



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**CENTRO DE TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE**  
**COMPUTAÇÃO**  
**CURSO DE MESTRADO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**  
**PPGEEC2324 - SISTEMAS ROBÓTICOS AUTÔNOMOS**

**PROJETO 1**

GUILHERME FÉLIX DE MEDEIROS BRANDT - Nº 20241009450  
JOSE LINDENBERG DE ANDRADE - Nº 20241009460  
LARISSA SOARES DE SOUZA - Nº 20241028376  
PIETRO AUGUSTO DE ALBUQUERQUE LIRA E SILVA - Nº 20241028474  
RAFAEL AUGUSTO DE OLIVEIRA GUEDES - Nº 20241009530

Natal-RN  
2024

GUILHERME FÉLIX DE MEDEIROS BRANDT - Nº 20241009450  
JOSE LINDENBERG DE ANDRADE - Nº 20241009460  
LARISSA SOARES DE SOUZA - Nº 20241028376  
PIETRO AUGUSTO DE ALBUQUERQUE LIRA E SILVA - Nº 20241028474  
RAFAEL AUGUSTO DE OLIVEIRA GUEDES - Nº 20241009530

## SEMINÁRIO SOBRE O COPPELIASIM

Relatório apresentado à disciplina de Sistemas Robóticos Autônomos, correspondente à avaliação do 1º projeto do semestre 2024.2 do curso de Mestrado em Engenharia Elétrica e de Computação da Universidade Federal do Rio Grande do Norte, sob orientação do **Profº Drº Pablo Javier**.

Professor: Drº Pablo Javier Alsina.

Natal-RN  
2024

## RESUMO

Esse relatório tem como objetivo simular um robô móvel com acionamento diferencial de forma que seja possível visualizar a movimentação dele através do espaço de trabalho que ele tem para movimentação e através de gráficos gerados. A simulação é realizada no *software* CoppeliaSim e o robô utilizado para a simulação é o pioneer P3Dx.

**Palavras-chave:** Robô móvel; CoppeliaSim; Pioneer P3Dx.

## Lista de Figuras

1	CoppeliaSim . . . . .	6
2	Caixa Model Browser e robô Pioneer P3Dx . . . . .	7
3	Controle de simulação. . . . .	7
4	Gráfico 1 - Velocidades das rodas . . . . .	8
5	Gráfico 2 - Configuração do robô ( $x$ , $y$ , $\theta$ ) . . . . .	8
6	Gráfico 3 - Posição ( $y(t)$ <i>versus</i> $x(t)$ ) . . . . .	9

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>6</b>
1.1	O que é CoppeliaSim? . . . . .	6
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>6</b>
2.1	Como fazer uma simples simulação . . . . .	6
2.2	Apresentar resultados por gráficos . . . . .	7
2.2.1	Gráfico de velocidades das rodas (entradas) em função do tempo . . . . .	7
2.2.2	Gráfico de configuração do robô ( $x$ , $y$ , $\theta$ ) (saídas) em função do tempo . . . .	8
2.2.3	Gráfico das posições ( $x(t)$ , $y(t)$ ) seguidas pelo robô no plano $xy$ . . . . .	9
<b>3</b>	<b>CONCLUSÃO</b>	<b>9</b>

# 1 INTRODUÇÃO

A finalidade do presente trabalho é simular no *software* CoppeliaSim um robô móvel com acionamento diferencial, de maneira a que o mesmo receba os comandos das velocidades de referências para as rodas e retorne a posição e orientação do robô  $(x, y, \theta)$  em um referencial global. E além do movimento do robô no espaço de trabalho, mostrar os gráficos de velocidades das rodas (entradas) em função do tempo; configuração do robô  $(x, y, \theta)$  (saídas), em função do tempo; e o gráfico das posições  $(x(t), y(t))$  seguidas pelo robô no plano  $xy$ .

