

Robótica Cognitiva – 2022/2023

Autonomous Mobile Robot for Desinfection

Final Submission

Duarte Cruz
2017264087Gonçalo Arsénio
2017246034Pedro Gomes
2018298280

1 Introdução

Para este projeto foi nos proposto a realização de um robô móvel autónomo para desinfeção, ou seja, um robô que navegue autonomamente por um mapa, construído *a priori*, dos espaços em que efetuará a desinfeção, tendo de conseguir navegar seguramente perante a presença de obstáculos estáticos e dinâmicos. Quando o robô é inicializado vai procurar o código QR em seu redor, que contém as informações necessárias para a plataforma realizar com sucesso a desinfeção, tais como, potência da lâmpada, a energia necessária para eliminar os vírus, a ordem das salas a desinfetar e o mecanismo de desinfeção, existindo 3 possíveis (Follow Waypoints, Follow Wall e Desinfect Room) explicados em detalhe no relatório.

2 Background

2.1 bringup_explorer_hw

Este ficheiro launch é o inicializador principal do modo explorar. Este ficheiro inicia o nó do visualizador "rviz" e mais dois launch files. Sendo estes do "move_base_explorer_hw" e o "detector", que pertence à package de deteção de pessoas que utilizamos "pal_detector_opencv".

2.1.1 move_base_explorer_hw

Neste ficheiro está presente todo o movimento do robô para o método de exploração. Neste ficheiro são inicializados os seguintes pacotes:

gmapping

Este pacote fornece SLAM baseado em laser (Localização Simultânea e Mapeamento), como um nó ROS chamado slam_gmapping. Utilizando slam_gmapping, pode criar um mapa de grelha de ocupação 2D (como uma planta de construção) a partir de laser e apresentar dados recolhidos por um robô móvel.

Neste ficheiro chamamos este pacote com vários parâmetros, calibrados por nós, para otimizar o seu desempenho no robô físico.

Como o gmapping não faz a navegação, tivemos que incluir mais dois pacotes essenciais para em conjunto comandarem a navegação do robô.

explore_lite

Este pacote proporciona uma exploração gananciosa baseada em fronteiras. Enquanto o nó estiver a funcionar, o robô explorará gananciosamente o seu ambiente até que nenhuma fronteira seja encontrada. Os comandos de movimento serão enviados para move_base.

move_base

Este pacote permite fazer o controlo da navegação da plataforma móvel. Este é o nó que subscreve ao tópico `/move_base/goal`, que indica o objetivo que robô deve atingir no mundo, envia os comandos de velocidade da plataforma e fornece informações sobre *status* das metas que são enviadas para a ação move_base através do tópico `/move_base/status`.

Estes três pacotes em conjunto, proporcionam uma exploração eficaz de toda a área em redor

do robô. Assim consegue navegar seguramente pela área, desviando-se de obstáculos e desenhando um mapa para depois ser usado para a desinfecção.

2.1.2 pal_person_detector_opencv

Este pacote subscreve a um tópico de imagem e aplica o detetor de pessoas OpenCV baseado numa cascata Adaboost de HoG. O nó publica ROIs das deteções e uma imagem de depuração mostrando a imagem processada com os ROIs suscetíveis de conter uma pessoa. Caso a pessoa seja detetada aparece com um quadrado verde à sua volta.

Mesmo que este algoritmo não seja usado para o movimento do explorador, achamos interessante manter esta "feature" para mostrar no rviz o robô a detetar as pessoas ao seu redor. Contudo, este será necessário e ativamente utilizado para o resto do trabalho.

2.2 bringup_hw

Este ficheiro launch é responsável por inicializar uma série de nós e outros launcher para todo o processo de desinfecção do robô. Habilita este a conseguir efetuar a desinfecção com base nas instruções obtidas através dos códigos QR.

