

UNIVERSIDADE FEDERAL DE MINAS GERAIS
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA ELÉTRICA



PLANEJAMENTO DE MOVIMENTO DE ROBÔS -
RELATÓRIO TP1

Leandro Henrique Vidigal Sousa
Túlio Pires Oliveira

7 de maio de 2024

Relatório do Trabalho Prático 1 da Disciplina de Planejamento de Movimento de Robôs

Algoritmos de Planejamento
de Movimento.

Esse trabalho aborda sobre a utilização de alguns algoritmos utilizados para o planejamento de movimento de robôs, dentre eles estão o Tangent Bug, Funções de Potencial com resolução de problemas de mínimos locais e utilização de equações paramétricas como trajetórias para locomoção de robôs.

7 de maio de 2024

1 Introdução

O planejamento de movimento de robôs móveis é uma área vital da robótica que se ocupa com o desenvolvimento de estratégias para permitir que robôs autônomos se desloquem de um ponto a outro em um ambiente, evitando obstáculos de maneira eficaz. Esta capacidade é crucial em muitas aplicações contemporâneas, desde veículos autônomos em ambientes urbanos até robôs exploradores em missões espaciais ou submarinas. O objetivo fundamental é desenvolver algoritmos que não apenas tracem uma rota viável, mas também garantam a segurança, eficiência e adaptabilidade do robô em ambientes dinâmicos e potencialmente hostis.

O campo do planejamento de movimento tem sido intensamente explorado e possui uma vasta literatura. Entre os trabalhos fundamentais, destacam-se as contribuições de Choset et al. [1], que fornecem uma compreensão abrangente das técnicas de planejamento de caminho, incluindo a decomposição de espaços e o algoritmo de busca A*. Outra referência significativa é o trabalho de LaValle [2], que sistematiza os métodos de planejamento de movimento, com especial atenção para os aspectos computacionais e algorítmicos.

As soluções propostas para resolver os desafios do planejamento de movimento de robôs podem ser amplamente categorizadas em algumas abordagens principais. O método do Tangent Bug, por exemplo, explora a detecção sensorial local do robô para navegar em torno de obstáculos, adaptando-se a mudanças imprevistas no ambiente. Este método é especialmente útil em ambientes onde os mapas precisos não estão disponíveis ou são incompletos. Em contrapartida, estratégias que envolvem seguir trajetórias formadas por equações paramétricas são frequentemente empregadas em cenários onde a trajetória precisa ser suave e contínua, como em aplicações industriais de robôs manipuladores.

Além disso, o uso de campos potenciais representa outra abordagem robusta, onde forças virtuais são aplicadas ao robô, atraindo-o para o objetivo enquanto repulsam-no dos obstáculos. No entanto, esta técnica pode sofrer com o problema de mínimos locais, onde um robô fica 'preso' em uma configuração sem alcançar o objetivo final. Métodos para resolver esses mínimos locais, como o aumento da dinâmica de repulsão ou a introdução de técnicas de escape, são essenciais para a eficácia dos campos potenciais.

Concluindo, o planejamento de movimento de robôs móveis é um campo de estudo que continua evoluindo rapidamente, impulsionado por avanços tecnológicos e crescente demanda por automação. As técnicas abordadas neste trabalho, incluindo o Tangent Bug, o seguimento de trajetórias formadas por equações paramétricas, os campos potenciais e as estratégias para lidar com mínimos locais, representam

soluções sofisticadas para os desafios enfrentados na navegação autônoma de robôs. À medida que a pesquisa avança, espera-se que novas e mais eficientes metodologias emergirão, ampliando ainda mais as capacidades e a aplicabilidade dos robôs autônomos em diversos domínios.

2 Metodologia e Simulações

O trabalho foi desenvolvido utilizando plataformas de desenvolvimento de robótica como o ROS e de visualização: Gazebo e Coppelia. Para simulações com o Gazebo, foi utilizado o Ubuntu 20.04 e o ROS Noetic com implementações em código python.

