

## Segundo Trabalho de FCR

Nome: Tulio Mariano da Silva Lima

Matrícula: 12/0054337

### Objetivos

O robô pioneer deve se deslocar por dez nós do mapa topológico criado no Roteiro 4 do Stage, construindo a grade de ocupação em cada um dos nós. A posição inicial e a posição do robô a cada instante de tempo são conhecidas (fornecidos pela odometria).

Para cada região no grafo (o mesmo usado no roteiro 4, e aqui considerado conhecido pelo robô), o robô deve construir uma grade de ocupação. Enquanto explora o ambiente para construir a grade de ocupação, o robô não deve tocar nenhum tipo de obstáculo detectado. As informações sobre cada ambiente constante no mapa fornecido não podem ser usadas na construção da grade de ocupação: somente os dados dos sensores podem ser usados para determinar a grade de cada ambiente/região do mapa.

Após o mapa estar construído, o robô deve ir até uma posição definida pelo usuário (usando mouse, coordenadas, ou outra forma de entrada). A cada instante da movimentação, o robô deve indicar o nó do mapa topológico onde está e indicar na grade de ocupação sua posição atual (de acordo com a odometria: não é necessário usar os sensores para identificar a localização do robô na grade).

O planejamento de trajetória pode ser realizado somente no grafo, a partir de dois nós de entrada (posições atual e final) e definindo uma sequência de nós que leve o robô do nó inicial até o nó final.

### Introdução

A implementação de uma rota até um objetivo pode envolver uma série de barreiras a serem superadas até a chegada ao destino, e pode ainda acarretar em perceber as alterações em um ambiente que é dinâmico. Embora pequena, a frase acima é um desafio muito grande no campo da robótica e pode ter inúmeras soluções, nas quais os resultados são discutíveis, e o foco desse trabalho é achar pelo menos uma forma de 'resolver' tal problema.

A plataforma usada na atividade será um ambiente simulado, contendo um robô pioneer com uma série de sensores que auxiliam na percepção do mundo simulado e visível pelo Stage. O robô deve ser orientado durante sua trajetória, e não deve encostar em nenhum objeto, podendo ser atribuído um limite de distância para o qual pode permanecer, evitando a todo custo colisões.

Partir de um ponto até outro envolve uma estratégia, nessa estratégia é usado o que se tem disponível ou que pode ser alcançado, ou seja, para sair de um ponto até outro é possível usar um mapa do ambiente, um grafo que simboliza caminhos, um sistema de coordenadas ou mesmo nada. Ambos com implementações diferentes, mas apenas o que envolve o uso de um sistema de coordenadas será o foco por enquanto, dessa forma o usuário deverá passar coordenadas em x e y para que o programa encontre e percorra o caminho calculado.

Além do uso de sensores para o reconhecimento do ambiente, uma outra abordagem é o uso de mapa de ocupação para auxiliar o usuário na orientar-se. E basicamente consiste em capturar os dados dos sensores disponíveis e com isso criar um mapa, para identificar objetos e rotas. Essa grade de ocupação é auxiliada pelo mapa topológico do ambiente, que constite basicamente de um grafo com todas as posições acessíveis.

### Aparato Experimental

Foram usados nesse trabalho:

- Um computador com ROS configurado
- Python(produção do programa que orquestra a movimentação e orientação)
- Stage e Rviz(para acompanhamento do robô)

## Metodologia

Dado o objetivo descrito no roteiro de trabalho, foi desenvolvido um programa em python que atende as propostas da atividade. Nesse programa, o usuário é questionado pelo nó que deseja ir e angulo para os quais ele deseja encaminhar o robô, e para isso o programa informa todos os pontos acessíveis e o nó atual que o robô está.

```
tuliolina@tuliolina-VPC5B35F8:~/catkin_ws$ rosrun fcr2017 trabalho2.py
Iniciando o Programa para mover o Pioneer
O Robo esta no noh: D1
Os nos acessiveis do ponto onde voce esta: ['D14', 'D15', 'D16', 'D17', 'D18', 'D11', 'D12', 'D13', 'E8', 'D18', 'D19', 'A14', 'A11', 'A10', 'A13', 'A12', 'E5', 'A1', 'A3', 'A2', 'A5', 'A4', 'A7', 'A6', 'C3', 'A8', 'C1', 'E6', 'E1', 'E3', 'C4', 'D8', 'E4', 'D9', 'A9', 'C2', 'E7', 'E2', 'E9', 'B1', 'B2', 'B3', 'D6', 'D7', 'D4', 'D5', 'D2', 'D3', 'D1']
Para qual noh voce deseja ir?A5
O noh de destino estah no ponto: 12.5825,72.205
O caminho ate o ponto eh: ['D1', 'D2', 'C4', 'C3', 'C2', 'C1', 'A2', 'A3', 'A4', 'A5']
Qual o angulo(Graus) final desejado?0
```

