



Universidade Federal de Itajubá

*Campus Itabira*

**ECAi21.2 – Laboratório de Robótica Móvel**

4 de dezembro de 2022

Docente: André Chaves Magalhães

Discente:

– Gabriel Toffanetto França da Rocha – 2019003646

## Exercício 5 – TangetBug

---

### Sumário

1	Introdução	2
2	Em um cenário sem obstáculos – <i>m-line</i>	2
3	Wall Follow	2
4	Detecção e Seleção de Descontinuidades	5
5	Troca de Contexto	5

---

# 1 Introdução

O TangetBug é um método reativo de navegação de robôs, onde com base do conhecimento da posição atual, da posição de destino e da leitura do ambiente, por um LiDAR por exemplo, é possível chegar ao destino tangenciando obstáculos. Para a realização dessa estratégia, é necessário a implementação de algoritmos que executam passos importantes, como a lógica de Wall Follow, de detecção de descontinuidades, de seleção da descontinuidade com menor custo e de troca de contexto. Para isso, cada um desses módulos foram desenvolvidos de forma mais independente possível, como mostrado abaixo.

## 2 Em um cenário sem obstáculos – *m-line*

Caso não existam obstáculos, o robô calcula o erro de ângulo entre sua direção atual e a direção do alvo, obtendo assim a velocidade angular. A velocidade linear é proporcional à distância entre o robô e o alvo e ao erro angular. Para armazenar a posição do robô e do alvo, foi criada uma **struc** Ponto, que armazena as coordenadas  $x$ ,  $y$  e  $z$ . Foram cradas **structs** também para armanezar a direção do robô em quatérnio e ângulos de Euler, assim como implementada uma função para essa conversão.

## 3 Wall Follow

O problema do seguidor de parede é utilizado para a implementação do Tangent Bug para possibilitar o contorno dos obstáculo até que o robô possa novamente passar a seguir a *m-line* em direção ao ponto de destino ( $q_{goal}$ ). Dessa forma o algoritmo de seguir parede faz com que o robô consiga manter uma distância constante,  $d_w$ , da parede e com trajetória tangente à mesma. Para isso, para percepção da parede é utilizado o sensor LiDAR, que como mostrado na Figura 1, realiza a medição de distância por meio de vários raios *laser* com angulação conhecida.

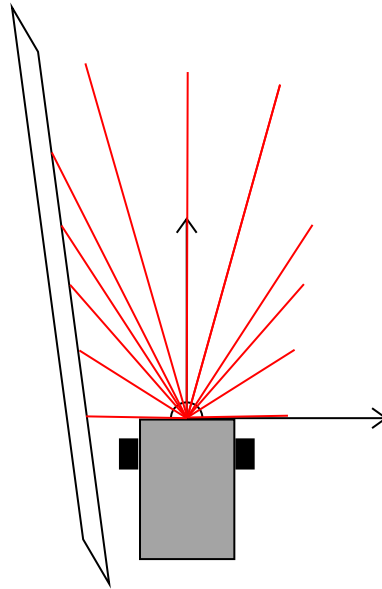


Figura 1: Representação do problema do seguidor de parede.

Uma forma de modelar o problema é mostrada na Figura 2, onde pegando dois raios aleatórios do LiDAR,  $\vec{V}_1$  e  $\vec{V}_2$ , que possuem ângulos  $\theta_1$  e  $\theta_2$ , respectivamente. A origem para esses ângulos é dada no sentido frontal do robô, sendo o centro do LiDAR o 0 rad. Uma vez posto esses dois vetores, o vetor resultante  $\vec{V}_r$  é mostrado em azul, sendo paralelo à parede que pretende-se seguir. Ao reduzir esse vetor à origem do LiDAR, têm-se o vetor  $\vec{V}_{r0}$ , que possui um ângulo  $\theta_w$ . Esse ângulo  $\theta_w$  pode ser utilizado como ângulo de erro entre a orientação do robô e a orientação da parede, sendo que quando o mesmo for zero, o robô estará se movendo em paralelo com a parede.

Porém, é necessário também controlar a distância que o robô se encontra da parede, dada por  $d_w$ .  $d_w$  será o módulo do feixe do LiDAR com o menor módulo, representando o ponto mais próximo do robô em relação à parede.

Em vista disso, considerando  $\vec{V}_1$  como o ponto mais próximo do robô à parede e  $\vec{V}_2$  o feixe consecutivo, pode-se obter o vetor resultante e por fim, a lei de controle para resolução do problema de seguidor de parede.