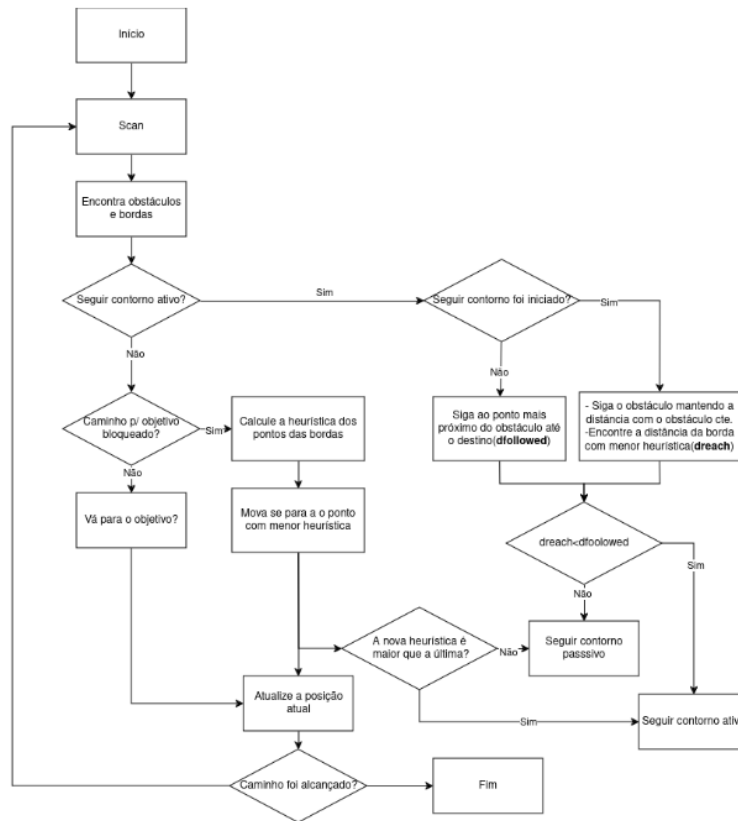
Foi utilizado um robô não-holonômico, modelo diferencial, com controlador Feedback Linearization para alguns problemas e para outros o controlador utilizado foi baseado em métodos de estabilidade, conforme será descrito nesse trabalho.

2.1 Tangent Bug

O algoritmo Tangent Bug é um método de planejamento de trajetórias que utiliza sensores de alcance finito para encontrar caminhos livres de colisões do ponto de partida até um ponto de destino conhecido. Este algoritmo é similar aos outros algoritmos Bug, como Bug1 e Bug2, mas se destaca por sua abordagem direta e pela completude em encontrar uma solução, se existir, dentro de ambientes definidos.

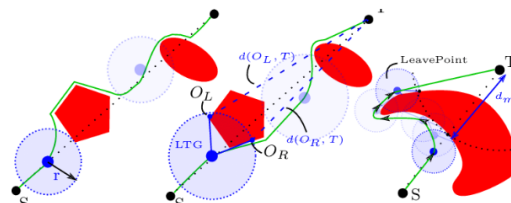
- **Movimento Direto ao Objetivo:** Se não houver obstáculos bloqueando o caminho, o robô se move diretamente em direção ao objetivo. Esse comportamento é representado por caixas azuis nas figuras.
- **Detecção de Obstáculos e Bordas:** Se o caminho estiver bloqueado, o robô identifica pontos de borda, que são pontos de descontinuidade dentro do alcance do sensor. Ele então calcula o valor heurístico para cada borda e escolhe a borda com o menor valor para se dirigir, representado por caixas verdes nas figuras.
- **Navegação pelas Bordas:** O robô navega em direção a esta borda selecionada, contornando eficazmente o obstáculo para retomar seu caminho em direção ao objetivo.

Figura 1: Diagrama de implementação do Tangent Bug



Fonte: Elaborada pelo autor

Figura 2: Implementação



Fonte: <https://www.researchgate.net/>

O cálculo heurístico no algoritmo Tangent Bug usa a fórmula:

$$h(x, n) = d(x, n) + d(n, q_{goal}) \quad (1)$$

que soma a distância entre o ponto de borda e o ponto atual com a distância entre o ponto de borda e o objetivo.

