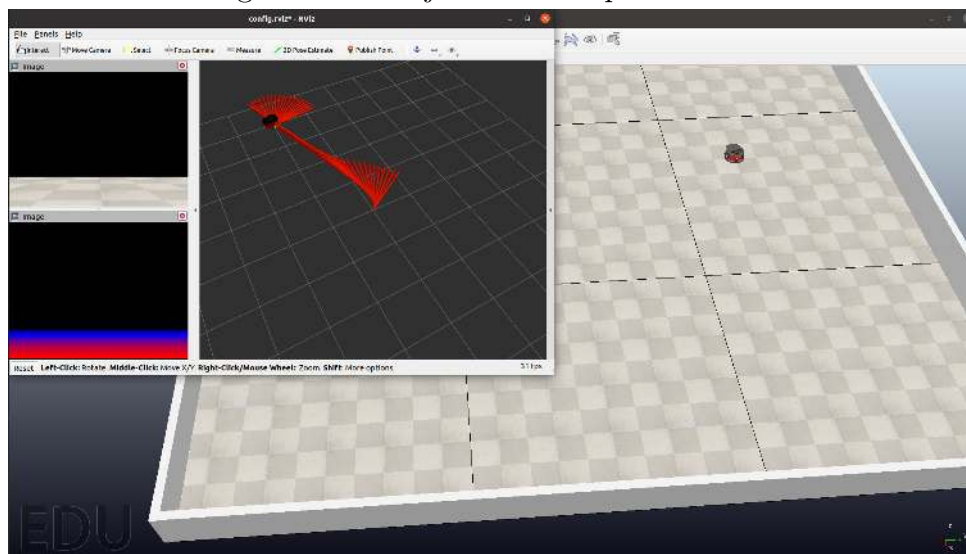


Figura 18 – Coppeliasim com ROS Experimento 3.

4.4 Experimento 4 - Implementação da nova lei de controle

Foi repetido o Experimento 1 onde tem como objetivo principal destacar a estratégia no quadrante positivo de referência. Neste caso, o robô está localizado na posição $(0m, 0m)$, com orientação igual a 0.5 radianos com respeito ao eixo x . Seu objetivo é chegar à posição $(2m, 3m)$, com orientação também igual a zero com respeito ao eixo x . Agora aplicando a nova lei de controle com o intuito de realizar a missão reduzindo o tempo total. No primeiro teste foi aplicado um ganho proporcional $k_v = 2.4$ onde a Figura 21 ilustra a evolução das variáveis do experimentos e a Figura 22 ilustra a trajetória do robô P3dx durante o experimento.

O objetivo da missão é realizado, o robô chega ao destino controlado pela lei de controle prevista. É importante mencionar que agora a mesma missão foi realizada em

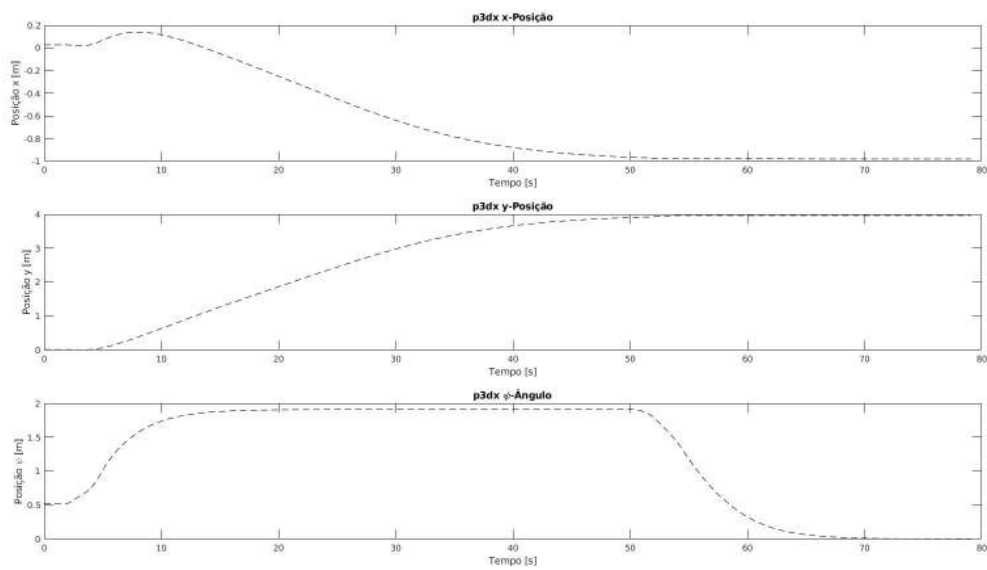


Figura 19 – Métricas do Experimento 3.