Essa lista é baseada no arquivo (grafo), que tem todos esses pontos especificados com as suas devidas coordenadas. Logo, quando o usuário informa para qual nó deseja ir, um cálculo de rota é realizado baseando-se na ligação que os nós tem (concretizada pela matriz de correlação dos nós) então a melhor rota é encontrada e informada logo em seguida. O movimento entre os nós é estabelecido de forma que o ponto central de cada nó seja o objetivo. Assim, conforme o robô se movimenta e se orienta dentro do mapa, o programa constroi a grade de ocupação de cada nó, a qual será utilizada posteriormente durante a rota em cada nó.

Quando o robô chega até o destino, uma nova etapa se inicia durante a execução do programa, pois as grades de ocupação dos nós desejados já foram criados e em seguida o código requisita por um novo nó, dentro dos explorados, para que ele possa mover-se até esse novo destino. Logo que informado esse dado, a mapa do nó atual conterá a grade de ocupação do nó que o usuário se encontra e a posição do robô dentro desse mapa, informações essas que são atualizadas enquanto o robô se movimenta até chegar ao seu novo destino.

A criação da grade de ocupação ocorre dinamicamente a cada movimento realizado, por isso uma grade lentidão foi observada na primeira parte da execução. Ao sair de cada nó o programa informa a grade de ocupação do mesmo e o usuário não deve fechar a janela com a imagem, mas sim apertar o botão zero para continuar a execução do programa de maneira correta, caso contrário o programa para.

## Dados experimentais e Resultados

Como entrada, o dado deve ser fornecido deve ser um nó contido e acessível dentro das coordenadas do mapa do prédio disponível no launch stage disponibilizado em ([github.com/fcr2018](https://github.com/fcr2018)), vide imagem do mapa topológico em anexo. Para executar o programa, basta ter a cópia do arquivo '.py', inserir essa cópia na pasta src do projeto fcr2018 e executar o comando rosrun (ex: rosrun fcr2017 roteiro3.py).

Uma vez executado o programa mostra qual nó o robô está dentro do mapa, que normalmente é o C4, e pedirá pelo nó final desejado e o ângulo que o robô deve ficar ao final da execução, assim como na imagem detalhada na metodologia. Ao chegar no ponto destino o programa requisita mais uma vez por um novo nó que deseja alcançar, mas dessa vez apenas para andar e não com o propósito de criar uma grade de ocupação. Como saída o programa tem os comandos que envia para os tópicos de velocidade, uma mensagem de "Alinhado" ao final execução, os nós que o usuário pode alcançar, imagens da grade de ocupação criada e imagens da grade de ocupação juntas da posição do robô durante a execução. O comportamento do algoritmo pode ser observado no vídeo disponibilizado no link descrito nos comentários de entrega de trabalho, no ambiente Moodle.

## Conclusão

O programa mostrou-se útil na exploração de espaços não conhecidos, fazendo uso da estrutura de grafos para inicialmente movimentar-se entre um conjunto de nós estabelecidos por um mapa primário (mapa topológico) e do uso do sensor laser para estimar a posição de objetos no caminho. E além de explorar, ajuda o usuário a se localizar dentro desse espaço explorado, ilustrado por um símbolo '+' dentro da imagem de grade de ocupação mostrada.

A solução faz uso de grades matrizes para construir as grades de ocupação, e ainda faz isso enquanto se movimenta dentro do espaço estabelecido por um nó, e isso acarreta em um grande uso de processamento que pode deixar a execução bem lenta a depender da máquina. E apesar da inúmeras tentativas de tornar esse processo paralelo a toda movimentação, nenhuma delas teve muito sucesso, portanto optou-se pela execução sequencial e não otimizada do programa.

Durante os testes, foi possível observar que o programa se comportou bem e atendeu as necessidades entendidas pelo roteiro.