

A Simple Local Path Planning Algorithm for Autonomous Mobile Robots

Fábio Demo da Rosa

Universidade Federal de Santa Maria
Pós-Graduação em Ciência da Computação
Disciplina de Robótica Móvel

faberdemo@gmail.com

24 de novembro de 2023

- ① Introdução
- ② Abordagens de Planejamento de Caminho
- ③ Planejamento de Trajetória (Local e Global)
- ④ Algoritmos PointBug
- ⑤ Simulação
- ⑥ Conclusões

- O planejamento de trajetória/rota/caminho é um elemento crucial para robôs móveis.
- É necessário determinar rotas, para passar ou chegar até pontos específicos do ambiente.
- As abordagens para planejar a trajetória são de acordo com o ambiente, o tipo de sensor, as capacidades do robô, entre outros.
 - tais abordagens estão gradualmente buscando um melhor desempenho em termos de tempo, distância, custo e complexidade (BUNIYAMIN et al., 2011).

Abordagens de Planejamento de Caminho I

- Deseja-se encontrar caminhos adequados para um robô com geometria específica.
- Objetiva-se alcançar uma posição e orientação final a partir de uma inicial.
- O problema de navegação do robô móvel pode ser dividido em três subtarefas, de acordo com (BUNIYAMIN et al., 2011):
 - **Mapeamento e Modelagem do Ambiente:** o robô deve ser capaz de construir um mapa do ambiente e modelar o ambiente;
 - **Planejamento de Caminho:** o robô deve ser capaz de planejar um caminho para alcançar o objetivo;
 - **Travessia de Caminho:** o robô deve ser capaz de seguir o caminho planejado e evitar colisões com obstáculos.
- Tipos de ambiente: estático (sem objetos móveis) e dinâmico (com objetos móveis).
- Abordagens: planejamento local e global.

Planejamento de Trajetória (Local e Global)

- **Planejamento de Trajetória Global:**

- Exige informação prévia do ambiente.
- Planejamento completo do trajeto antes do movimento do robô.
- Abordagens incluem Grafos de Visibilidade, Diagramas de Voronoi, Decomposição Celular, e métodos modernos como Algoritmo Genético, Redes Neurais e Otimização de Colônia de Formigas.

- **Planejamento de Trajetória Local:**

- Essencial para robôs em ambientes dinâmicos, foca na evasão de obstáculos usando sensores.
- Robôs seguem a rota mais direta, alterando-a ao encontrar obstáculos.
- **Algoritmo PointBug:** Evita o perímetro dos obstáculos, diferenciando-se dos algoritmos Bug tradicionais.
 - **Implementação do PointBug:** Utiliza sensores de curto alcance e sistemas de navegação para decisões de trajetória.

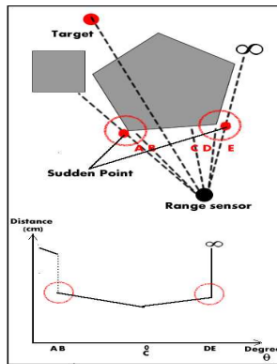
Algoritmos PointBug I

- O PointBug é um algoritmo de navegação recentemente desenvolvido, para robôs em ambientes planares desconhecidos com obstáculos estacionários.
- Utiliza um sensor de alcance para detectar mudanças súbitas na distância até o obstáculo mais próximo (detecção de pontos).
 - Mudanças súbitas são identificadas quando a distância varia significativamente em um curto intervalo de tempo.
- Eficaz para resolver o problema de mínimos locais em ambientes desconhecidos, identificando pontos súbitos de maneira confiável.
- **Determinação do Próximo Ponto:** O próximo ponto de movimento é determinado pela saída do sensor de alcance, baseando-se na variação (Δd) da distância detectada.
- **Funcionamento do Algoritmo:** O robô inicialmente se direciona ao ponto-alvo, rotacionando para localizar um ponto súbito e mover-se em sua direção. Esse processo se repete até o robô atingir o ponto-alvo.

Algoritmos PointBug II

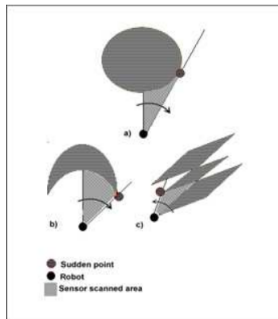
```
1 While Not Target
2   If robot rotation <= 360
3     Robot rotates right of left according to position of dmin
4   If sudden point
5     If 180 degree rotation
6       Ignore reading /* to avoid robot return to previous point */
7     Else
8       Get distance from current sudden point to next sudden point
9       Get angle of robot rotation
10      Move to new point according to distance and rotation angle
11      Record New dmin value
12      Reset rotation
13    End if
14  End if
15 Else
16   Robot Stop /* No sudden point and exit loop */
17 End if
18 While end
19 Robot Stop /* Robot successfully reaches target */
```

Algoritmos PointBug III



Fonte: (BUNIYAMIN et al., 2011)

Figura 1: O sensor de alcance está detectando um obstáculo da esquerda para a direita e da direita para a esquerda.



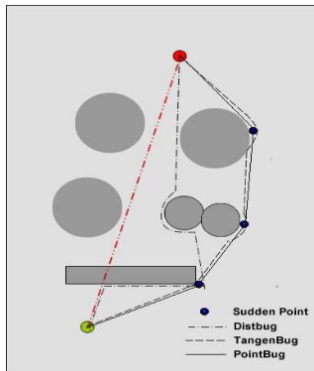
Fonte: (BUNİYAMIN et al., 2011)

Figura 2: Trajetória gerada pelo PointBug para resolver o problema de mínimos locais.

Simulação I

- A simulação do algoritmo de navegação ponto a ponto (PointBug) foi realizada utilizando ActionScript 2.0 no Adobe Flash CS3.
- Foram simulados três tipos de ambientes: um ambiente livre, um ambiente baseado em labirinto e um ambiente semelhante a um escritório.

Simulação II



Fonte: (BUNİYAMIN et al., 2011)

Figura 3: Trajetória gerada usando os algoritmos Distbug, TangenBug e PointBug em ambiente livre.

- De acordo com o artigo (BUNİYAMIN et al., 2011), tratou-se sobre:
 - Introdução do Algoritmo de Planejamento de Trajetória Baseado em Sensor Ponto a Ponto, com mínima necessidade de informações prévias.
 - Provou-se a eficiência do PointBug em ambientes dinâmicos, utilizando informações em tempo real de sensores de alcance.
 - Desempenho é diretamente influenciado pelo número de pontos súbitos detectados.
 - Menos pontos indicam maior eficiência.
 - Efetividade dependente diretamente da forma dos obstáculos.
 - Obstáculos circulares geram menos pontos súbitos.

BUNIYAMIN, Norlida et al. A simple local path planning algorithm for autonomous mobile robots. **International journal of systems applications, Engineering & development**, v. 5, n. 2, p. 151–159, 2011.

A Simple Local Path Planning Algorithm for Autonomous Mobile Robots

Fábio Demo da Rosa

Universidade Federal de Santa Maria
Pós-Graduação em Ciência da Computação
Disciplina de Robótica Móvel

faberdemo@gmail.com

24 de novembro de 2023