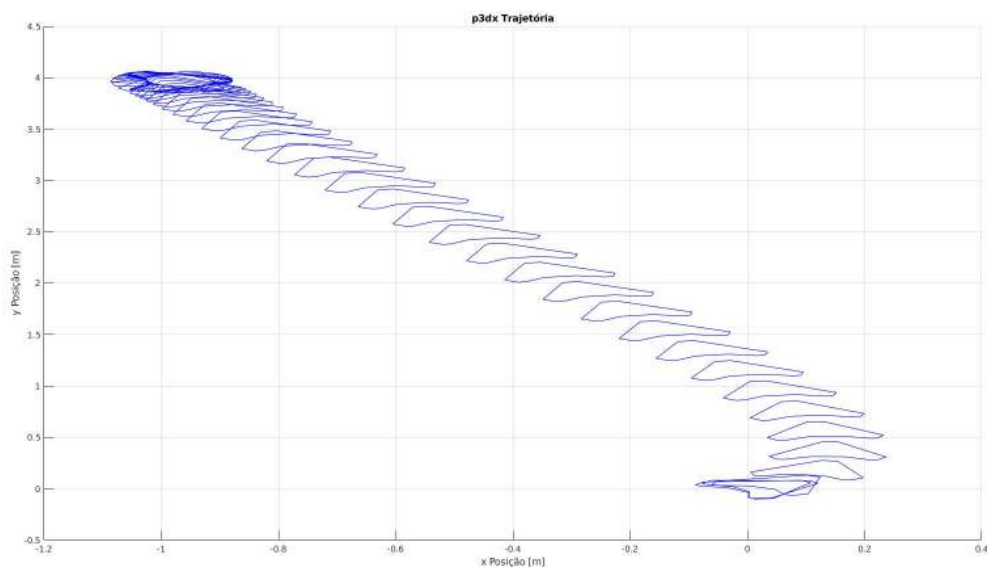


Figura 20 – Trajetória do Experimento 3.

torno de 40s, ou seja, uma redução de 33% do tempo de execução da missão. Após esses testes, foi feita uma sintonia no controlador utilizando o método de ajuste Ziegler-Nichols. O resultado foi um ganho de $k_v = 9.6$, então o Experimento 1 foi repetido onde tem como objetivo principal destacar a estratégia no quadrante positivo de referência. Neste caso, o robô está localizado na posição $(0m, 0m)$, com orientação igual a 0.5 radianos com respeito ao eixo x. Seu objetivo é chegar à posição $(2m, 3m)$, com orientação também igual a zero com respeito ao eixo x. Agora aplicando a nova lei de controle e com o ganho sintonizado. A Figura 23 ilustra a evolução das variáveis do experimentos e a Figura 24 ilustra a trajetória do robô P3dx durante o experimento.

O objetivo da missão é realizado, o robô chega ao destino controlado pela lei de

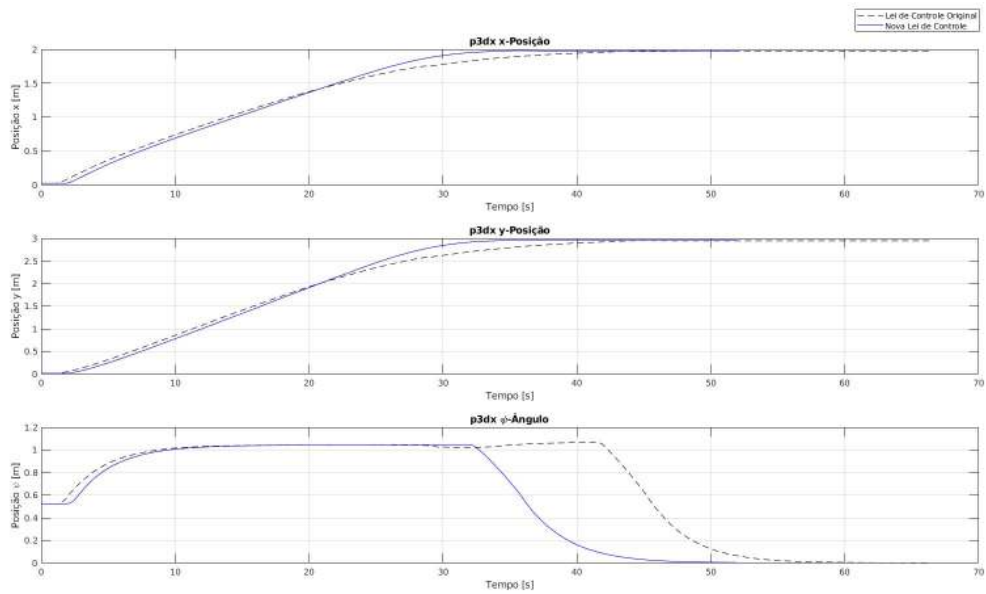


Figura 21 – Métricas do Experimento 4 com $k_v = 2.4$.