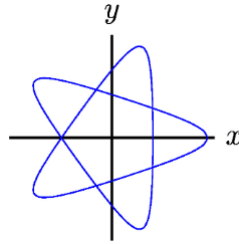
Se o valor heurístico previamente calculado for menor que o novo valor, o robô habilita o acompanhamento de contorno, seguindo a borda do obstáculo a uma distância constante, conforme visto no diagrama da Figura 1.

Durante esse processo, ele compara continuamente d_{reach} (a distância até o objetivo de pontos na borda) com $d_{followed}$ (a menor distância ao objetivo do obstáculo), e termina o acompanhamento de contorno se $d_{reach} < d_{followed}$, demonstrado na Figura 2.

2.2 Trajetória em Curvas Paramétricas

Para implementação do seguimento de trajetórias de curvas paramétricas, foi utilizado como solução, a discretização da equação paramétricas para obtenção de N pontos distintos dessa equação. Foi enviado ao robô, ponto a ponto, em sequência, fazendo com que o robô se desloca-se por toda a curva até o último ponto, após esse último, o primeiro ponto é novamente enviado ao robô, fazendo todo o ciclo se repetir.

Figura 3: Representação da Equação

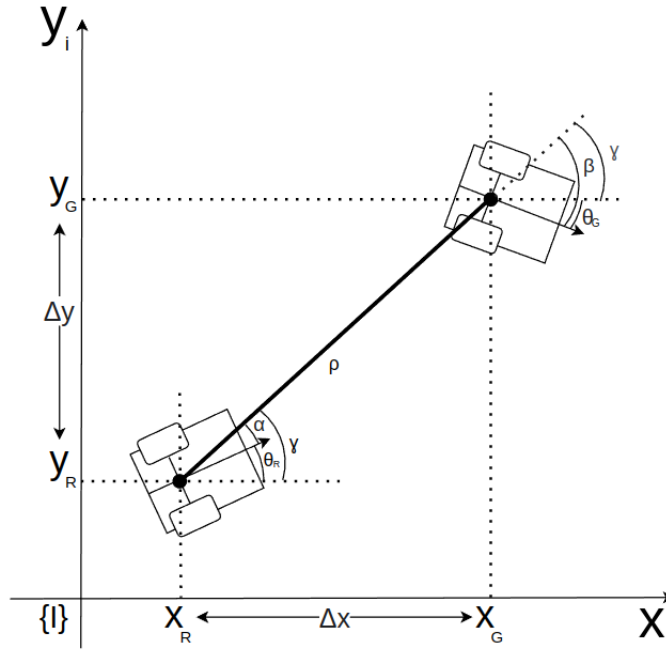


A equação paramétrica da Figura 3 utilizada nesse problema, como referência foi uma Hypotrochoid, com formato de estrela, dada a Equação 2.

$$\begin{aligned} x &= 2 \cdot \cos(t) + 5 \cdot \cos\left(\frac{2t}{3}\right) \\ y &= 2 \cdot \sin(t) - 5 \cdot \sin\left(\frac{2t}{3}\right) \end{aligned} \quad (2)$$

Para esse problema, foi desenvolvido um controlador por meio do da análise de estabilidade nos pontos de equilíbrio, com o objetivo de minimizar a distância ρ , determinada pela distância atual e o ponto de interesse, o ângulo α determinado pela diferença entre o ângulo atual do robô em relação ao ambiente γ e ângulo β que é o ângulo de chegada no ponto de interesse, conforme a Figura 4.

Figura 4: Representação em Coordenadas



Fonte: Elaborada pelo autor

Para esse controlador, cada uma dessas variáveis podem ser avaliadas com o erro, e são escritos pelas seguintes Equações 3.

$$\begin{aligned}\rho_t &= \sqrt{(\Delta x_t)^2 + (\Delta y_t)^2} \\ \alpha_t &= \gamma_t - \Theta_{R,t} \\ \beta_t &= \Theta_{G,t} - \gamma_t\end{aligned}\tag{3}$$