## 1.1 O que é CoppeliaSim?

O simulador de robô CoppeliaSim, com ambiente de desenvolvimento integrado, é baseado em uma arquitetura de controle distribuída, assim, cada objeto/modelo pode ser controlado individualmente por meio de um *script* embutido (com as linguagens LUA ou Python), um *plugin* (em C, C++), um nó ROS ou BlueZero, um cliente API (*Application Programming Interface*) remoto ou uma solução customizada. Isso torna o CoppeliaSim uma ferramenta bastante versátil e ideal para aplicações multi-robôs. Os controladores podem ser escritos em C / C ++, Python, Java, Lua, Matlab ou Octave.

O CoppeliaSim é usado para desenvolvimento rápido de algoritmos, simulações de automação de fábrica, prototipagem e verificação rápida, educação relacionada à robótica, monitoramento remoto, verificação dupla de segurança, como gêmeo digital entre outras atribuições (Coppelia, 2024).

Figura 1: CoppeliaSim



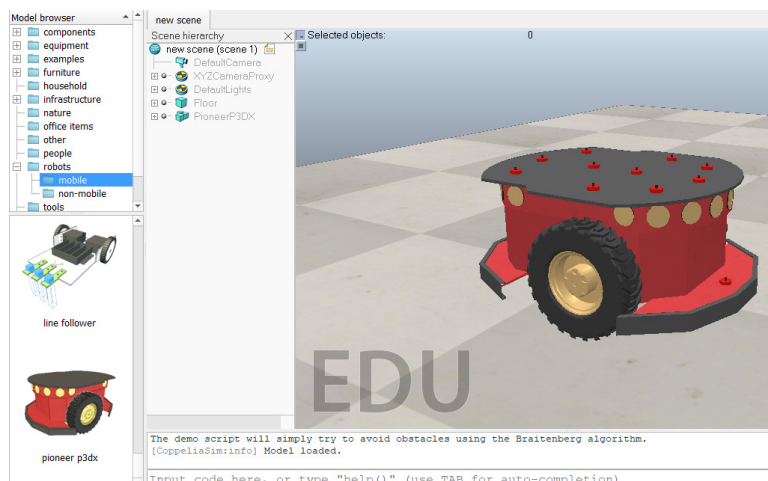
Fonte: Coppelia Robotics

## 2 DESENVOLVIMENTO

### 2.1 Como fazer uma simples simulação

O primeiro passo para realizar uma simulação é escolher o robô. Para isso é necessário observar uma caixa aberta no lado esquerdo do Coppelia onde tem o nome "Model Browser". Nesta existem várias pastas e é preciso clicar na que tem escrito "Robots" e, em seguida, no "Mobile". Após a realização desses passos, logo abaixo dessa caixa aparecerão vários tipos de robôs, mas o que é utilizado nesse trabalho é o Pioneer 3-DX. Tudo isso é possível ser visto na Figura abaixo.

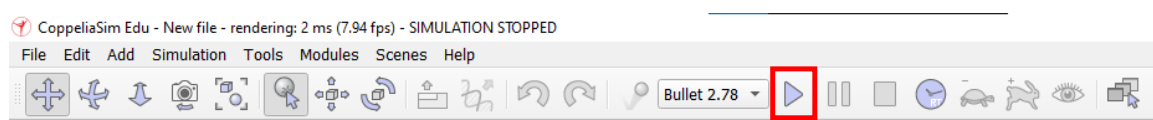
Figura 2: Caixa Model Browser e robô Pioneer P3Dx



Fonte: Autoria Própria.

Com tudo já pronto, para rodar a simulação é necessário apenas apertar a seta, mostrada na Figura 3, que fica na parte central do topo da página do simulador.

Figura 3: Controle de simulação.



Fonte: Autoria Própria.

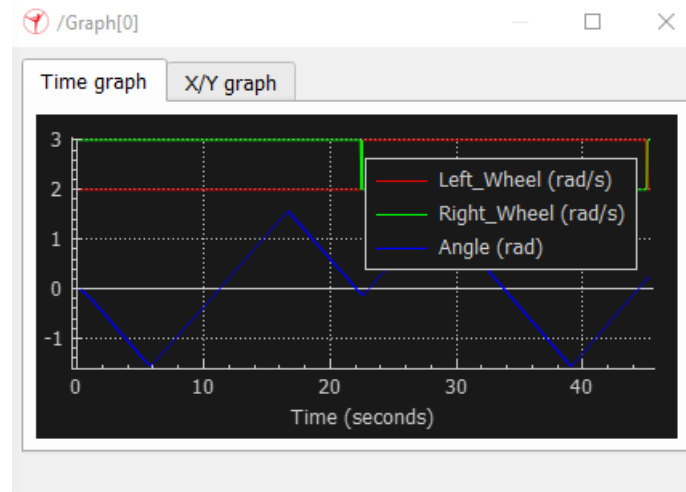
## 2.2 Apresentar resultados por gráficos

