

# Fundamentos Computacionais de robótica

## Relatório - Trabalho 1

Gustavo Henrique Fernandes Carvalho  
14/0021671

### →Executando a simulação

Foi criado um *'launch file'* para simplificar a execução da simulação sendo necessário apenas o comando

```
$ roslaunch fcr2017 trabalho1.launch
```

para iniciar a simulação e carregar todos os módulos dependentes.

Obs: Antes de executar a simulação é necessário instalar as dependências descritas no arquivo README.md.

### →Entrada

A única entrada da simulação é a posição para onde o Pioneer deve ir, ela é feita pela janela do RViz utilizando a ferramenta *'2D Nav Goal'* como mostra a Figura 1. O mapa no fundo do RViz pode ser utilizado como uma referência para facilitar a seleção do destino porém não representa exatamente os dados lidos pelos sensores.

Ao selecionar o destino uma mensagem é mostrada no terminal confirmando a posição selecionada, como na Figura 2.

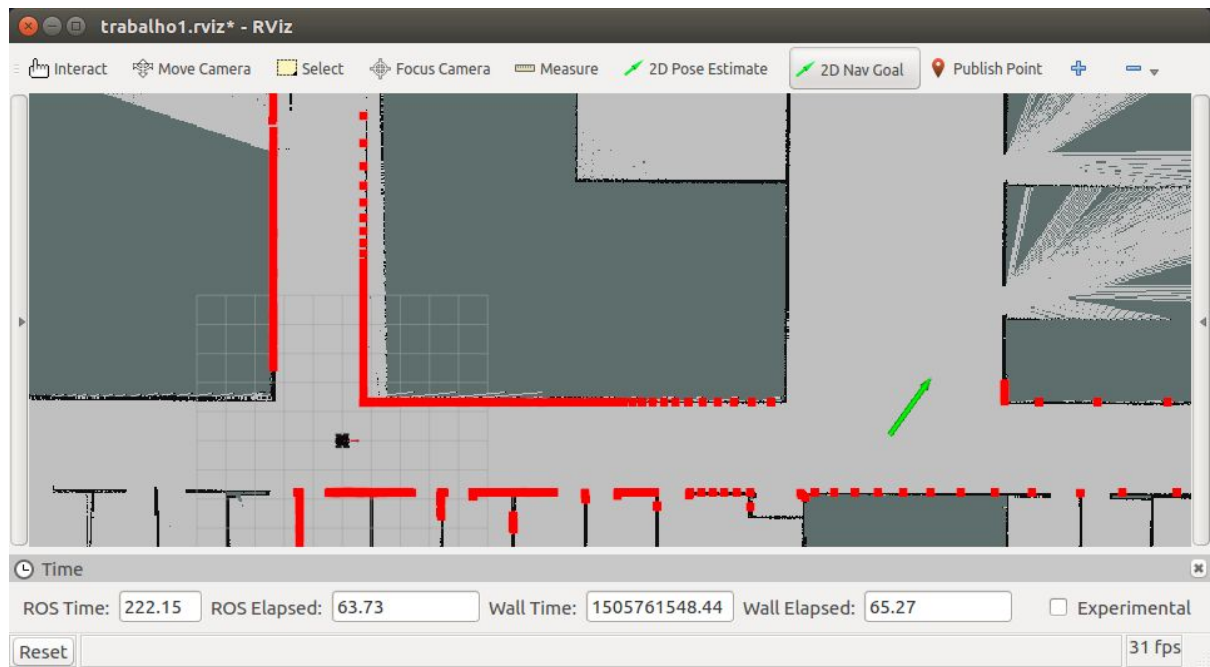


Figura 1 - Selecionando o destino

```
[ INFO] [1505761549.801269263, 223.514000000]: Indo para a posicao (18.81, 0.18, 0.94)
```

Figura 2 - Destino selecionado

## →Saída

A saída do programa é feita em 3 interfaces diferentes, o Gazebo, RViz e terminal.

No gazebo é mostrado a representação gráfica da simulação.

