

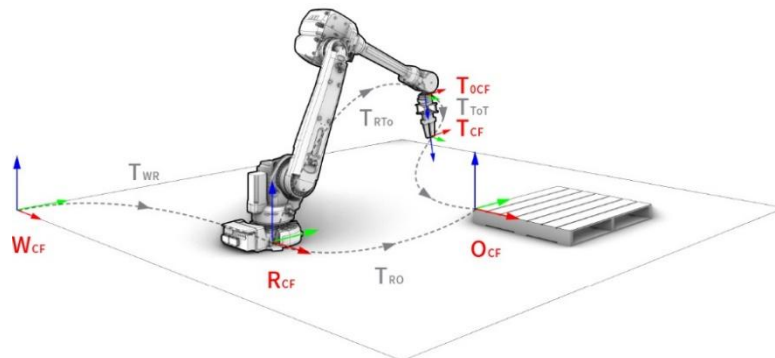
## Trabalho Prático 1 – Ferramentas e Transformações

Valor: 8,0 pontos  
Entrega: 09/09/2025

### 1. Instruções:

O objetivo deste trabalho prático é familiarizar o aluno com as ferramentas e conceitos básicos de descrição espacial e transformações. Instale o CoppeliaSim (<https://www.coppeliarobotics.com/>) e faça cada um dos exercícios abaixo. Deve ser utilizado Python 3 para as tarefas de programação.

- 1) [0,5 pts] Crie uma nova cena, adicione um robô móvel (e.g., Pioneer 3DX) e outros cinco elementos diferentes (e.g., móveis, pessoas, outros robôs, etc) espalhados pelo ambiente.
- 2) [0,5 pts] Defina os sistemas de coordenadas de todos os itens (você deve adicioná-los na cena e afixar aos elementos para facilitar a visualização). Em seguida, você deverá representar os sistemas de coordenadas e as transformações entre eles (não precisa ser todos para todos) utilizando um diagrama, similar à figura abaixo (pode ser uma versão mais simplificada).



Fonte: [https://gramaziokohler.github.io/compas\\_fab/latest/examples/01\\_fundamentals/02\\_coordinate\\_frames.html](https://gramaziokohler.github.io/compas_fab/latest/examples/01_fundamentals/02_coordinate_frames.html)

- 3) [1,0 pt] Considerando que a configuração do robô no referencial global é conhecida e dada por  $q = [x, y, \theta]$ , defina as matrizes de transformação homogêneas que representam as posições de todos os outros elementos da cena no referencial local do robô. Escreva um script que plota esses referenciais e os relacionamentos (vetor) entre eles (dica: use os códigos vistos em aula como base). Para os cálculos, as posições globais dos elementos podem ser recuperadas utilizando-se a RemoteAPI.

- 4) [0,5 pts] Coloque o robô em outras três posições diferentes da cena e faça os respectivos plots (referenciais e relacionamentos), verificando que a implementação funciona para diferentes casos. Lembre-se de também variar a orientação do robô em relação aos elementos, por exemplo, colocando ele de frente, de lado e de costas.
- 5) [1,5 pts] Substitua o robô adicionado inicialmente na cena pelo robô com laser mostrado em aula. No exemplo que vimos, o plot da leitura do laser está sendo feito no referencial local do laser. Defina as matrizes de transformações  ${}^R_L T$  (laser  $\rightarrow$  robô) e  ${}^W_R T$  (robô  $\rightarrow$  mundo), e em seguida modifique o script original para que os pontos agora sejam plotados no referencial global, de acordo com a posição atual do robô (recuperada pela RemoteAPI). Coloque o robô em outras diferentes posições da cena (e.g., três) e faça os plots das leituras do laser.
- 6) [2,0 pts] Por fim, utilizando o script básico de navegação visto em aula, faça um plot incremental que mostra o caminho executado pelo robô (sequência de posições) ao longo da navegação (por exemplo, usando uma linha tracejada) e combina todas as leituras de laser realizadas ao longo do trajeto. Essas informações devem ser todas representadas em um único gráfico considerando o referencial global.

Você pode (e deve!) adaptar os scripts de exemplo da melhor maneira que achar necessário, adicionando novas funções auxiliares, utilizando orientação à objetos, etc.

## 2. Documentação:

Organize a documentação de acordo com os itens descritos acima. Para cada tópico, apresente figuras/diagramas, equações/matrizes, gráficos, etc., de acordo com o pedido no enunciado.

Crie também uma seção descrevendo e detalhando sua implementação, deixando claro como executar. Você pode mencionar sobre possíveis estruturas de dados e algoritmos utilizados (de preferência com diagramas ilustrativos), bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado.

Por fim, introduza uma seção de conclusão, faça comentários gerais sobre o trabalho e as principais dificuldades encontradas. Você também pode mencionar a bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites se for o caso.

### O que deve ser entregue:

Envie um arquivo ZIP com nome ‘tp1-primeironome(s).zip’, contendo os seguintes arquivos:

- Arquivo README com o nome completo e número de matrícula do aluno (dupla).
- A(s) cena(s) do CoppeliaSim utilizada(s).
- O código fonte do programa em Python bem formatado e comentado.
  - O programa deve ser fácil de executar, ou seja, apenas chamando-se o script.
- A documentação do trabalho bem escrita e detalhada.

**Comentários Gerais:**

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza e boas práticas no programa também serão avaliados.
- Alunos de graduação podem fazer o trabalho em dupla.
- Trabalhos copiados serão penalizados conforme anunciado.

**Critérios de avaliação:**

- Implementação (6,0 pts).
- Documentação (2,0 pts).