

# DOCUMENTAÇÃO TRABALHO 1

## ROBÓTICA MÓVEL

Sistemas da Informação

Clarissa Lima Loures

2017023340

Ciências da Computação

Taynara Lorraine Marcelino da Cruz

2017015177

## 1. Introdução

Robótica móvel leva em conta a capacidade de mobilidade e autonomia dos robôs. Desta forma, um **robô móvel** deve ser capaz de se movimentar, perceber e reagir ao ambiente com o objetivo de atingir uma meta.

Este trabalho teve como objetivo o desenvolvimento de algoritmos de planejamento de caminho e à análise destes em diferentes mapas. Foram abordados os algoritmos BUG 2 e Campos Potenciais. Os códigos foram desenvolvidos no Robot Operating System (ROS), utilizando a linguagem de código Python.

## 2. Implementação

### 2.1. Algoritmo Bug 2

Os algoritmos BUG assumem uma visão local do ambiente e uma posição no espaço como objetivo. Há dois passos básicos para a realização da tarefa:

1. Ir em direção ao **goal** em linha reta;
2. Caso encontre um obstáculo, contorná-lo.

Para a realização deste trabalho foi necessário implementar o algoritmo BUG 2. Tal algoritmo busca realizar uma trajetória em linha reta até o goal (**m\_linha**). Ao encontrar um obstáculo em sua frente o robô irá contornar este até achar novamente à **m\_linha**.

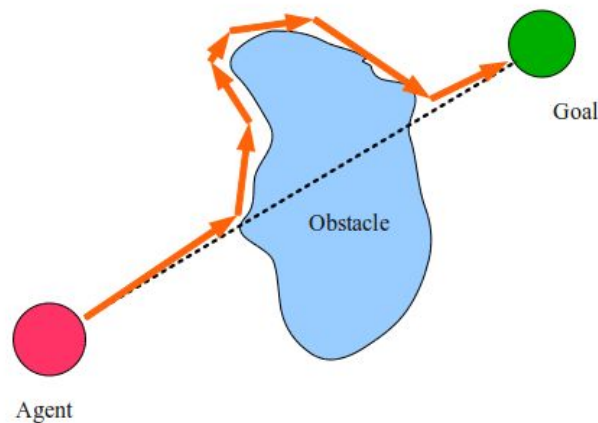


Figura 1: Modelagem do problema utilizando Bug 2.

Para este algoritmo foi utilizado um robô holonômico. Este tipo de robô não possui restrições em relação à sua movimentação.

## 2.2. Algoritmo Campos Potenciais

O algoritmo Campos Potenciais consistem em realizar a navegação levando em conta forças de atração e repulsão. O robô e o goal possuem sinais opostos, enquanto os obstáculos possuem o mesmo sinal do robô. Desta maneira, o robô é atraído pelo goal e repelido pelos obstáculos, o que torna este algoritmo segundo em relação a não atingir obstáculos.

No algoritmo foi utilizado o controlador Desai et al. 1998 para a realização da transformação entre as velocidades desejadas em cada coordenada e as velocidades linear e angular. Tal fórmula é dada por:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}$$

Um problema notável deste algoritmo é a existência de mínimos locais, locais onde o somatório das forças é nulo. Neste momento, o robô não consegue se mover para a direção do goal, não atingindo sua meta.

---

### Algorithm Campos Potenciais: Função Atração

---

```
def Att_Force(Erro_em_x, Erro_em_y):  
    #Calcular a posição do goal relativa ao referencial do robô  
    distância = math.sqrt(Erro_em_x ** 2 + Erro_em_y ** 2)  
    theta_velocidade = atan2(Erro_em_y, Erro_em_x)  
    K = K_atração * distância;  
    Força_em_x = K*cos(theta_velocidade)  
    Força_em_y = K*sin(theta_velocidade)  
    return Força_em_x, Força_em_y
```

---

---

### Algorithm Campos Potenciais: Função Repulsão

---

```
def Rep_Force():  
    Força_em_x = 0  
    Força_em_y = 0  
    #Para cada obstáculo dentro do range do sensor  
    for i in range(len(laserMsg.range)):  
        theta_velocidade = (laserMsg.angle_increment * i)  
        m = laserMsg.ranges[i]  
        if m < p_0:  
            force_x += Krep * ((1/m) - (1/p_0)) * (1/(m*m))*(cos(theta_velocidade +  
theta)/m)  
            force_y += Krep * ((1/m) - (1/p_0)) * (1/(m*m))*(sin(theta_velocidade +  
theta)/m)  
    return  
Constante  
  
Valor Força_em_x, Força_em_Y
```

