

Trabalho Avaliado 1 – Sistema de Navegação e Controle da Missão com ROS 2

Objetivo Geral

Projetar e implementar um robô autônomo que **explora o ambiente, detecta uma bandeira** por visão computacional e **se posiciona adequadamente para capturá-la**, com controle baseado em **máquina de estados** e percepção sensorial.

1. Controle da Missão com Máquina de Estados

Este é o **módulo central** do trabalho. Seu robô deve **modelar os estados da missão** e transitar entre eles de forma **reativa ou planejada**, com base nos dados dos sensores e nos objetivos da tarefa.

Sugestão de estados :

1. EXPLORANDO

- O robô explora o ambiente e procura pela bandeira.
- Movimento pode ser aleatório ou sistemático (ex: linha serpentina).

2. BANDEIRA_DETECTADA

- Bandeira foi visualmente identificada.
- Robô calcula posição da bandeira em relação à câmera.

3. NAVEGANDO_PARA_BANDEIRA

- Robô traça rota até a bandeira.
- Desvia de obstáculos com auxílio do LIDAR.

4. POSICIONANDO_PARA_COLETA

- Ajusta sua orientação e distância até ficar de frente para a bandeira.
5. *(Opcional – para integração futura com o Trabalho 2):*
CAPTURANDO_BANDEIRA, RETORNANDO_PARA_BASE

Outras sugestões de estados (extra ou substituição):

- AGUARDANDO_COMANDO – robô permanece parado até receber start via tópico
- RECALCULANDO_ROTA – se houver bloqueio inesperado
- REDETECTANDO_BANDEIRA – caso a bandeira seja perdida do campo de visão

Cada estado deve ser documentado e controlado com clareza no código.

2. Representação do Mapa e Planejamento Simples

- Mapa representado por **matriz 2D** ou outra estrutura leve
- Planejamento pode ser feito com:
 - Movimento por gradiente
 - Algoritmo simples (ex: busca A*, Dijkstra)
 - Exploração por zonas ou direção heurística

3. Estrutura Geral do Trabalho

Familiarização com ROS 2

- Criação e compilação de pacotes
- Setup de nós e estruturas de projeto

Modelagem do Robô

- Modificar URDF/Xacro fornecido

- Adicionar sensores: câmera, LIDAR, IMU
- Publicar TF corretamente

Deteção Visual da Bandeira

- Usar plugin de labels do Gazebo
- Nó Python ROS 2 para processar imagens e identificar a bandeira

Navegação e Controle (ênfase deste trabalho)

- Implementar **máquina de estados**
- Lógica de navegação reativa ou simples planejamento
- Deteção de obstáculos com LIDAR
- Posicionamento final diante da bandeira

Entrega

- Código funcional organizado em um **pacote ROS 2**
- **Máquina de estados** implementada e **documentada** no código
- Arquivo **README** com instruções de compilação, execução e descrição da arquitetura geral do sistema
- A entrega deve ser realizada **via Moodle**, por meio do **link do repositório GitHub** contendo o pacote desenvolvido

Código Base para Desenvolvimento

Realize o clone do pacote base utilizando o link abaixo:

- `git clone https://github.com/matheusbg8/prm`

Este pacote contém um robô diferencial com sensores básicos (câmera, LIDAR e IMU), e pode ser modificado livremente conforme as exigências do trabalho.

Critérios de Avaliação

1. Organização do Projeto ROS 2 (2.0 pontos)

Subcritério	Descrição	Pontuação
Estrutura do pacote ROS 2	Uso correto de <code>setup.py</code> , <code>package.xml</code> , <code>launch/</code> , <code>scripts/</code> , <code>urdf/</code> etc.	0.5
Compilação e execução	Compila corretamente com <code>colcon</code> , execução documentada (README)	0.5
Uso de launch files	Lançamento automatizado do robô, sensores e nós de controle	0.5
Organização geral do repositório	Código limpo, comentado, com README claro e instruções	0.5

2. Modelagem do Robô e Sensores (2.0 pontos)

Subcritério	Descrição	Pontuação
Modificações no URDF/Xacro	Alterações bem feitas no robô base (estrutura, links, joints)	0.5
Publicação de TFs	Correta publicação dos frames, compatibilidade com <code>robot_state_publisher</code>	0.5
Sensores	Inclusão/configuração de câmera, LIDAR, IMU, etc.	0.5
Qualidade geral da modelagem	Robô estável, sem warnings/erros visuais, integra com a simulação	0.5

3. Controle da Missão com Máquina de Estados (3.0 pontos)

Subcritério	Descrição	Pontuação
Implementação da máquina de estados	Lógica clara e funcional, com transições bem definidas	1.0
Divisão adequada dos estados	Cobertura de pelo menos: explorando, detectando, navegando, posicionando	1.0
Coerência e robustez	O robô reage corretamente a eventos (ex: bandeira perdida, obstáculos)	1.0

4. Detecção Visual da Bandeira (1.5 pontos)

Subcritério	Descrição	Pontuação
Nó de processamento de imagem	Nó Python funcional, com subscrição ao tópico da câmera	0.5
Detecção da label correta no ambiente	Interpreta corretamente as labels do plugin do Gazebo	0.5
Uso da informação visual	Direciona o comportamento do robô com base na detecção	0.5

5. Navegação e Desvio de Obstáculos (2.0 pontos)

Subcritério	Descrição	Pontuação
Representação do ambiente	Uso de matriz, grid ou estrutura adequada de mapa	0.5
Navegação básica	Capacidade de ir do ponto inicial até a bandeira	0.5
Desvio de obstáculos com LIDAR	Uso do sensor para evitar colisões ou recalcular rota	0.5
Posicionamento final	Robô para a uma distância correta e com orientação adequada	0.5

6. Criatividade e Funcionalidades Extras (0.5 ponto)

Subcritério	Descrição	Pontuação
Melhorias além do mínimo	Animações, feedback visual, visualizações RViz, melhorias no robô	até 0.5

Pontuação Total: 11,0 pontos (nota será convertida para 10)

Observação: A pontuação extra permite diferenciar projetos excepcionais sem penalizar os que cumprirem todos os requisitos mínimos com qualidade.

Critérios de Penalização

- Código que **não compila**: -1 ponto
- Robô que **não se move ou trava** na simulação: -1 ponto
- README **incompleto ou ausente**: -0.5 ponto

Dúvidas sobre o trabalho, encaminhar para [matheus.m.santos \[at\] icmc.usp.br](mailto:matheus.m.santos@icmc.usp.br)