A variação no tempo de cada uma dessas equações pode ser relacionadas com a velocidade linear v e angular do robô w , conforme visto na Equação 4:

$$\begin{bmatrix} \dot{\rho}_t \\ \dot{\alpha}_t \\ \dot{\beta}_t \end{bmatrix} = \begin{bmatrix} \cos(\alpha_t) & 0 \\ \frac{\sin(\alpha_t)}{\rho} & -1 \\ -\frac{\sin(\alpha_t)}{\rho} & 0 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (4)$$

Essas velocidades estão diretamente ligadas aos erros descritos, assim, podendo ser reescritas com variáveis de controle, assim, $v_t = k_\rho \rho_t$ e $\omega_t = k_\alpha \alpha_t + k_\beta \beta_t$, resultando na Equação 5.

$$\begin{bmatrix} \dot{\rho}_t \\ \dot{\alpha}_t \\ \dot{\beta}_t \end{bmatrix} = \begin{bmatrix} -k_\rho \rho_t \cos(\alpha_t) \\ k_\rho \sin(\alpha_t) - k_\alpha \alpha_t - k_\beta \beta_t \\ -k_\rho \sin(\alpha_t) \end{bmatrix} \quad (5)$$

Esse é um sistema não linear e pode ser aproximado de um sistema linear por meio do seu jacobiano, resultado na Equação 6:

$$\mathbf{J}_f \begin{pmatrix} p \\ \alpha \\ \beta \end{pmatrix} = \begin{bmatrix} -k_\rho \cos \alpha & k_\rho \rho \sin \alpha & 0 \\ 0 & k_\rho \cos \alpha - k_\alpha & -k_\beta \\ 0 & -k_\rho \cos \alpha & 0 \end{bmatrix} \quad (6)$$

Avaliando esse sistema na origem, têm-se a Equação 7:

$$J_f \begin{pmatrix} p=0 \\ \alpha=0 \\ \beta=0 \end{pmatrix} = \begin{bmatrix} -k_\rho & 0 & 0 \\ 0 & k_\rho - k_\alpha & -k_\beta \\ 0 & -k_\rho & 0 \end{bmatrix} \quad (7)$$

Para se definir em quais regiões esse sistema é estável, é necessário encontrar os autovalores por meio do polinômio característico que é derivado a partir do determinante da matriz obtida subtraindo um múltiplo escalar da identidade, tal que, $\det(A - \lambda I)$, resultando na Equação 8:

$$(\lambda - k_\rho)\lambda^2 + (k_\alpha - k_\beta)\lambda - k_\rho k_\beta = 0 \quad (8)$$

As Equações 9 e 10 descrevem os autovalores encontrados:

$$\lambda_1 = -k_\rho \quad (9)$$

$$\lambda_{2,3} = \frac{-(k_\alpha - k_\rho) \pm \sqrt{(k_\alpha - k_\rho)^2 + 4k_\rho k_\beta}}{2} \quad (10)$$

Resolvendo, tem-se as Inequações 11,12 e 13, que descrevem as condições de estabilidade para funcionamento desse controlador:

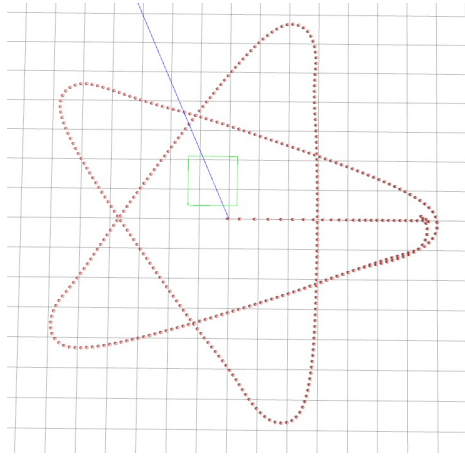
$$k_\rho > 0 \quad (11)$$

$$k_\alpha > k_\rho \quad (12)$$

$$k_\beta < 0 \quad (13)$$

Como resultado desse desenvolvimento pode ser visto o caminho percorrido pelo robô por meio da Figura 5. Percebe-se que o robô consegue seguir de forma fiel o caminho previsto pela sua equação paramétrica, outra questão importante é que a utilização do controle de k_β auxiliou na suavização da curva, na medida em que, o ângulo determinado para cada ponto foi realizado pela tangente entre os próximo pontos fazendo que o robô sempre estivesse direcionado para o próximo ponto a ser seguido.