---

Após à realização de diversos testes, foi descoberto que, dado o sistema utilizado e à maneira como foi implementado o algoritmo, os valores das variáveis que geram os melhores resultados são:

Constante	Valor
Constante de ganho em X	1
Constante de ganho em Theta	1
Constante de atração	10
Constante de repulsão	3
P_0	1

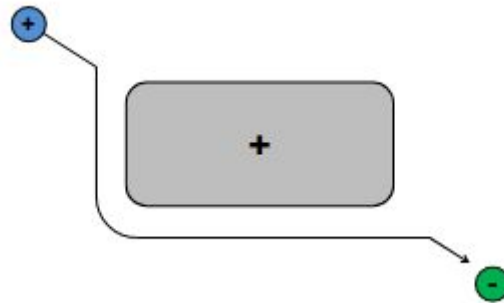


Figura 2: Modelagem do problema utilizando Campos Potenciais

Para este algoritmo foi utilizado um robô diferencial. Este tipo de robô possui restrições com relação à sua movimentação.

### 3. Testes

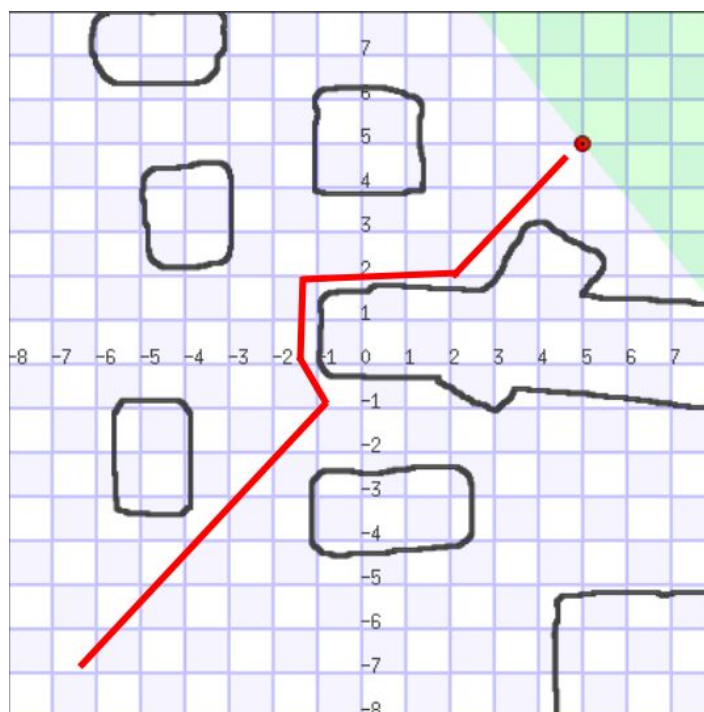
Os testes realizados para os dois algoritmos ocorreram nos mesmos mapas. Foram utilizados 4 mapas distintos com tamanhos e obstáculos diferentes, para uma melhor análise dos algoritmos.

#### 3.1. Análise Bug 2

Durante a primeira execução dos testes, foi possível analisar que as constantes escolhidas para auxiliar no projeto estavam muito grandes. Desta forma o robô não conseguia executar de forma correta pois não encontrava caminhos estreitos.

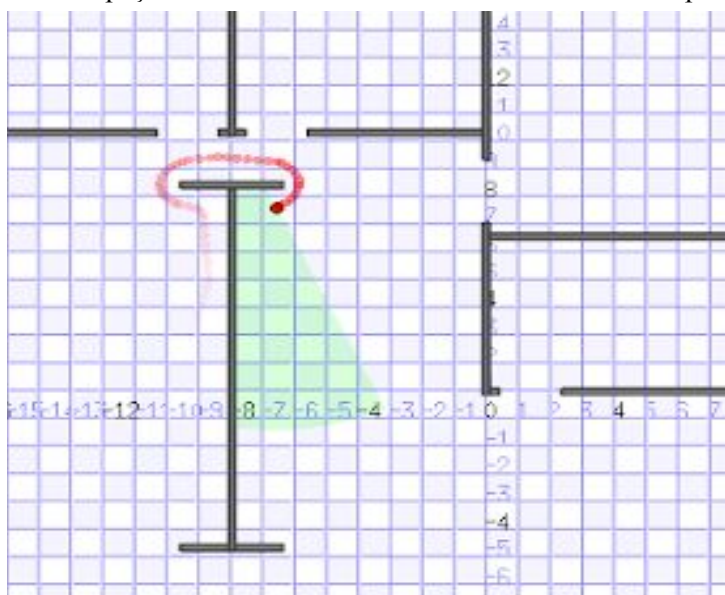
- Teste 1:

O primeiro teste foi realizado no mapa Cave. Neste ambiente o robô sai da posição  $X = -7$  e  $Y = -7$  com direção ao goal localizado em  $X = 5$  e  $Y = 5$ . O movimento do robô neste cenário foi satisfatório, o robô conseguiu identificar o obstáculo e contornar este até encontrar à ***m\_linha*** novamente.