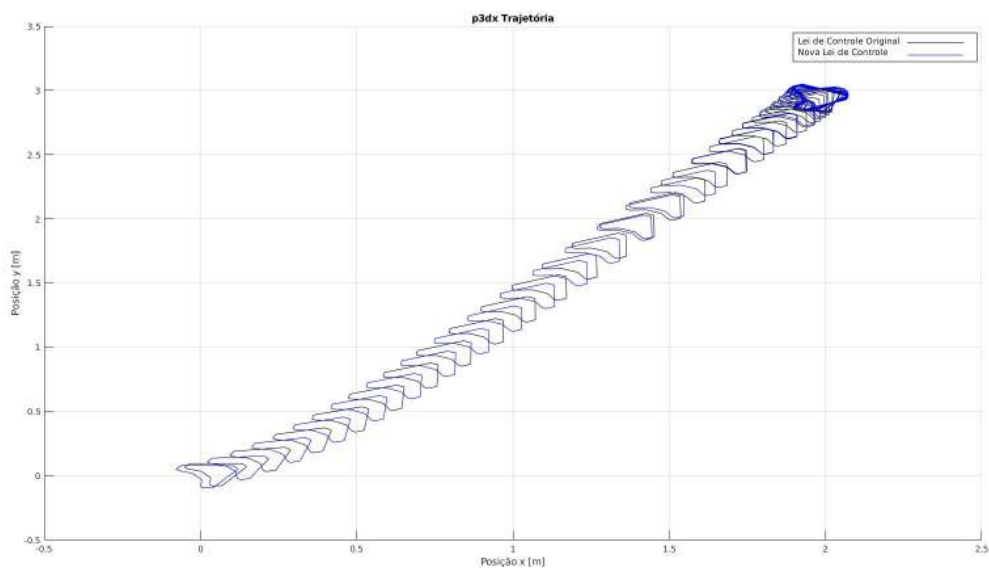


Figura 22 – Trajetória do Experimento 4 com $k_v = 2.4$.

controle prevista. É importante mencionar que agora a mesma missão foi realizada em torno de 35s, ou seja, uma redução de 42% do tempo de execução da missão.

4.5 Experimento 5 - Implementação em um Robô Real

A implementação da estratégia de controle em um robô real depende de algumas etapas, a primeira é definir sistema robótico, onde nele deve ter sensores, atuadores e uma plataforma de programação onde será embarcada o controlador. O robô utilizado é um Kit robótico ilustrado na Figura 25, onde sua estrutura é baseada no robô P3dx, possuindo sensor de distancia e controle de velocidade nos motores para o movimento. A

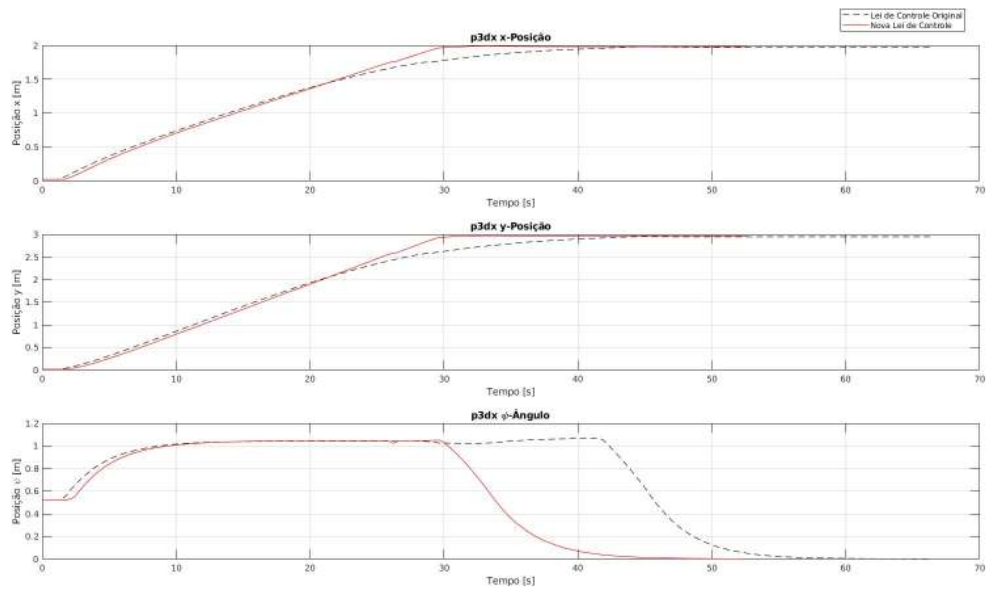


Figura 23 – Métricas do Experimento 4 com $k_v = 9.6$.