Figura 5: Resultado Obtido da Equação Paramétrica



Link: <https://youtu.be/tFY7wHUzhqc>

2.3 Função de Potencial

Tendo em vista limitações, de uso de algoritmos, como uso do Tangent Bug em espaços bidimensionais, outros algoritmos foram desenvolvidos para solucionar o problema de movimentação de robôs. Conforme Choset Et al [1], a função de potencial utiliza o princípio de cargas de potencial distintos que podem ser atraídas

ou repelidas entre si de forma que o robô pode ser considerado uma determinada carga será atraído a um alvo de destino e obstáculos terão a função de repelir esse robô.

Assim, o ponto de alvo onde o robô irá se deslocar, pode ser obtido uma função de potencial, que por Choset Et al [1] é descrita como:

$$U_{\text{att}}(q) = \begin{cases} \frac{1}{2}\xi d^2(q, q_{\text{goal}}), & d(q, q_{\text{goal}}) \leq d_{\text{goal}}^* \\ d_{\text{goal}}^* \xi d(q, q_{\text{goal}}) - \frac{1}{2}\xi (d_{\text{goal}}^*)^2, & d(q, q_{\text{goal}}) > d_{\text{goal}}^* \end{cases} \quad (14)$$

O gradiente dessa função sempre irá ser direcionado ao ponto de equilíbrio, assim:

$$\nabla U_{\text{att}}(q) = \begin{cases} \xi(q - q_{\text{goal}}), & \text{if } d(q, q_{\text{goal}}) \leq d_{\text{goal}}^* \\ \frac{d_{\text{goal}}^* \xi (q - q_{\text{goal}})}{d(q, q_{\text{goal}})}, & \text{if } d(q, q_{\text{goal}}) > d_{\text{goal}}^* \end{cases} \quad (15)$$

Cada obstáculo medido por um sensor de distâncias, terá uma função de potencial definida por ?? e seu gradiente de cada um dos obstáculos ??:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{D(q)} - \frac{1}{Q^*} \right)^2, & \text{if } D(q) \leq Q^* \\ 0, & \text{if } D(q) > Q^* \end{cases} \quad (16)$$

$$U_{\text{rep}_i}(q) = \begin{cases} \frac{1}{2}\eta \left(\frac{1}{d_i(q)} - \frac{1}{Q_i^*} \right)^2, & \text{if } d_i(q) \leq Q_i^* \\ 0, & \text{if } d_i(q) > Q_i^* \end{cases} \quad (17)$$

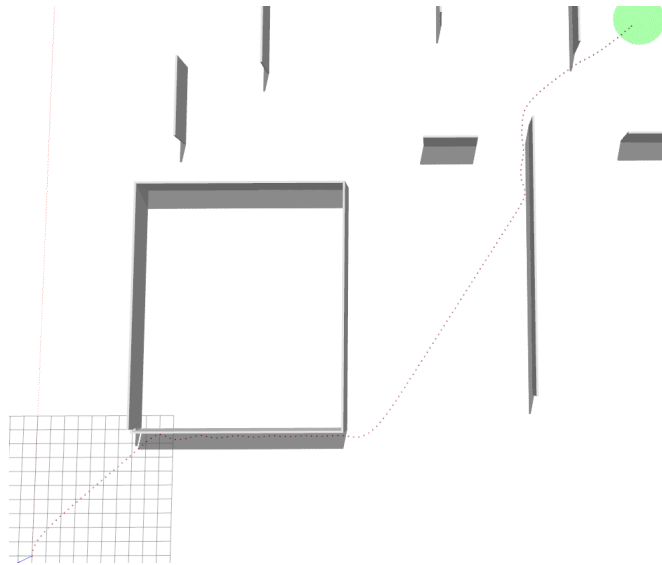
A função de potencial é determinada pelo somatório do potencial atrativo e repulsivo:

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \quad (18)$$

Foi utilizado para controle do robô um controlador baseado em que os valores de referencia utilizados são obtidos pelas funções de potencial. Têm-se portanto, o seguinte resultado.