No RViz é mostrado a leitura do sensor laser, o trajeto feito pelo Pioneer e um mapa aproximado do CIC para facilitar a localização e seleção do destino.

No terminal é mostrado mensagem informando para onde o Pioneer está indo e se ele chegou no destino.

## →Algoritmos

O algoritmo desenvolvido possui dois modos de comportamento distintos, o seguinte pseudocódigo apresenta uma visão geral do comportamento nesses dois modos.

```
enquanto não chegou_no_objetivo()
    modo == modoIndoParaObjetivo
        se objeto_no_caminho()
            calcula_linha_Bug2()
            modo = modoBug2
        senão
            vai_em_direcao_do_objetivo()
        fim se
    senão modo == modoBug2
        se em_cima_da_linha_Bug2() E esta_mais_proximo_do_objetivo()
            modo = modoIndoParaObjetivo
        senão
            contorna_objeto()
    fim se
fim enquanto
```

O algoritmo para ir em direção ao objetivo é simples, o Pioneer rotaciona até ficar próximo da direção do objetivo e segue em direção ao objetivo fazendo pequenos ajustes no ângulo caso necessário até chegar no objetivo ou encontrar um obstáculo no caminho.

Caso um obstáculo seja encontrado o comportamento do Pioneer muda para o modo Bug 2, isso é, ele armazena a informação da reta que passa pelo posição dele e o objetivo e começa a contornar o obstáculo mantendo uma distância constante até retornar para a linha calculada anteriormente, em um ponto mais próximo do objetivo que o ponto do início do algoritmo Bug 2.

Para manter uma distância constante enquanto contorna o obstáculo é utilizado a leitura lateral do sensor laser para calcular, pela Equação 1, o ângulo  $\Theta$ , entre a direção que o Pioneer está se movendo a lateral do obstáculo.

$$\theta_X = \arctan\left(\frac{D_M - D_X}{\sin(30^\circ) * D_X}\right)$$

Equação 1

Onde  $D_M$  é a distância medida no ângulo  $\pm 90^\circ$ ,  $D_{Frente}$  é a distância no ângulo  $\pm 120^\circ$  e  $D_{traz}$  é a distância no ângulo  $\pm 60^\circ$ . Então ajustando ângulo  $\Theta$  de forma a deixá-lo próximo de zero permite que o Pioneer mantenha uma distância constante do obstáculo.

## → Estrutura do package

O pacote fcr2017 possui muitos arquivos padrões ou não utilizados para a simulação e não serão descritos nessa seção, os arquivos novos ou modificados para o trabalho 1 são:

- fcr2017/package.xml

Descrição do pacote e suas dependências, foi adicionado uma nova dependência nesse arquivo.

- fcr2017/launch/trabalho1.launch

Configuração para iniciar a simulação carregando todos os módulos necessários.

- fcr2017/launch/trabalho1.rviz

Configuração do RViz para mostrar o mapa e o trajeto completo feito pelo Pioneer.

- fcr2017/src/140021671\_Trabalho1.cpp

Código fonte do nó responsável por receber as informações dos sensores do Pioneer e enviar os comandos de velocidade de forma que ele chegue em segurança até o objetivo.

## → Referências

- ◆ <http://wiki.ros.org/>
- ◆ [http://www.cs.cmu.edu/~./motionplanning/lecture/Chap2-Bug-Alg\\_howie.pdf](http://www.cs.cmu.edu/~./motionplanning/lecture/Chap2-Bug-Alg_howie.pdf)
- ◆ [http://spacecraft.ssl.umd.edu/academics/788XF14/788XF14L14/788XF14L14.pathbu\\_gsmapsx.pdf](http://spacecraft.ssl.umd.edu/academics/788XF14/788XF14L14/788XF14L14.pathbu_gsmapsx.pdf)
- ◆ <http://user.ceng.metu.edu.tr/~saranli/courses/ceng786/lectures/Lecture2-BugAlgorithms.pdf>
- ◆ Maja J. Mataric, The robotics Primer

O vídeo da simulação está em [https://www.youtube.com/watch?v=BDsXJKN\\_wEk](https://www.youtube.com/watch?v=BDsXJKN_wEk).