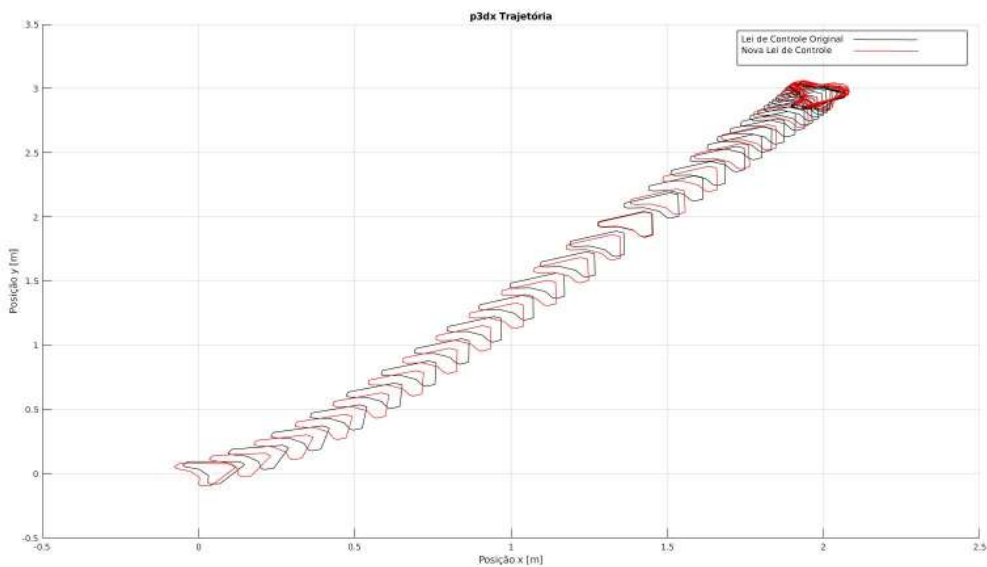


Figura 24 – Trajetória do Experimento 4 com $k_v = 9.6$.

segunda etapa é definir o ambiente de programação pra implementação da lei de controle. O ambiente de programação foi o ROS, onde seus tópicos e publicações permite que o robô navegue de forma reativa no ambiente. A terceira etapa é definir o ambiente de trabalho, onde o robô deve utilizar a lei de controle prevista para se localizar e navegar para a posição assim comandada.

A implementação das duas primeiras etapas do projeto foi possível concluir com os recursos disponíveis, mas a terceira etapa não foi possível implementar pela necessidade de um sensor global. Como a lei de controle depende que o robô se localize no espaço, apenas o sensor de distância embarcado no robô, não foi possível definir sua localização. A utilização do sensor de odometria presente no robô implicaria na necessidade de im-

plementar técnicas de estimação de posição, técnicas que não estão nos objetivos deste trabalho de conclusão de curso. Então foi decidido a conclusão da implementação em um robô real com apenas as duas primeiras etapas.

