

## Trabalho Prático 2 – Planejamento e Navegação

Valor: 14 pontos

Data de entrega: 02/05/2024

### 1. Instruções

O objetivo do trabalho é familiarizar o aluno com conceitos e técnicas de planejamento de caminhos, navegação e controle. O exercício consiste em implementar diferentes algoritmos para dois veículos diferentes. Caso necessário, implemente um controlador para executar o caminho planejado.

Algoritmos a serem implementados e tipos de veículos a serem utilizados:

1. Graduação e Pós-graduação
  - a. Roadmap – Robô Holonômico
  - b. Campos Potenciais – Robô Diferencial
2. Pós-graduação
  - a. RRT – Robô Holonômico

Para cada algoritmo, realize pelo menos **dois** experimentos, considerando diferentes cenários (crie cenas com diferentes níveis de dificuldade). Em cada experimento, escolha posições inicial e final que não sejam triviais e “estressem” o método. Algumas sugestões de mapas estão disponíveis no Moodle, você pode usá-los como inspiração para montar suas cenas no CoppeliaSim. Faça uma breve análise/comparação dos resultados obtidos, discutindo eficiência e eficácia dos métodos.

Como algoritmo de Roadmap você pode escolher entre quaisquer uma das técnicas que vimos, por exemplo, grafo de visibilidade, grid, probabilistic roadmap. No caso da RRT, você pode implementar a técnica clássica ou qualquer uma das suas extensões, por exemplo, RRT\*, Informed RRT\*, etc.

No algoritmo de Campos Potenciais, você pode utilizar um dos controladores abaixo para fazer a transformação entre a força (vetor velocidade) calculada e as velocidades linear e angular do veículo.

- [De Luca e Oriolo, 1994]

$$\begin{aligned}v &= k_p(\dot{x} \cos \theta + \dot{y} \sin \theta) \\ \omega &= k_\theta(\text{atan2}(\dot{y}, \dot{x}) - \theta)\end{aligned}$$

- [Desai et al. 1998]

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\frac{\sin \theta}{d} & \frac{\cos \theta}{d} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

## 2. Documentação

A documentação deve conter:

1. Introdução: detalhamento do problema e visão geral sobre o funcionamento do programa.
2. Implementação: descrição detalhada sobre a implementação. Deve ser discutido as 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.
4. Testes: descrição dos testes realizados e ilustração dos resultados obtidos (não edite os resultados). Você deve propor experimentos considerando diferentes cenários.
5. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas.
6. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites, etc.

### O que deve ser entregue:

Envie um arquivo ZIP com o nome 'tp2-primeironome(s).zip', contendo os seguintes arquivos:

- Arquivo README com o nome completo e número de matrícula do aluno (dupla).
- As cenas do CoppeliaSim utilizadas.
- 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 (9 pts).
  - Funcionamento correto, aplicação dos conceitos, estruturação, eficiência, ...
- Documentação (5 pts).
  - Texto (3 pts): clareza e coesão na explicação, análises, gráficos, ...
  - Apresentação (2 pts): vídeo de até 8min detalhando o trabalho e os resultados.
    - a. Obs: Os vídeos serão disponibilizados no canal da disciplina!