Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação

EEC1507 - Sistemas Robóticos Autônomos

RELATÓRIO

1º PROJETO DE SISTEMAS ROBÓTICOS AUTÔNOMOS - META 1

Alunos:
Edel Mary Quinn de Oliveira Figueiredo
Luís Gabriel Pereira Condados
Samigo Ricardo de Oliveira Silva
Professor orientador: Pablo Javier Alsina

Natal-RN
2020

Universidade Federal do Rio Grande do Norte

Programa de Pós-graduação em Engenharia Elétrica e de Computação

EEC1507 - Sistemas Robóticos Autônomos

RELATÓRIO

Relatório apresentado à disciplina de EEC1507- Sistemas Robóticos Autônomos, correspondente 1º unidade do semestre 2020.2, sob orientação do **Prof. Pablo Javier Alsina**.

Alunos:
Edel Mary Quinn de Oliveira Figueiredo
Luís Gabriel Pereira Condados
Samigo Ricardo de Oliveira Silva

Natal-RN
2020

Conteúdo

1 INTRODUÇÃO

O uso de robôs móveis está presente em qualquer área de atuação, devido a sua boa manobrabilidade, sendo a direção diferencial o movimento mais comum. Contudo o aprendizado de seu uso também pode trazer um gasto temporal devido a busca por respostas ideais, dependendo do modelo, diâmetro das rodas, quantidade de rodas entre outros.

Para chegar a respostas mais rápidas e mais próximas das ideais podemos usar plataformas de simulações robóticas disponíveis no mercado, como o V-REP, Open HRP, Gazebo, entre outras. Contudo a plataforma Virtual de Experimentação Robótica (Virtual Robot Experimentation Platform), é tratada apenas como V-REP na literatura, destacando-se por se tratar de uma plataforma fácil e didática, traindo desde leigos a estudantes, ou até profissionais na área da robótica. Esta plataforma além do uso de linguagem de programação interna através do Lua também disponibiliza o uso de algumas linguagens de forma remota como a própria Lua, além do Python, C++, Java e Matlab. Sendo escolhido para a realização deste relatório a linguagem de programação Python, pelo fato da mesma ser a linguagem com mais similaridade entre os membros.

Este trabalho tem como objetivo simular um robô móvel com acionamento diferencial, desenvolvendo um sistema de controle cinemático (comando de velocidades e recepção de posição) que permita ao mesmo executar movimentos especificados em espaço livre de obstáculos. A simulação será realizada no ambiente CoppeliaSim em configuração cliente/servidor e através de gráficos serão analisados a orientação e posição do robô ao longo do tempo, bem como suas entradas.

2 REFERENCIAL TEÓRICO

Para a construção da modelagem do robô com rodas de dois graus de liberdade, conhecida também na literatura como roda padrão, temos que levar em consideração a restrição de rolamento puro, onde todo o movimento da roda tem que ser acompanhado pela rotação correspondente da roda, enquanto que para a restrição de derrapagem lateral todo o movimento deve ser restrito ao plano da roda, ou seja, a roda não pode ser movimentada em direção ao eixo. Com a escolha do robô móvel com acionamento diferencial, o uso da relação entre a velocidade das rodas pode ser obtido pelo giro do robô com o raio r também pode ser obtido por análise simples a partir da figura 1:

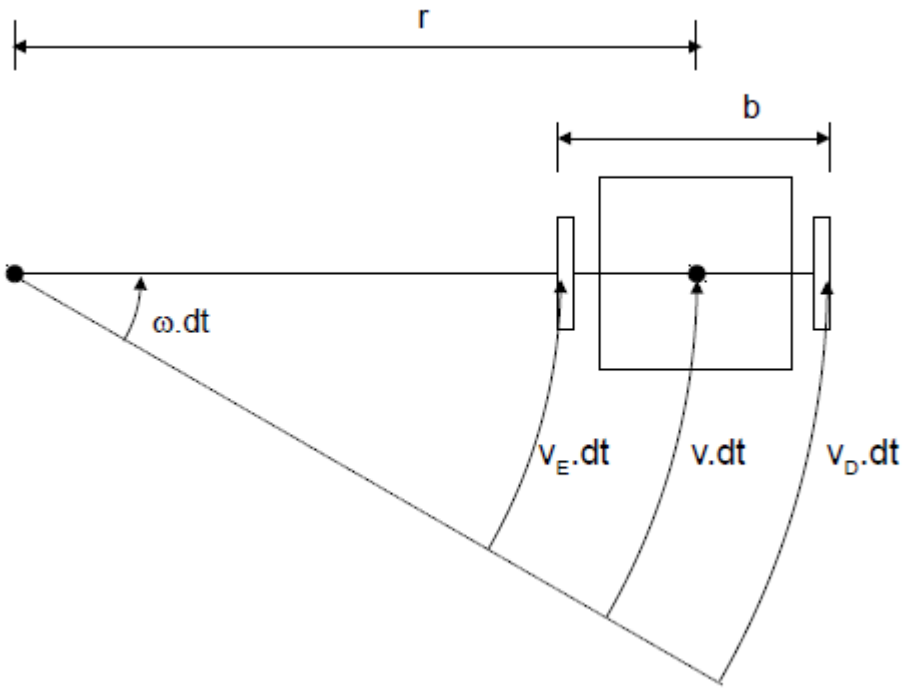

Figura 1: Movimentos Infinitesimais para o modelo de robô utilizado.

Analisando a natureza do movimento circular na figura 1 podemos observar as seguintes relações,

$$\omega \left(r - \frac{b}{2} \right) = v_e \tag{1}$$

$$\omega \left(r + \frac{b}{2} \right) = v_d \tag{2}$$

Somando v_e e v_d temos:

$$v = \frac{(\omega_d + \omega_e).r_w}{2} \tag{3}$$

E subtraindo v_e e v_d podemos chegar em:

$$\omega = \frac{(\omega_d - \omega_e).r_w}{b} \tag{4}$$

Podemos também relacionar as velocidades das rodas com o raio de giro instantâneo do robô

$$\frac{\omega_e}{\omega_d} = \frac{r - \frac{b}{2}}{r + \frac{b}{2}} \tag{5}$$

Por fim temos a **função de cinemática direta do robô**, considerando o espaço de configuração: $\begin{bmatrix} x & y & \theta \end{bmatrix}^T$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{(\omega_d+\omega_e).r_w}{2}.\cos\theta \\ \frac{(\omega_d+\omega_e).r_w}{2}.\sin\theta \\ \frac{(\omega_d-\omega_e).r_w}{b} \end{bmatrix} \tag{6}$$

3 METODOLOGIA

A simulação foi realizada no ambiente *CoppeliaSim*[2] em configuração cliente/servidor, sendo o servidor responsável pela simulação da física em tempo real e o cliente em *Python*[1] responsável por transmitir as velocidades que serão aplicadas em cada motor, a comunicação foi feita utilizando a biblioteca para Python disponibilizada pelo *CoppeliaSim* (Legacy remote API [2]), gráficos (gerados com o uso da biblioteca *matplotlib*[3]) também serão gerados e exibidos pelo cliente, em tempo de execução, mostrando a posição e orientação do robô em coordenadas globais em cada instante de tempo, um link para o código fonte bem como os demais recursos utilizados encontram-se em anexo 1.