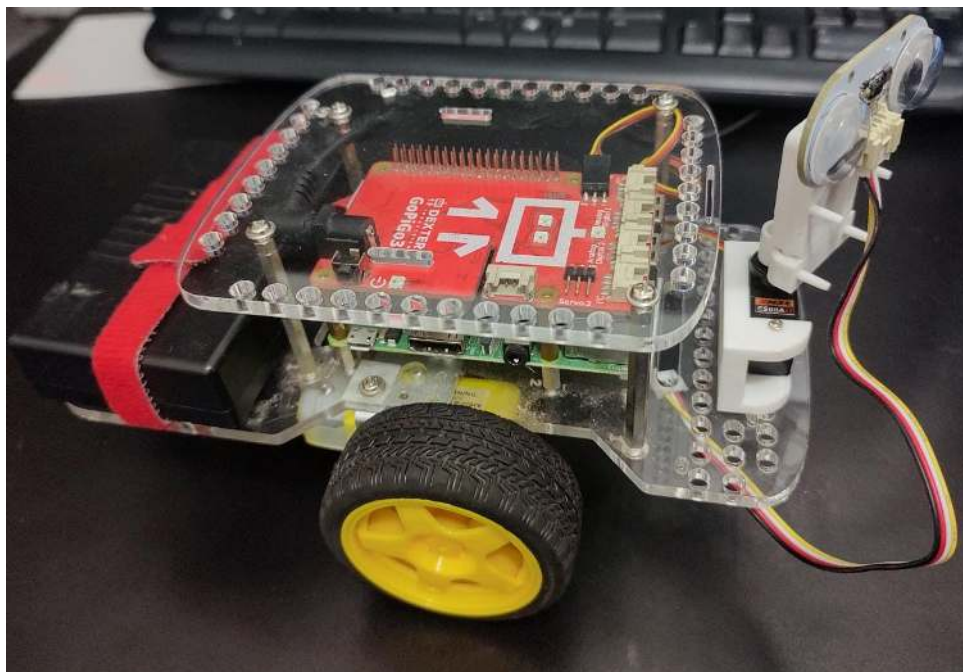


Figura 25 – Robô Real.

5 Conclusão

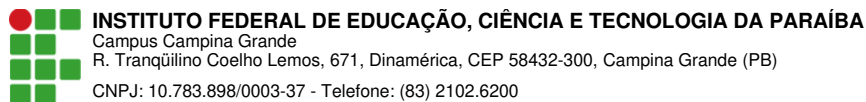
Através do controlador proposto é possível comprovar, experimentalmente, a convergência assintótica das variáveis controladas (ρ, θ) para seus valores desejados, resultando, assim, na navegação desejada. Adicionalmente, é possível constatar que o controle proposto originalmente, necessitaria de melhorias, e essas melhorias reduziram drasticamente o tempo de execução da missão. Mediante a estratégia controle de navegação, constatou-se que é possível realizar a aplicação utilizando diversos métodos de implementação, como o Python, MATLAB e ROS. A nova lei de controle proposta nesse trabalho de conclusão de curso se mostrou bastante eficaz, reduzindo drasticamente o tempo de execução da missão de navegação.

5.1 Trabalhos Futuros

Os controladores projetados neste trabalho somente contemplam a cinemática do robô móvel. A adição do modelo dinâmico de um robô móvel tipo unicycle na lei de controle apresentará uma proposta para a compensação da dinâmica deste robô. Tal compensação pode ser agregada ao controlador aumentando sua robustez. Neste trabalho contempla a execução do controle de navegação com simulações, então a aplicação do controle de navegação em um robô real, vencendo as dificuldades em um ambiente real levantas nesse trabalho seria o primeiro passo para a continuação desta pesquisa.

Referências

- ALATISE, M. B.; HANCKE, G. P. A review on challenges of autonomous mobile robot and sensor fusion methods. **IEEE Access**, v. 8, p. 39830–39846, 2020.
- JAIEM, L.; CRESTANI, D.; LAPIERRE, L.; DRUON, S. Energy consumption of control schemes for the pioneer 3dx mobile robot: Models and evaluation. **Journal of Intelligent & Robotic Systems**, Springer, v. 102, n. 1, p. 1–15, 2021.
- MOHANAN, M.; SALGOANKAR, A. A survey of robotic motion planning in dynamic environments. **Robotics and Autonomous Systems**, Elsevier, v. 100, p. 171–185, 2018.
- TOIBERO, J. M.; CARELLI, R.; KUCHEN, B. Switching control of mobile robots for autonomous navigation in unknown environments. In: IEEE. **Proceedings 2007 IEEE International Conference on Robotics and Automation**. [S.l.], 2007. p. 1974–1979.
- VILLANI, V.; PINI, F.; LEALI, F.; SECCHI, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. **Mechatronics**, Elsevier, v. 55, p. 248–266, 2018.
- ZACHARAKI, A.; KOSTAVELIS, I.; GASTERATOS, A.; DOKAS, I. Safety bounds in human robot interaction: A survey. **Safety science**, Elsevier, v. 127, p. 104667, 2020.



Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto: Versão Final do TCC
Assinado por: Davi Juvencio
Tipo do Documento: Anexo
Situação: Finalizado
Nível de Acesso: Ostensivo (Público)
Tipo do Conferência: Cópia Simples

Documento assinado eletronicamente por:

- **Davi Juvêncio Gomes de Sousa, ALUNO (201911250046) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE**, em 03/01/2023 10:27:35.

Este documento foi armazenado no SUAP em 03/01/2023. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 711745
Código de Autenticação: 3ee9ef7433

