

Trabalho Prático 2 de Robótica Móvel

Ferramentas e Transformações

Marco T. Tristão¹

¹ Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, BH – Brasil

marcotuliopin@ufmg.br

Resumo. *O presente relatório descreve as atividades realizadas no Trabalho Prático 1 da disciplina de Robótica Móvel, focado em ferramentas e conceitos básicos de descrição espacial e transformações. O objetivo deste trabalho é proporcionar aos alunos uma introdução prática aos fundamentos da robótica móvel, utilizando a plataforma de simulação CoppeliaSim e a integração com a linguagem de programação Python.*

1. Introdução

Este relatório organizará a documentação do trabalho. Cada tópico será detalhado em uma seção distinta, de acordo com o pedido no enunciado. Também serão apresentadas outras informações julgadas relevantes, como descrição de partes da implementação e como executar o código produzido. As duas últimas seções compõem os comentários finais sobre o trabalho e a bibliografia utilizada.

Nesse trabalho, utilizamos duas cenas para o CoppeliaSim: **tp1.ttt** e **aula03_pioneer.ttt** (esta foi disponibilizada como parte do material da terceira aula da disciplina). Também usamos um notebook Python3 chamado **tp1.ipynb**. Todos esses materiais estão disponíveis no envio do trabalho.

2. Tópico Um: Criar a Cena

Criamos uma cena composta de seis modelos e enumeramos seus nomes a seguir:

1. **Pioneer_p3dx** (o robô utilizado como referência para o restante do trabalho).
2. Bill
3. Bowl
4. conferenceChair
5. r2d2
6. 240cmHighWallT / Manta

Adicionamos um sistema de coordenada a cada objeto para auxiliar a visualização da orientação dos mesmos.

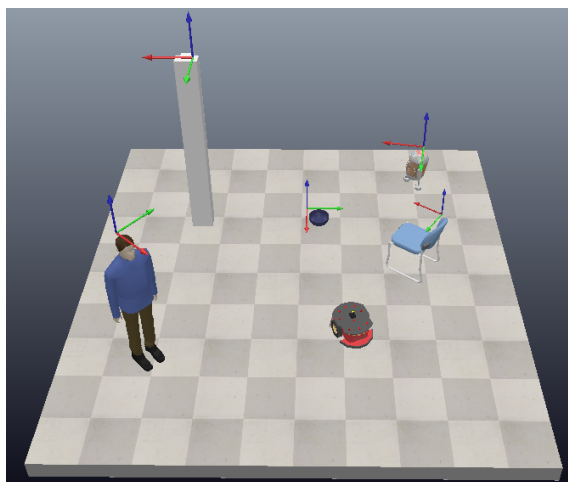
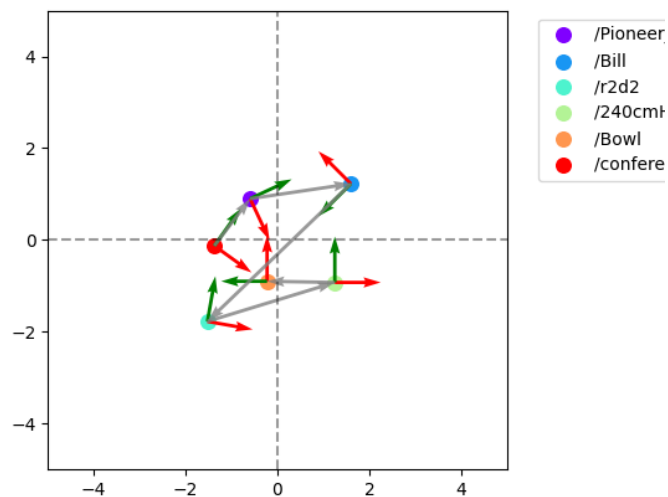


Figure 1. Primeira cena.

3. Tópico Dois: Definir os Sistemas de Coordenadas

Usamos da conexão com o simulador CoppeliaSim por meio da API ZeroMQ. Conseguimos a informação da posição e da orientação do modelo pela API e implementamos a construção de uma matriz de transformação homogênea. Na figura 2, plotamos a pose dos modelos e as relações entre os mesmos, representadas nas setas em cinza.

Figure 2. Pose dos modelos na cena utilizada no primeiro tópico.



