

Relatório - Projeto CC7711

Detecção de colisão com caixas

Hugo Linhares Oliveira - RA: 22.120.046-2

João Pedro Rosa Cezarino - RA: 22.120.021-5

Thales de Oliveira Lacerda, RA: 22.120.056-1

Vitor Martins Oliveira, RA: 22.120.067-8

Esse projeto implementa um robô E-Puck utilizando o software Webots, que é uma plataforma de simulação de robótica. O robô é capaz de detectar e reagir a obstáculos em seu caminho, bem como reconhecer um objeto específico: uma caixa com uma massa mais leve que as demais presentes.

Primeiramente, o código define algumas constantes globais, como: **PASSO_TEMPO**, que define a taxa de atualização dos sensores e atuadores do robô em milissegundos. **VELOCIDADE_MAX** é a velocidade máxima em radianos por segundo que as rodas do robô podem alcançar. **LIMIAR Obstaculo** é o valor limite dos sensores de proximidade para determinar se um obstáculo está próximo. **QTD_LEDS** é a quantidade de LEDs no robô utilizado. Por fim, **TOLERANCIA** é um valor de referência para identificar se a caixa foi movida e, por consequência, definir que a caixa de menor massa foi encontrada.

Em seguida, existem certas funções que demandam definição:

- **inicializar_robô()**: Esta função inicializa o robô e seus componentes. Ela retorna o objeto robô, os motores das rodas esquerda e direita e uma lista de LEDs.
- **inicializar_sensores_proximidade(robô)**: Esta função inicializa os sensores de proximidade do robô e os retorna como uma lista.
- **ajustar_velocidade(valores_sensores)**: Esta função determina a velocidade das rodas esquerda e direita do robô com base na detecção de obstáculos pelos sensores de proximidade. A velocidade é ajustada para evitar colisões.

A função **main()** realiza a maior parte do trabalho. Ela inicializa o robô e seus sensores, além de configurar o Supervisor para monitorar a caixa de menor massa. Em seguida, entra em um loop infinito em que lê os valores dos sensores de proximidade, ajusta a velocidade do robô e verifica a posição da caixa. Se a caixa se mover além de um determinado limite de tolerância após o contato do robô, os motores do robô são parados, o estado de um LED é alterado e uma mensagem é impressa no terminal indicando a colisão com a caixa de menor massa.

Portanto, este código realiza o controle de um robô E-Puck que detecta obstáculos e reage a eles, além de monitorar um objeto específico (caixa de menor massa) e reagir a mudanças na posição deste. Os fundamentos deste código podem ser aplicados em cenários de robótica móvel, como navegação autônoma e seguimento de objetos em tempo real.

Observações para a execução da simulação:

Se a propriedade **"Supervisor"** não estiver definida como **TRUE** ou se o nome (DEF) do NODE da caixa de menor massa não estiver definido como **"wooden-box"**, o código não funcionará corretamente, pois essas são configurações essenciais para a operação correta do robô.

Link do repositório completo:

<https://github.com/akajhon/E-Puck-Robot-Control>

Link do controlador:

[https://github.com/akajhon/E-Puck-Robot-Control/blob/main/CC7711 Project/controllers/Controlador/Controlador.py](https://github.com/akajhon/E-Puck-Robot-Control/blob/main/CC7711_Project/controllers/Controlador/Controlador.py)