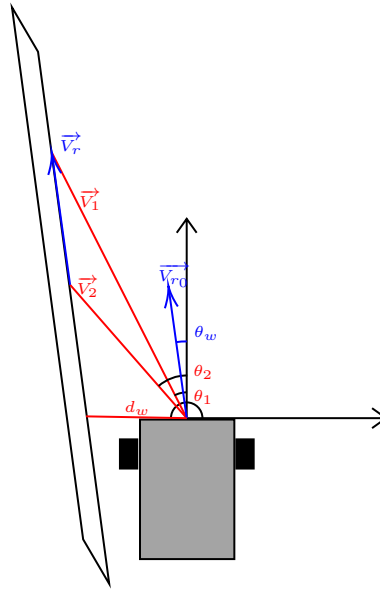


Figura 2: Modelagem do problema do seguidor de parede.

$\vec{V}_r$  é dado por:

$$\vec{V}_r = \vec{V}_2 - \vec{V}_1 \quad (1)$$

$\vec{V}_{r0}$  é dado por:

$$\vec{V}_{r0} = \vec{V}_r - \vec{V}_1 \quad (2)$$

$\theta_w$  é dado por:

$$\theta_w = \arccos \frac{\langle \hat{i}, \vec{V}_{r0} \rangle}{||\vec{V}_{r0}||} \quad (3)$$

O erro da distância até a parede,  $e_{dw}$  é dada por:

$$e_{dw} = d_{min} - d_w \quad (4)$$

E dessa forma as leis de controle são dadas por:

$$v = K_{pL}V_{max} - K_{Ld}|\theta_w| - K_{Lw}|e_{dw}| \quad (5)$$

$$\omega = K_{Ad}|\theta_w| + |e_{dw}|K_{Aw} \quad (6)$$

O TangetBug é um algoritmo completo, ou seja, ele tem a capacidade de informar se não houver solução. A detecção de não existência de solução é feita no contexto de seguidor de parede, onde o robô irá seguir o obstáculo, e se ele der uma volta completa no obstáculo sem trocar de contexto, quer dizer que o alvo está dentro do obstáculo, e que não há como chegar até ele. Dessa forma, ao se detectar que o robô passou pelo ponto onde ele começou a seguir o obstáculo, o mesmo tem seu movimento finalizado e é avisado que não existe solução.

## 4 Detecção e Seleção de Descontinuidades

Para realização da lógica, foi criado uma **struct** para armazenar as informações sobre as descontinuidades. Essa estrutura de dados armazena o ponto onde a descontinuidade foi encontrada, o custo dessa descontinuidade e se o ponto representa o início ou fim de um obstáculo. A função de custo da descontinuidade é dada pela distância euclidiana do robô até a descontinuidade somada a distância euclidiana da descontinuidade até o destino.

As descontinuidades obtidas com o processamento dos dados do LiDAR são armazenadas em um **vector**, que é ordenado por meio de uma *lambda function*, para obter a descontinuidade com o menor custo, a qual é escolhida e o robô navega até ela para iniciar o processo de Wall Follow.

A descontinuidade é detectada quando ocorre uma variação da distância medida pelo *laser* maior que 1 metro. Para obter o ponto no plano que ocorre a descontinuidade, o raio do *laser* que ainda detecta o obstáculo, podendo ser o atual ou o anterior, é decomposto em  $x$  e  $y$ , em relação ao sistema de coordenadas do *laser*, utilizando a distância medida pelo *laser* e o ângulo do feixe. É feita uma aproximação considerando que o sistema de coordenadas do *laser* é sobreposto ao sistema de coordenadas do robô. Uma vez feita a decomposição e obtido o ponto em relação ao sistema de coordenadas do *laser*, é feita a translação para o sistema de coordenadas gerais.

Uma vez obtido a posição da descontinuidade, é calculada a função de custo, e aplicado um *offset* à posição da descontinuidade, afastando-a do obstáculo por uma distância segura, de forma que o robô possa alcançar esse ponto sem que haja colisão.

## 5 Troca de Contexto

A troca de contexto entre o modo de seguir a *m-line* para o modo de ir até uma descontinuidade ocorre quando é detectado um obstáculo à frente do robô, sendo necessário que o mesmo desvie para não colidir.

A partir disso, o robô passa a navegar até o ponto da descontinuidade com *offset*, e chegando ao mesmo, realiza a troca de contexto para o modo de Wall Follower.

Por fim, a troca de contexto entre o modo de Wall Follower e de seguir o obstáculo/descontinuidade é dado por meio da obtenção de uma visão livre de obstáculos para o alvo, ou a obtenção de uma descontinuidade com uma função de custo menor que a que foi usada para seguir o obstáculo atual.