Como pode ser visto, o robô é capaz de reproduzir o caminho de forma suave, com pequenas oscilações que são normais do método utilizado. Entretanto, o método possui o problema da criação de mínimos locais, que pode fazer com que o robô permaneça em um lugar fora do ponto de interesse.

Figura 6: Caminho Realizado



Link: <https://youtu.be/dcqBQgVfQv8>

2.4 Mínimos Locais

Mínimos locais, conhecidos em estratégia de funções de potencial, são pontos de mínimo na função gerada que não são globais para toda função e portanto é um problema que precisa ser solucionado. Como metodologia para resolução desse problema, foi utilizado como ideia principal o preenchimento virtual do mínimo local. A estratégia utilizada para sair dos mínimo local foi de realizar o preenchimento dos mínimos locais, de forma que o mínimo local se torne um lugar inacessível, pode-se imaginar como um copo preenchido por esferas, em um determinado momento irá transbordar pra fora, como visto na figura 7.

Para determinação da existência desse mínimo local, foi realizado uma verificação de posição do robô a partir de um raio ao redor do mesmo. Ao robô permanecer nesse raio por um valor determinado de tempo, considera-se então que o robô está em um mínimo local, em intervalos constatantes de tempo um novo raio é verificado.

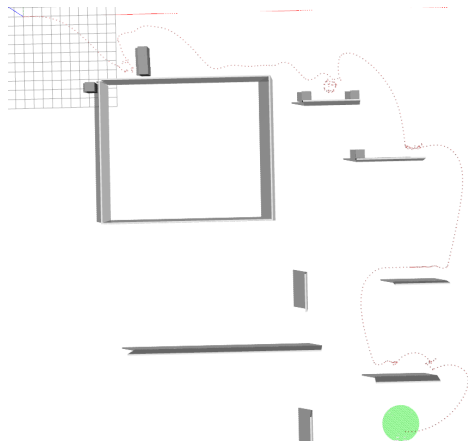
Ao se detectar um mínimo local, uma função de potencial, com a equação, é aplicada nessa localização com a finalidade de expulsar o robô daquela localização. Cada circulo foi preenchido por um potencial repulsivo com a Equação 19:

Figura 7: Copo Preenchido



Fonte: Elaborado pela DALL-E 3 — OpenAI

Figura 8: Caminho para escapar do mínimo local



Link: <https://youtu.be/bXtrt8KRDOY>

$$f(d) = \begin{cases} 0, & \text{se } d > r \\ \frac{k}{d^2}, & \text{se } d \leq r \end{cases} \quad (19)$$

Essa solução se mostrou eficiente para solução do problema, fazendo o robô convergir para o alvo após algum tempo, como pode ser visto na Figura 8.

3 Conclusão

Este trabalho investigou uma variedade de algoritmos essenciais para o planejamento de movimento de robôs, demonstrando tanto suas aplicações práticas quanto seus desafios teóricos. Por meio da exploração de métodos como o Tangent Bug, trajetórias paramétricas, e o uso de campos potenciais, identificamos estratégias eficazes para navegar em ambientes dinâmicos e imprevisíveis, apesar dos desafios impostos pelos mínimos locais.

As simulações realizadas confirmaram a aplicabilidade dos métodos estudados em contextos realísticos, onde a autonomia e precisão são cruciais. Além disso, as estratégias de superação de mínimos locais demonstraram ser um avanço significativo, potencializando a eficácia dos campos potenciais em evitar estados de bloqueio operacional do robô.

Por fim, a integração de abordagens tradicionais com inovações recentes sugere um caminho promissor para futuras pesquisas, onde a eficiência e a robustez dos sistemas robóticos podem ser ainda mais aprimoradas. Assim, este estudo não apenas reforça o entendimento das técnicas existentes mas também pavimenta o caminho para o desenvolvimento de novas soluções que poderão enfrentar os desafios emergentes no campo da robótica móvel.

Referências

- [1] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [2] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.