Mostramos abaixo algumas das matrizes de transformação homogênea obtidas pelas transformações indicadas pelas setas em cinza:

```
[array([[ -0.939693 , -0.34201909,  0.          ,  1.32191736],
       [  0.34201909, -0.939693 ,  0.          ,  1.78153441],
       [  0.          ,  0.          ,  1.          ,  0.13879307],
       [  0.          ,  0.          ,  0.          ,  1.          ]]])]
```

a) Pioneer para Bill

```
[array([[ -8.19152044e-01, -5.73576436e-01,  3.69007732e-16,
         2.55657975e+00],
       [  5.73576436e-01, -8.19152044e-01,  6.50660192e-17,
         3.49707383e+00],
       [  2.64953102e-16,  2.64953102e-16,  1.00000000e+00,
         4.47034824e-08],
       [  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         1.00000000e+00]])]
```

b) Bill para R2D2

```
[array([[ 9.84807753e-01,  1.73648178e-01, -3.74700271e-16,
        -2.77500285e+00],
       [-1.73648178e-01,  9.84807753e-01,  0.00000000e+00,
        -8.50002978e-01],
       [ 3.69007732e-16,  6.50660192e-17,  1.00000000e+00,
        -1.20000057e+00],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         1.00000000e+00]])]
```

c) R2D2 to HighWall

```
[array([[ -1.02745420e-13,  1.00000000e+00,  0.00000000e+00,
        -2.49970302e-02],
       [-1.00000000e+00, -1.02745420e-13,  0.00000000e+00,
        -1.47500280e+00],
       [ 0.00000000e+00,  0.00000000e+00,  1.00000000e+00,
         1.19500052e+00],
       [ 0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
         1.00000000e+00]])]
```

d) HighWall to Bowl

Figure 3. Resultado dos sensores do robô para diferentes posições na cena.

4. Tópico Três: Mudar o Referencial dos Modelos

Tomando o robô R como referência, calculamos as transformações homogêneas das coordenadas dos demais objetos O na cena. Ou seja, o resultado desta operação são matrizes do tipo ${}^R_O T$, obtidas por meio dos cálculos abaixo:

$${}^R_O T = {}^R_W T * {}^W_O T$$

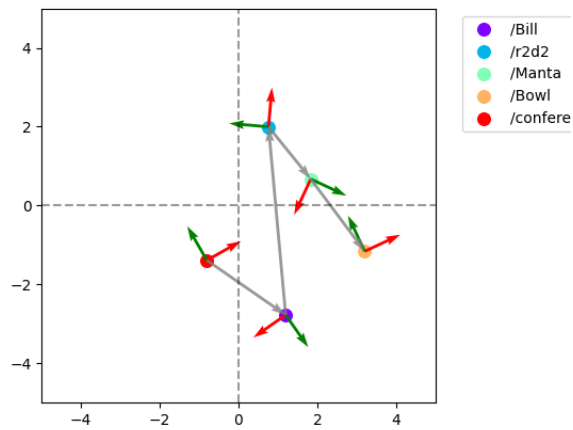
$${}^R_O T = {}^W_R T^{-1} * {}^W_O T$$

Além disso, usamos também das propriedades da matriz de transformação homogênea para invertê-la eficientemente.

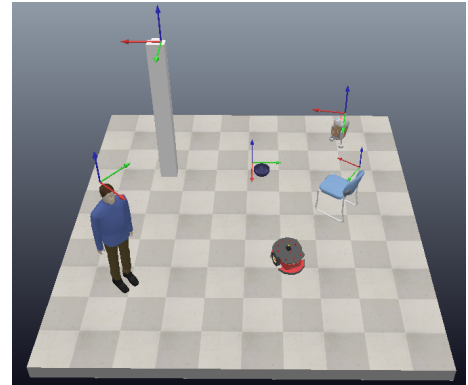
$${}^W_R T^{-1} = \begin{pmatrix} {}^W_R R^T & -{}^W_R R^T * {}^R P \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Os detalhes da implementação estão disponíveis da documentação do código.

Após as transformações, plotamos os objetos no referencial do robô para várias poses distintas. Apresentamos os resultados a seguir



a) Representação local do robô.