- Teste2:

O segundo teste foi realizado no mapa Autolab. Neste ambiente o robô sai da posição  $X = -16$  e  $Y = -6$  com direção ao goal localizado em  $X = -5$  e  $Y = 2$ . O robô conseguiu identificar em um espaço reduzido com muitos obstáculos o caminho possível.



- Teste 3:

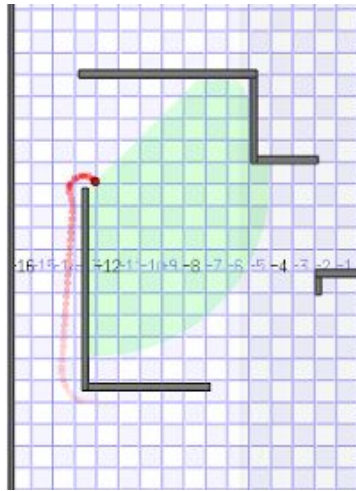
O terceiro teste foi realizado no mapa Parede. Neste ambiente o robô sai da posição  $X = -7$  e  $Y = -7$  com direção ao goal localizado em  $X = -15$  e  $Y = 11$ . Os passos do robô podem ser observados pela sequência abaixo:



Passo 1



Passo 2

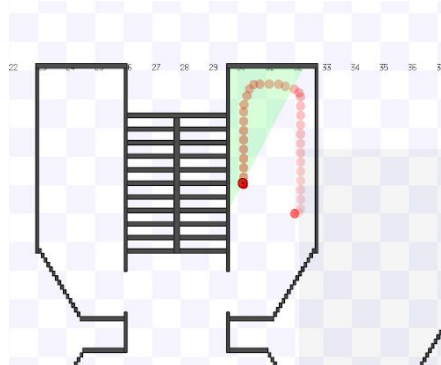


Passo 3

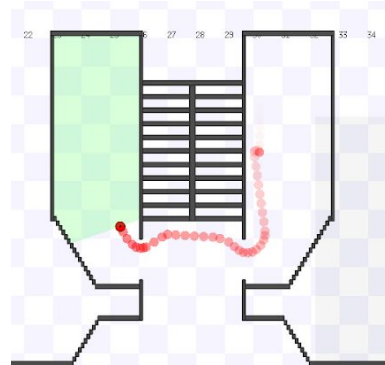


Passo 4

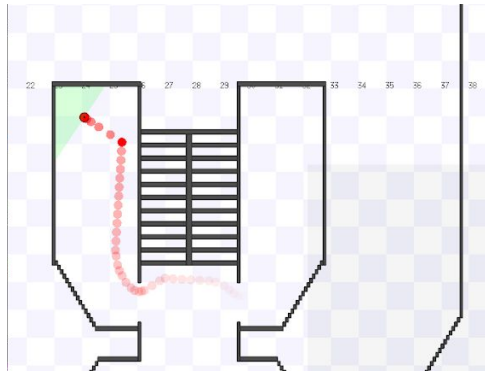
- Teste 4:  
O quarto teste foi realizado no mapa ICEx. Neste ambiente o robô sai da posição  $X=32$  e  $Y=-5$  com direção ao goal localizado em  $X=24$  e  $Y=-2$ .



Passo 1



Passo 2



Passo 3

### 3.2. Análise Campos Potenciais

Os testes realizados para o Algoritmo de Campos Potenciais demonstraram que a escolha inicial para as constantes não era satisfatória. Embora no mapa Cave o algoritmo conseguisse chegar até o destino definido em certos casos, ao ser testado o mapa Autolab o robô não identificava paredes o que levou à sua colisão com estas.

Após reajustes de valores foi possível encontrar números bons o suficiente para que esta colisão não ocorresse. Porém, vale ressaltar que a grande maioria dos casos levou o robô à mínimos locais, o que impediu que o robô chegasse até o ponto desejado.