Nessa seção está apresentado os resultados gráficos das curvas do robô e os trechos do código responsável pela plotagem dos gráficos.

### 2.2.1 Gráfico de velocidades das rodas (entradas) em função do tempo

O primeiro gráfico (Figura 4) ilustra as velocidades das rodas do robô em função do tempo, apresentando como cada roda (direita e esquerda) se comporta individualmente, assim como o ângulo. O gráfico apresenta o comportamento do robô ao fazer curvas e mudanças de direção. A roda esquerda (linha vermelha) e a roda direita (linha verde) mantêm suas velocidades constantes na maior parte do tempo. No entanto, há momentos em que ocorre uma variação na velocidade de uma ou ambas as rodas, o que está associado às mudanças de direção do robô. Essas alterações são refletidas no ângulo de orientação do robô, representado pela linha azul, que inicialmente diminui (indicando uma curva ou desvio à esquerda), para em seguida voltar a aumentar (indicando uma correção ou curva para a direita). Esse padrão de variação nas velocidades das rodas e no ângulo demonstra o comportamento do robô ao realizar manobras, como curvas e mudanças de direção.

Figura 4: Gráfico 1 - Velocidades das rodas



Fonte: Elaborado pelos autores

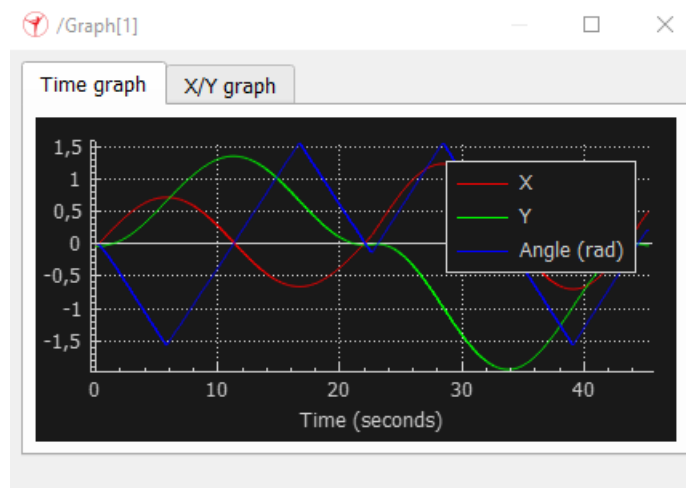
O código utilizado para gerar os dados da Figura 4 está representado abaixo.

```
graph = sim.getObject("/Graph")
chassis_graph = sim.addGraphStream(graph, 'Left Wheel', 'rad/s', 0,{1, 0, 0})
chassis_graph2 = sim.addGraphStream(graph, 'Right Wheel', 'rad/s', 0,{0, 1, 0})
chassis_graph3 = sim.addGraphStream(graph, 'Angle', 'rad', 0,{0, 0, 1})
```

### 2.2.2 Gráfico de configuração do robô ( $x$ , $y$ , $\theta$ ) (saídas) em função do tempo

O segundo gráfico (Figura 5) foi gerado a partir da coleta de dados da posição  $x$ ,  $y$  e  $\theta$  do robô em função do tempo. A variação conjunta dos valores de  $x$  e  $y$  indica que o robô está realizando um movimento curvo. A mudança do ângulo  $\theta$  reflete as alterações na orientação do robô levando a uma alteração na sua trajetória. As oscilações no gráfico sugerem que essas variações no ângulo estão associadas às mudanças de direção ou sentido ao longo do percurso.