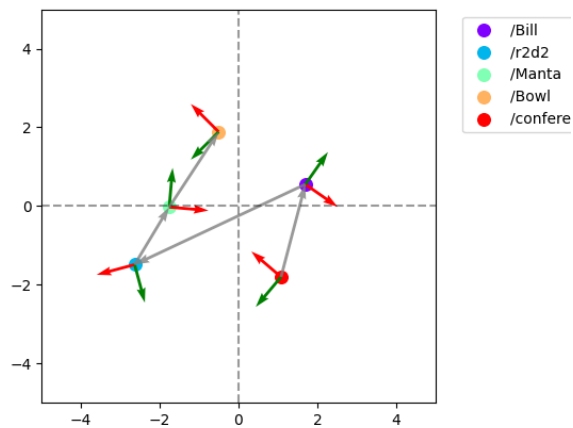


b) Primeira pose.

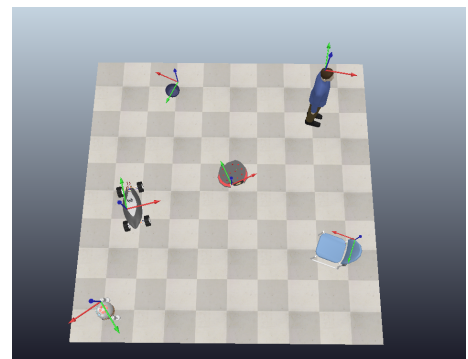
Figure 4. Representação dos objetos para a primeira pose.

5. Tópico 4: Variar a Posição do Robô

Variamos a posição do robô mantendo o restante da cena imóvel, mostrando o sucesso da transformação de coordenadas. As cenas e suas transformações estão apresentadas nas figuras a seguir.

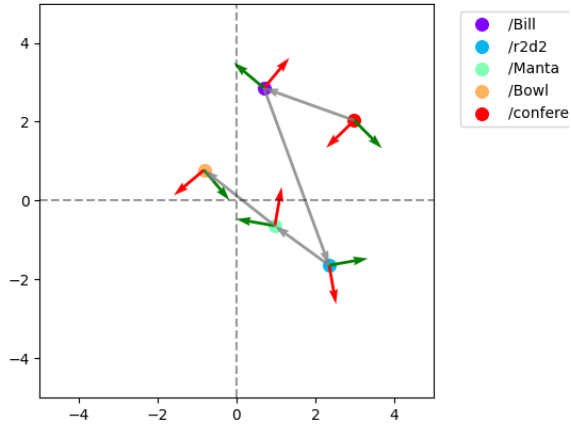


a) Representação local do robô.

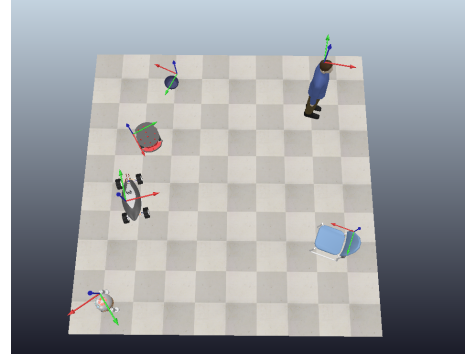


b) Segunda pose.

Figure 5. Representação dos objetos para a segunda pose.

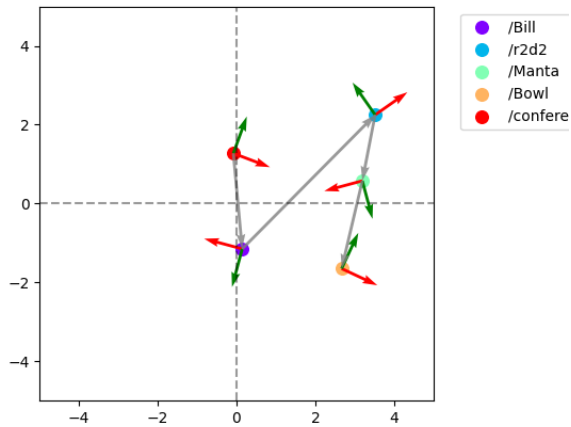


a) Representação local do robô.