Pacotes a serem inicializados:

1. zbar_ros
2. map_server
3. rviz
4. robot_pose_publisher

Launchers a serem inicializados:

1. amcl
2. move_base_hw
3. detector

zbar_ros

Este pacote subscreve ao tópico da câmara do robô, para detetar códigos QR. Com o processamento da câmara, consegue retirar os frames que contêm o código QR e consegue extrair a

sua informação. Depois publica a informação desses códigos detetados num tópico para serem usados.

map_server

Este pacote fornece o nó ROS map_server, que oferece dados do mapa como um serviço ROS. Isto habilita aos outros nós associados ao mesmo ros master, a acederem ao mesmo mapa. Incluindo a sua visualização no rviz.

robot_state_publisher

Este pacote permite-lhe publicar o estado de um robô usando a TF2 (second generation of the transform library). O pacote toma os ângulos articulares do robô como entrada e publica as poses 3D das ligações do robô, usando um modelo cinemático de árvore do robô. Este nó permite sabermos a posição do robô em tempo real, para ser usado nos algoritmos de navegação e desinfecção.

2.2.1 amcl

Este pacote é responsável pela localização da plataforma móvel no mapa. É um sistema de localização probabilística que implementa o algoritmo Adaptativo de Monte Carlo. O nó **amcl** lê um mapa, leituras de Lidar e mensagens **tf** para estimar a pose da plataforma móvel. Neste inicializador declaramos vários parâmetros, calibrados por nós, para otimizar o seu desempenho no robô físico.

2.2.2 move_base_hw

Este ficheiro é parecido com o "move_base_explorer_hw", mas só usando a parte do pacote "move_base". Visto que, para o movimento da desinfecção, o processamento do movimento é feito pelo "guide node" e este launcher só serve para aceitar comandos de movimento da plataforma vindos desse nó.

2.2.3 pal_person_detector_opencv

Este pacote é o mesmo que já foi explicado anteriormente. Contudo, agora vai ser utilizado

para o movimento do robô. Como vai ser explicado em seguida.

3 Guide Node

Tendo o mapa previamente construído, o objetivo agora é que a plataforma tenha capacidade de extrair as informações fornecidas pelo utilizador, através dos códigos QR, e efetuar a desinfecção consoante as instruções obtidas, no final o robô deverá sempre voltar ao ponto inicial. A desinfecção pode ser feita das seguintes formas:

3.1 Follow Waypoints

Mecanismo simples, que consoante os pontos extraídos do código QR, deve assegurar que o robô alcança todos esses pontos e efetua a desinfecção, caso algum desses pontos seja inalcançável a plataforma deve conseguir passar ao seguinte.

O algoritmo faz uso do *MoveBaseClient* para enviar comandos de posição para o *move_base* e receber um callback de objetivo atingido.

3.2 Follow Wall

Tal como o nome indica, este mecanismo de desinfecção trata-se de um algoritmo que permita ao robô seguir uma parede em seu redor, enquanto efetua a desinfecção, a plataforma vai seguir uma parede à sua direita ou esquerda, consoante a que estiver mais próxima.

Primeiramente, é necessário extrair informação do nó LaserScan do ROS e guardá-la. O laser disponibiliza distâncias para os 180° da frente do robô, no entanto, não precisamos de toda esta informação, para tal foi feita a divisão em 3 secções (frente, esquerda e direita), guardando apenas a distância mais pequena dentro de cada secção, consoante a secção que estiver mais próxima de um obstáculo será adotado um algoritmo de seguimento à esquerda ou à direita.

O algoritmo é inicializado com uma variável global *follow_dir* a -1, onde vai procurar uma parede para seguir, este valor mudará quando uma parede for selecionada. É utilizado um threshold mínimo e máximo para definir a distância da plataforma à parede.

