

Trabalho Prático I - Navegação

Prazo: 15/04/2019

Considerações gerais

O trabalho deverá ser feito em DUPLA.

A avaliação do trabalho será feita através da análise do funcionamento das implementações: primeiramente no simulador MobileSim, e posteriormente com o robô real. O teste no robô real só poderá ser feito quando a implementação estiver funcionando bem no simulador.

Além de apresentar o funcionamento da implementação daquilo que for pedido, entregue via Moodle um arquivo compactado com os códigos desenvolvidos.

1. Desvio de obstáculos

Objetivo: Implementar uma estratégia reativa de navegação aleatória com desvio de obstáculos.

Complete a função `wanderAvoidingCollisions` da classe `Robot` no arquivo `Robot.c`. Essa função é chamada a cada passo do robô (que ocorre de 100 em 100ms), e deverá ser responsável por controlar a velocidade do robô.

A ideia básica é monitorar a distância medida pelos sensores (sonar/laser), e modificar a direção do robô sempre que o mesmo se aproximar muito de obstáculos. A forma como isso será feito fica a critério de cada um, só não pode deixar o robô colidir, enquanto ele anda autonomamente pelo ambiente.

Para testar o funcionamento da estratégia de navegação implementada, basta modificar o modo de movimentação para WANDER, apertando a **tecla 3** durante a execução do framework.

DICA: para facilitar o trabalho, estão disponíveis as funções `getMinSonarValueInRange(i, j)` e `getMinLaserValueInRange(i, j)`, descritas na classe `PioneerBase`, que retornam a menor distância medida pelos sensores do índice `i` ao índice `j`.

2. Controle PID

Objetivo: Implementar uma estratégia de *wall-following* usando controle PID.

Complete a função `wallFollow` da classe `Robot` no arquivo `Robot.c`.

Aqui a ideia básica é fazer com que o robô mantenha-se navegando em linha reta a uma distância x das paredes utilizando controle PID. Para tal, a cada instante é preciso computar o CrossTrack Error, ou seja, a diferença entre a distância medida até a parede pelos sensores e a distância desejada x . Assim como no exercício anterior,

sugere-se a utilização das funções auxiliares `getMinSonarValueInRange(i, j)` ou `getMinLaserValueInRange(i, j)`.

O ajuste dos parâmetros referentes aos ganhos P, I, D, pode ser feito de forma empírica mesmo (i.e. no tradicional método do chute).

Para testar o funcionamento da estratégia de navegação implementada, basta modificar o modo de movimentação para `WALLFOLLOW`, apertando a **tecla 4** durante a execução do framework.