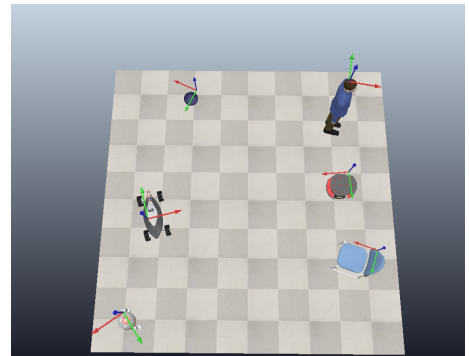


b) Terceira pose.

Figure 6. Representação dos objetos para a terceira pose.



a) Representação local do robô.



b) Quarta pose.

Figure 7. Representação dos objetos para a quarta pose.

6. Tópico Cinco: Utilizar o Sensor à Laser

Temos a matriz de transformação do laser para as coordenadas globais (${}^W_L T$) (obtida por meio da API de conexão remota) e a matriz de transformação do robô para o mesmo referencial ${}^W_R T$ (calculada em 3). Então, queremos calcular:

$${}^R_L T = {}^R_W T * {}^W_L T$$

$${}^R_L T = {}^W_R T^{-1} * {}^W_L T$$

Em nosso notebook provamos que ${}^W_R T * {}^R_L T = {}^W_L T$.

O sensor utilizado foi o *fastHokuyo*, que retorna a distância d do sensor ao objeto detectado e o ângulo θ no qual ele foi detectado. Para encontrar o ponto do objeto em

relação ao sensor, fazemos os cálculos:

$${}^L P = \begin{pmatrix} d * \cos(\theta) \\ d * \sin(\theta) \end{pmatrix}$$

Como os pontos serão plotados em uma superfície bidimensional, não nos preocupamos com o eixo Z no momento.

Aplicamos a transformação ${}^W P = {}^W_R T * {}^L P$ para encontrar a localização dos pontos de detecção do laser no referencial global. Com essas informações, variamos a posição do robô em relação à cena e captamos a detecção do sensor para criar as imagens a seguir.