- Teste 1:  
O primeiro teste foi realizado no mapa Cave. Neste ambiente o robô sai da posição  $X=5$  e  $Y=5$  com direção ao goal localizado em  $X=-7$  e  $Y=-7$ . É possível perceber neste teste a movimentação do robô em uma região cercada por obstáculos e como as forças de atração e repulsão modificam sua movimentação.

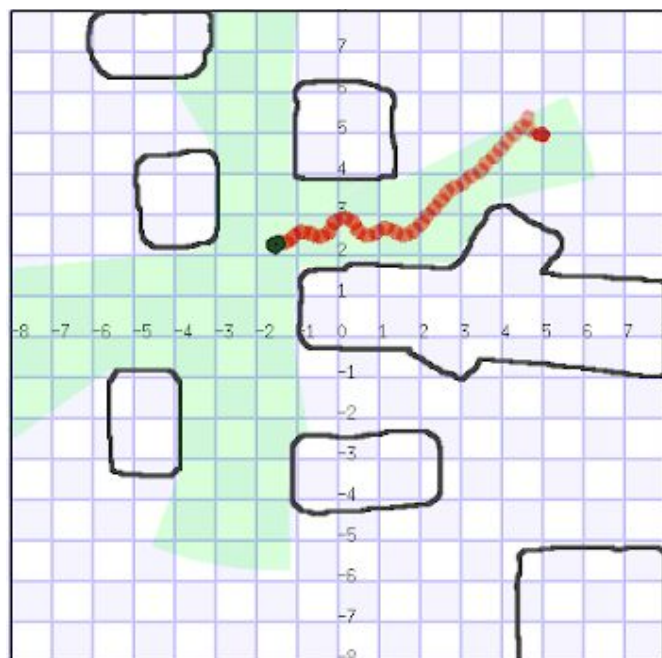


Figura 8: Robô realiza com sucesso a tarefa, mapa Cave

- Teste 2:

O segundo teste foi realizado no mapa Autolab, em um dos experimentos foi possível identificar a situação na qual o robô não consegue chegar em seu destino, devido à existência de um mínimo local. O goal é identificado na imagem como o local onde se encontra o círculo amarelo, a posição do robô no mapa é a região de mínimo local. Nessa região, as forças de repulsão e atração são idênticas, gerando uma região de Força de atuação com valor zero.

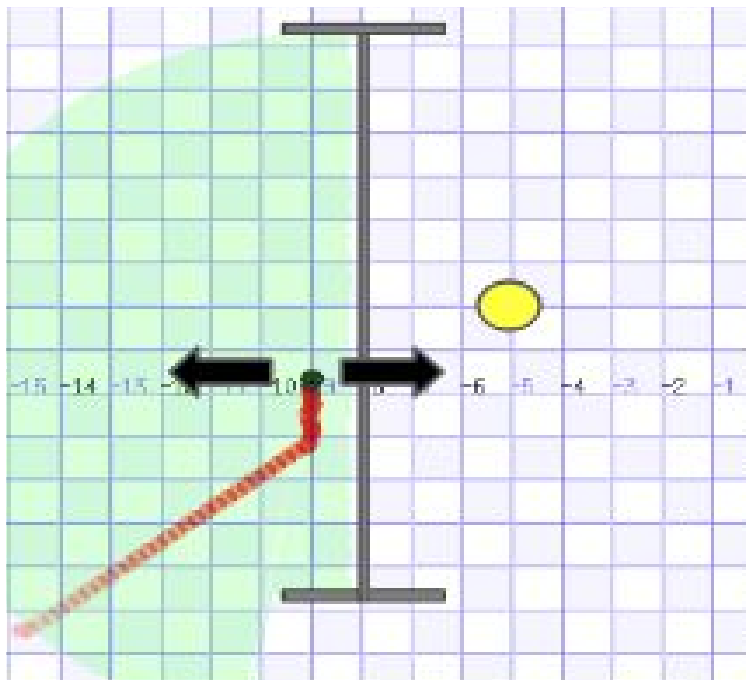


Figura 7: Robô encontrou mínimo local, não irá se movimentar

- Teste 3:

O terceiro teste foi realizado no mapa Paredes. Neste cenário, o robô sai do ponto inicial  $X = -13$  e  $Y = 14$  e segue em direção ao goal localizado em  $X = 5$  e  $Y = 5$ . Neste mapa, por não terem obstáculos próximos entre si, é possível perceber uma movimentação com menos curvaturas. Desta maneira, há pouca atuação da força de repulsão neste trajeto, podendo ser percebido o aumento desta apenas no momento em que o robô se aproxima da parede próxima ao ponto  $X = -7$  e  $Y = 9$ .