Quando o robô detetar uma parede, a variável muda para 0 (parede à esquerda) ou 1 (parede

à direita). Conforme o valor desta variável e das leituras das distâncias obtidas é sempre selecionada uma das seguintes ações: 0 - Find Wall, 1 - Turn Right, 2 - Follow Wall, 3 - Turn Left, 4 - Diagonally Right, 5 - Diagonally Left, 6 - Stop.

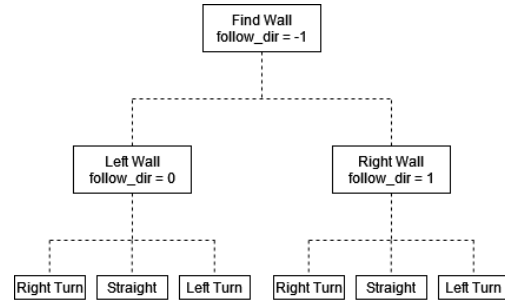


Figure 1: Diagrama do funcionamento básico do algoritmo

3.3 Desinfect Room

Com este mecanismo o robô simplesmente deve dirigir-se ao centro da sala e efetuar a desinfecção. Neste algoritmo, tal como no follow waypoints, é usado o *move_base* que vai receber o ponto do centro da sala.

4 Detecção de pessoas

Para além das funcionalidades mencionadas, o robô tem de ser capaz de detetar pessoas em seu redor, isto pois os seres humanos não devem ser expostos a radiação, sempre que uma pessoa for detetada no momento da desinfecção o robô deve pará-la imediatamente e mandar um aviso para a pessoa sair da sala.

Com esta finalidade foi utilizada a package *pal_person_detector_opencv*, que subscreve a um tópico de imagem e aplica o algoritmo de deteção de pessoas do openCV que usa o descritor de features HoG (Histogram of Oriented Gradients) e o algoritmo Adaboost para treinar um classificador. Este nó publica as regiões de interesse das deteções e uma imagem de debug com as regiões de interesse mais prováveis de conter uma pessoa.

5 Testes e Resultados

Os testes foram todos realizados no laboratório

ISRsea num mapa desenhado pelo Professor Sedat e com QR codes fornecidos por ele.



Figure 2: Mapa Real

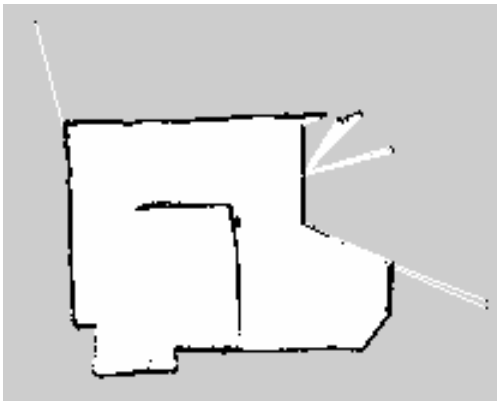


Figure 3: Mapa Gerado pelo Explorador

O robô usado para os testes foi o pioneer-3DX que tinha duas câmaras disponíveis, RealSense e Astra. Sendo que, optamos por escolher a última por motivos externos.

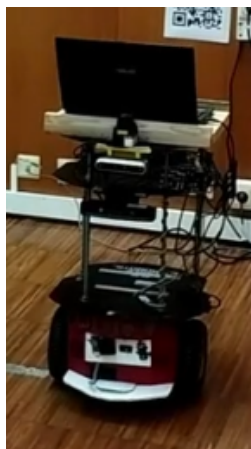


Figure 4: Robô

Como é possível ver no vídeo disponibilizado na submissão, o robô consegue efetuar com

sucesso os três mecanismos de desinfecção. No entanto, apresenta algumas dificuldades na paragem da desinfecção quando deteta uma pessoa, pois devido às falhas de internet e aos frames por segundo da câmara serem muito baixos. Por consequência, muitas vezes quando o robô deteta uma pessoa já terminou a desinfecção nesse local, impossibilitando a paragem da mesma e o envio do aviso para a pessoa sair da sala.