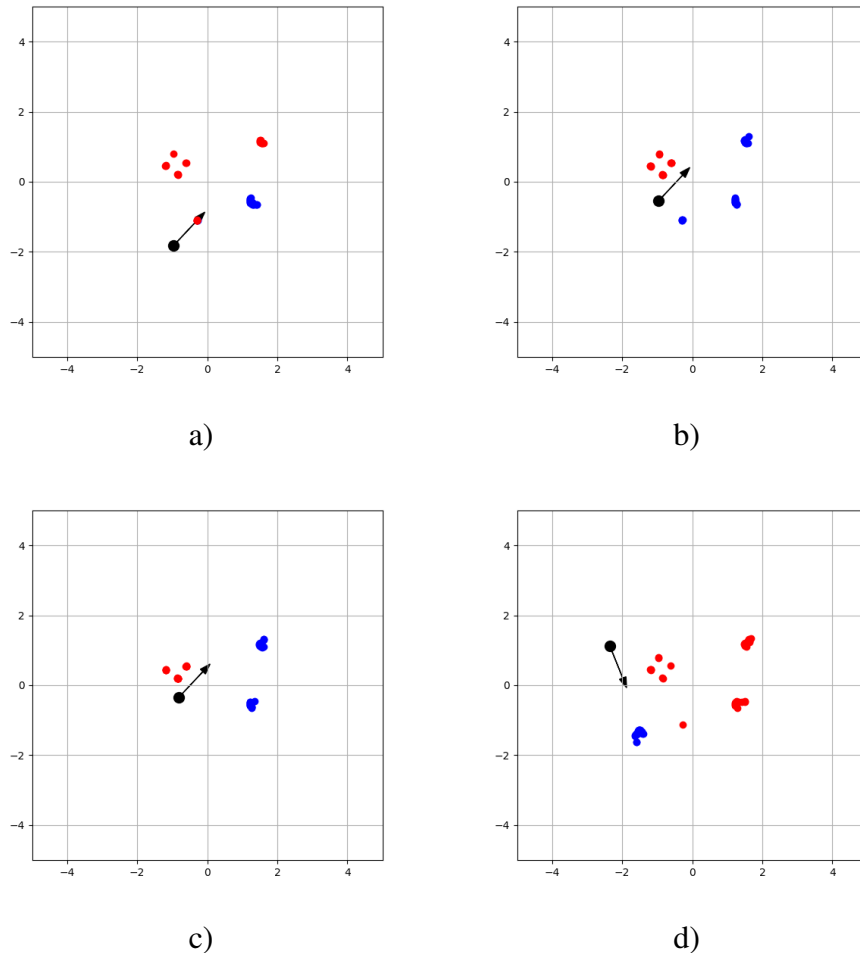


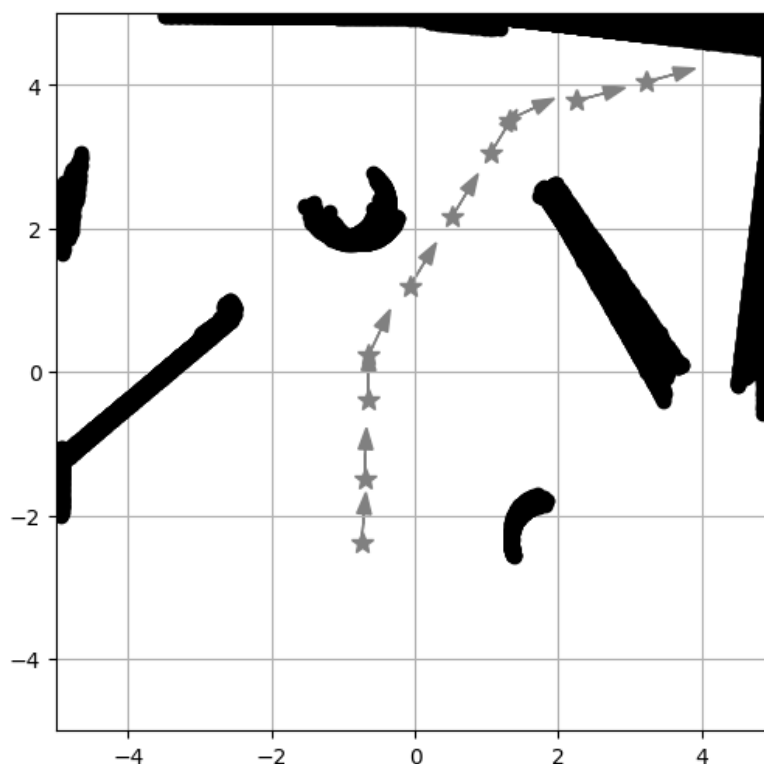
Figure 8. Resultado dos sensores do robô para diferentes posições na cena.

7. Tópico Seis: Desenhar a Trajetória do Robô

Nessa última tarefa, usamos a cena disponibilizada na terceira aula da disciplina, pois ela possui uma área maior coberta por objetos a serem observados. Também foi utilizado o script de locomoção do robô disponibilizado na mesma aula. Assim como no código usado nas demais tarefas, o script foi adaptado para obedecer à nova API.

Para traçar a trajetória do robô, a cada iteração do algoritmo de locomoção, nós guardamos as informações a seguir: dados do laser; pontos de detecção no referencial global; matriz de transformação das coordenadas do robô. Ao final da execução, usamos uma função adaptada do tópico anterior para desenhar todas essas informações tal que seja possível observar o trajeto e as detecções do robô. O resultado está na imagem 9.

Figure 9. Trajeto do robô pela cena.



8. Considerações Finais

O trabalho realizado proporcionou o aprofundamento do conhecimento nas técnicas de transformações rígidas e descrição espacial, assim como a familiarização com as ferramentas a serem utilizadas pelo decorrer da disciplina.

Os maiores desafios encontrados foram na transformação das detecções do laser para o referencial global e na descrição da trajetória do robô. A simplicidade do script de locomoção fez com que o robô se prendesse a cantos no cenário, assim não conseguimos analisar um caminho longo. No entanto, o caminho já é bem descritivo da cena como um todo. Além disso, a adaptação ao simulador CoppeliaSim levou alguma dificuldade ao processo, mas que foi superada no percorrer do mesmo.

References

- [1] COPPELIA ROBOTICS. *CoppeliaSim API Functions*. Disponível em: <<https://manual.coppeliarobotics.com/en/apiFunctions.htm>>. Acesso em: 24 mar. 2024.
- [2] MACHARET, Douglas G. Material da disciplina de Robótica Móvel. UFMG, 2024. (Material não publicado, fornecido em aula).