Figura 5: Gráfico 2 - Configuração do robô ( $x$ ,  $y$ ,  $\theta$ )



Fonte: Elaborado pelos autores



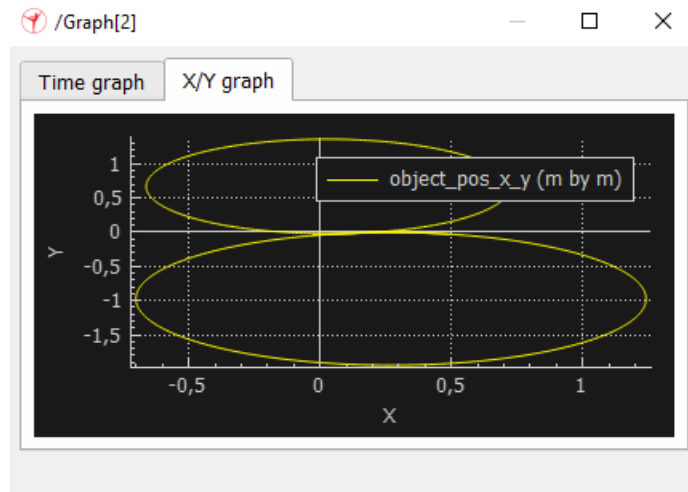
O código utilizado para gerar os dados da Figura 5 está representado abaixo.

```
graphB = sim.getObject("/Graph[1]")
chassis_graphB1 = sim.addGraphStream(graphB, 'X', '', 0, {1, 0, 0})
chassis_graphB2 = sim.addGraphStream(graphB, 'Y', '', 0, {0, 1, 0})
chassis_graphB3 = sim.addGraphStream(graphB, 'Angle', 'rad', 0, {0, 0, 1})
```

### 2.2.3 Gráfico das posições $(x(t), y(t))$ seguidas pelo robô no plano xy

Por fim, no terceiro gráfico, (Figura 6) houve uma modificação em relação aos anteriores, onde, em vez da plotagem no tempo, foi aplicada uma relação de dependência entre os eixos y e x. A linha amarela do gráfico apresenta o desenho da trajetória do robô no plano xy, Isso indica que o robô está realizando um movimento cíclico, em que o robô está repetindo o mesmo padrão de movimento ao longo do tempo, retornando a pontos próximo de sua posição anterior, mas com pequenas variações.

Figura 6: Gráfico 3 - Posição  $(y(t) \text{ versus } x(t))$



Fonte: Elaborado pelos autores

O código utilizado para gerar os dados da Figura 6 está representado abaixo.

```
graphXY = sim.getObject('/Graph[2]')
objectPosX = sim.addGraphStream(graphXY, 'object pos x', 'm', 1)
objectPosY = sim.addGraphStream(graphXY, 'object pos y', 'm', 1)
sim.addGraphCurve(graphXY, 'object pos x/y', 2, {objectPosX, objectPosY}, {0, 0}, 'm by m')
```

## 3 CONCLUSÃO

Realizamos a simulação através do software CoppeliaSim Edu, utilizando em conjunto o modelo de robô mobile Pioneer 3Dx, que permitiu a visualização prática dos conceitos teóricos relacionados ao controle de robôs móveis, no qual podemos observar o comportamento do robô em termos de suas velocidades individuais das rodas, além de sua configuração de posição e orientação no plano, nas

quais são gerados gráficos durante as simulações de forma eficaz com intuito de ilustrar a relação das entradas de controle e as trajetórias seguidas pelo robô.

Utilizamos a linguagem LUA e APIs que nos ajudaram com a integração e controle do robô em um ambiente simulado e minimamente controlado, demonstrando a importância da simulação na fase inicial do projeto com robôs autônomos e resultados que forneceram aprimoramentos no controle e autonomia de robôs móveis.

## Referências

Simulador CoppeliaSim. Coppelia Robotics, 2024. Disponível em: <<https://www.coppeliarobotics.com/>>. Acesso em 17 de Out. de 2024.

CoppeliaSim User Manual, 2022. Disponível em: <<https://manual.coppeliarobotics.com/>>. Acesso em 17 de Out. de 2024.