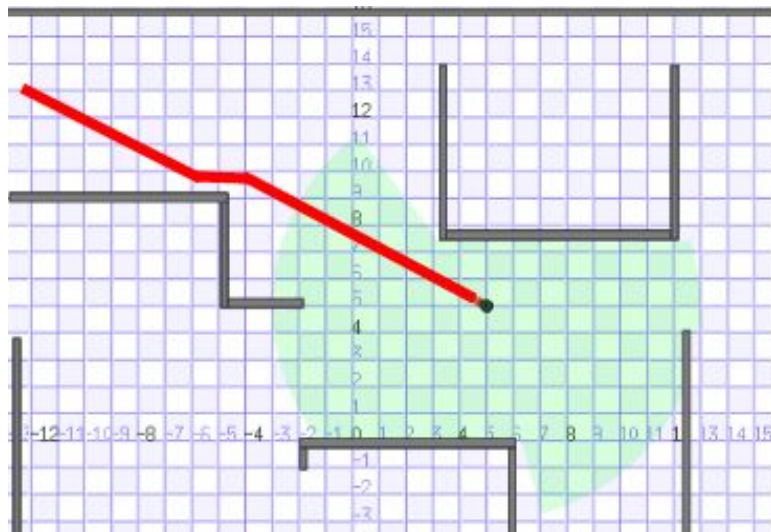


Figura 9: Robô realiza com sucesso a tarefa, mapa Paredes

- Teste 4:

O quarto teste foi realizado no mapa ICEx. Neste cenário, o robô sai do ponto inicial  $X = -50$  e  $Y = 47$  e segue em direção ao goal localizado em  $X = 65$  e  $Y = 50$ . Neste mapa, podemos observar o robô passando por um obstáculo. Por não terem obstáculos próximos entre si, é possível perceber uma movimentação com menos curvaturas. Desta maneira, há pouca atuação da força de repulsão neste trajeto, podendo ser percebido o aumento desta apenas no momento em que o robô se aproxima da parede próxima ao ponto  $X = -7$  e  $Y = 9$ .

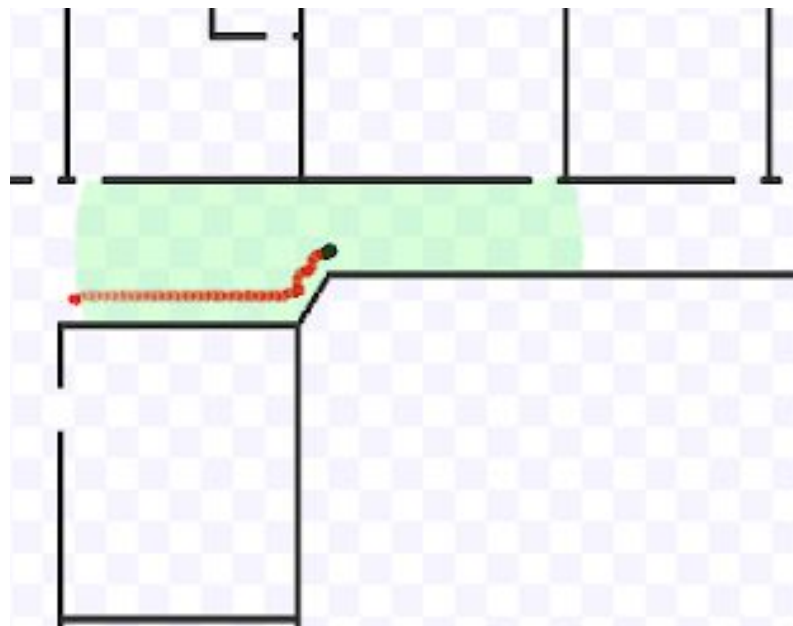


Figura 10: Robô desviando de obstáculo, mapa ICEx

## 4. Conclusão

Neste trabalho, foi desenvolvido dois algoritmos e à análise destes em ambientes mapas. Foi possível perceber que pequenos ajustes certas variáveis podem causar grandes mudanças na execução do algoritmo, por este motivo, foi necessário à realização de diversos testes até que os programas demonstram resultado satisfatório.

## 5. Referências

*Youtube - Bug2 - Path Planning Algorithm Explanation*

Canal olzhas

*Introduction to Robotics: Mechanics and Control*

John J. Craig

Prentice Hall, 2004, 3rd Edition