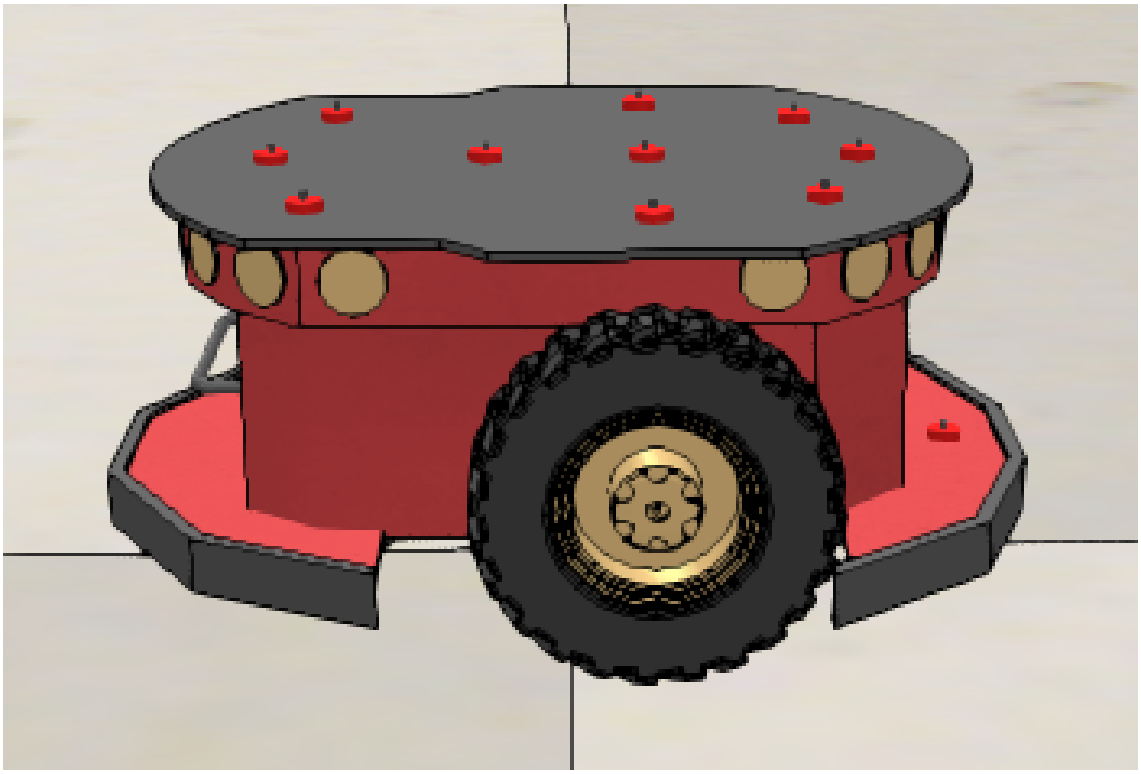

Figura 2: Modelo simulado do robô Pioneer.

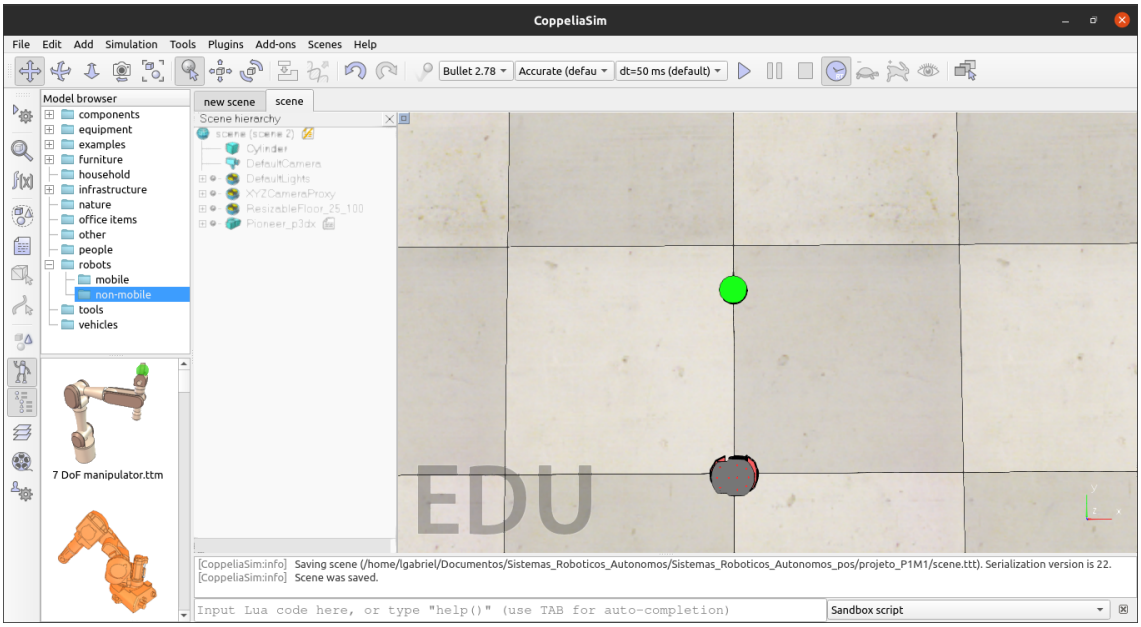

Figura 3: Cenário no *CoppeliaSim* utilizado no experimento.

No ambiente simulado foi criado um cenário simples, contendo apenas um robô Pioneer e uma marcação, como podemos ver nas Figuras 2 e 3. O cliente foi programado para enviar as velocidades para cada motor de forma que o robô realize um movimento circular completo, com raio $2m$ ($r = 2m$), em $30s$, para isso foi utilizado as equações de cinemática do robô apresentadas em (1)(2)(5), simplificando e colocando em função da velocidade ω desejada, temos:

$$\omega = \frac{2.\pi}{\Delta t}$$
$$\omega_d = \omega . \frac{r + b/2}{r_w}$$
$$\omega_e = \omega_d . \frac{r - b/2}{r + b/2}$$

O robô utilizado possui rodas com rádio igual a $0.09751m$ (r_w) e a distância entre os centros das mesmas é de $0.331m(b)$.

4 RESULTADOS E CONCLUSÕES

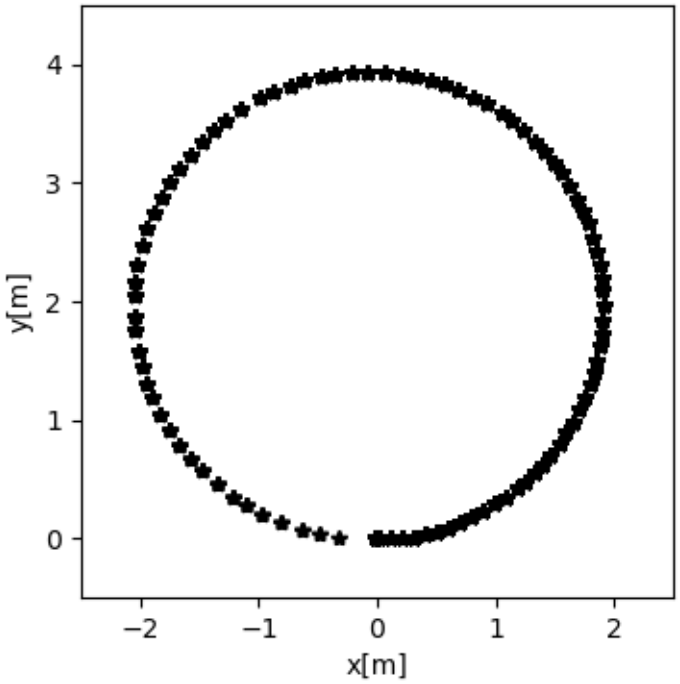

Figura 4: Gráfico Posição do robô no plano (x,y).

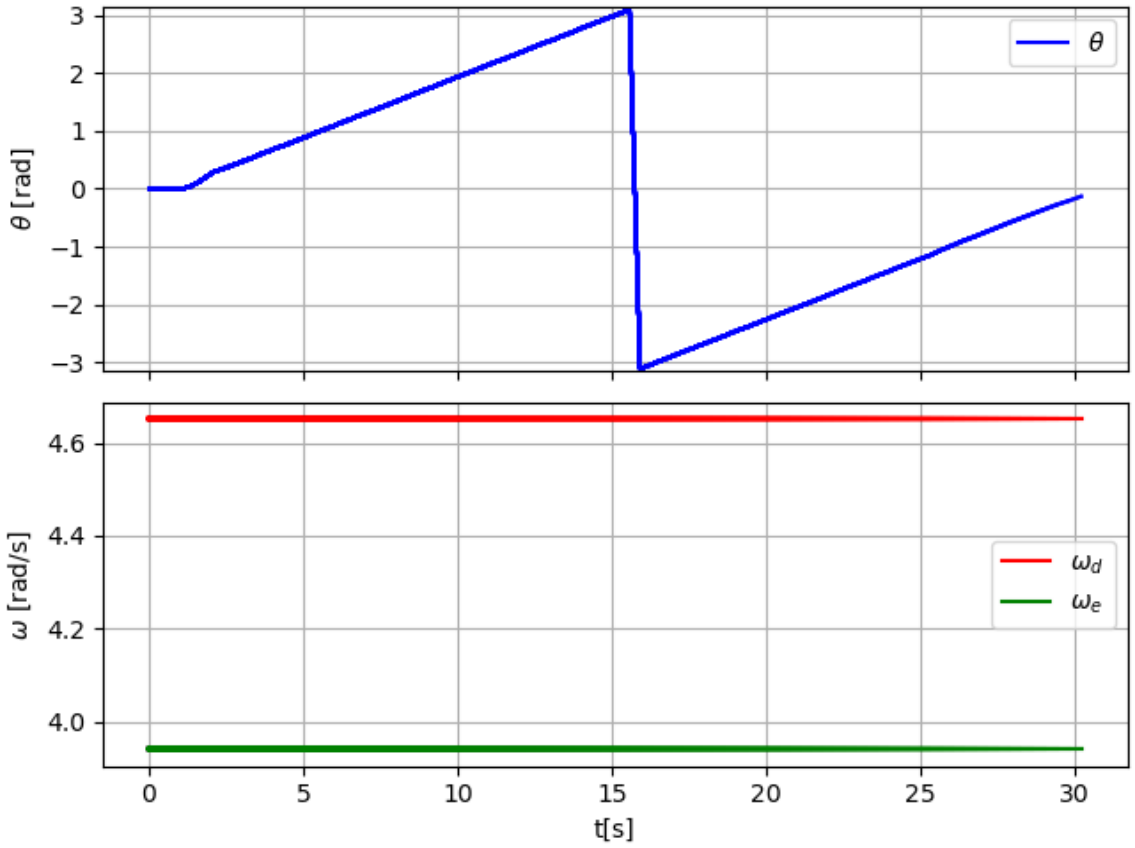

Figura 5: Gráfico da orientação e velocidade angular das rodas em função do tempo. (Autoria Própria)

A Figura 4 mostra a trajetória realizada pelo robô nos eixos x e y simultaneamente para as velocidades calculadas: $w_d = 4.65\ rad/s$, $w_e = 3.94\ rad/s$. Para os ômegas aplicados, $wd = 4.65$ e $we = 3.94$, podemos estimar o comportamento do robô do ponto de vista cinemático.

Movimento circular com raio a partir de (5) e resolvendo para r,

$$r = \frac{w_e + w_d}{w_d - w_e}.\frac{b}{2}$$
$$r \approx 2.0m$$

Velocidade linear de acordo com (3),

$$v = \frac{(4.65 + 3.94).0.09751}{2}$$
$$v \approx 0.42m/s$$

Tempo para percorrer uma circunferência de raio $2m$,

$$
\begin{aligned}
t &= v.d \\
t &= v.(2\pi r) \\
t &= 0.42(2\pi 2) \\
t &\approx 30s
\end{aligned}
$$

Podemos observar pelo gráfico de posição que o percurso realizado corresponde ao círculo de raio $2m$, e pelo gráfico de orientação pelo tempo podemos concluir que o tempo gasto para realizar uma volta completa, ou seja, o tempo que levou para o robô repetir a orientação, foi de aproximadamente $30s$, portanto os resultados obtidos corresponderam ao predito pelo modelo cinemático.

5 REFERÊNCIAS

Referências

[1] Python Org. Disponível em: <https://www.python.org/. Acesso em: 22 set. 2020.

[2] E. Rohmer, S. P. N. Singh, M. Freese, "CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework", IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2013. www.coppeliarobotics.com

[3] Matplotlib: Visualization with Python. Disponível em: <https://matplotlib.org/. Acesso em: 22 set. 2020.

[4] The NumPy Org. Disponível em: <https://numpy.org/. Acesso em: 22 set. 2020.

6 ANEXO 1

Código fonte: `https://github.com/Gabriellgpc/Sistemas_Roboticos_Autonomos_pos.git`

Vídeo de demostração: `https://www.youtube.com/watch?v=HhLZLSti9ks`