

Planejamento De Trajetórias Com Desvio De Obstáculos Usando Campos Potenciais Artificiais e *Probabilistic Roadmaps* Para Robótica Móvel

Paulo Victor M. R. Huguenin de Lima
Departamento de Engenharia Eletrônica e Telecomunicações
Universidade do Estado do Rio de Janeiro
Rio de Janeiro, Brasil
Email: pv.huguenin@gmail.com

Resumo—Devido aos avanços tecnológicos na área de veículos autônomos e robótica móvel, vem crescendo o interesse em buscar métodos para planejamento de trajetória com desvio de obstáculos para veículos autônomos. Um dos principais métodos globais mais usados para planejamento de trajetória são os campos potenciais artificiais e *Probabilistic Roadmaps*. Neste trabalho será visto a aplicação desses métodos para um veículo do tipo *car-like* para um cenário com obstáculos e sua aplicação para estacionamento paralelo, onde o objetivo é ilustrar a eficácia do algoritmo ao achar uma rota até o destino.

I. INTRODUÇÃO

A robótica móvel é amplamente utilizada em diversos campos industriais e tem sido bastante desenvolvida no decorrer dos anos. Seu objetivo é desenvolver dispositivos capazes de interagir com humanos e meio ambiente de forma autônoma e segura, sendo capaz de evitar obstáculos e assim cumprir suas devidas tarefas. Para isso o dispositivo deve ser capaz de perceber, localizar-se e mover-se pelo ambiente [1].

Como pedra fundamental das melhorias cabíveis por intermédio da movimentação autônoma figura o planejamento de trajetórias, definido como o processo de determinar uma lei de movimento que permita a um robô móvel se movimentar de uma pose inicial até uma pose final, sujeito à restrições. Trata-se de um tema fundamental em robótica móvel, existente em inúmeras aplicações tais como a navegação autônoma e cooperativa de múltiplos robôs [2], aspiradores de pó inteligentes, operação de veículos agrícolas [3] e carros autônomos.

O planejamento de trajetória é um dos problemas a serem resolvidos ao lidar com com navegação para robótica móvel. Os primeiros resultados no campo [4] datam do início da segunda metade do último século, mas na década de 80 a quantidade de pesquisas aumentaram, dado que com os resultados obtidos são usados até nos dias de hoje dependendo da aplicação [5], [6]. O planejamento da trajetória para um robô móvel é encontrar uma rota livre de colisões, evitando obstáculos, dado um local de início especificado a um desejado destino e pode ser interpretado de duas maneiras diferentes [7]. Primeiro, o planejamento global, em que o caminho é calculado a priori conhecendo o ambiente e o modelo do robô para determinar um caminho livre de colisão. Neste caso, uma

solução pode ser avaliada para cenários muito complexos, mas incertezas (mudança) no modelos do ambiente, ou do robô, podem levar ao fracasso. Nesta categoria é possível encontrar abordagens baseadas em *Roadmaps*, algoritmos de Dijkstra [8], decomposição de células, métodos de Campos Potenciais Artificiais [9], entre outros. A segunda categoria é representada pelas abordagens baseadas em sensores (reativas), também conhecida como planejamento local, a fim de evitar o conhecimento a priori do mapa e lidar com condições desconhecidas; entre eles é possível listar algoritmos baseados no comportamento de insetos que vivem em colônia e algoritmos de aprendizado de máquina como algoritmos genéticos e rede neurais. É fácil entender que uma combinação das duas categorias é a melhor solução para o problema de planejamento de caminho para robôs móveis e é de fato majoritariamente adotado [1], [9]. É possível incorporar também modelos probabilísticos nos modelos e sensores em todos níveis de planejamento [10]. Um dos modelos probabilístico mais usado é o PRM (*Probabilistic Roadmap*). Encontrar um caminho seguro em um ambiente perigoso para o robô móvel é um requisito essencial para o sucesso de qualquer robô móvel sistemas. Portanto, a pesquisa sobre algoritmos de planejamento de trajetória é um requisito fundamental para a segurança do robô móvel em tais ambientes.

Neste trabalho iremos focar no método de campos potenciais artificiais e probabilístico para planejamento de trajetória usando o modelo cinemático do tipo *car-like*. Estes algoritmos têm atraído pesquisadores devido a sua simplicidade de implementação e baixo custo computacional, além de poder ser implementado em modo *online* e *offline* [11]. Nesta abordagem nosso modelo de veículo aplicamos campos potenciais percorrendo um ambiente com obstáculos nos quais possuem forças repulsivas, enquanto o destino apresentará forças atrativas. Além disso, o algoritmo propõe uma possível solução para mínimos locais, um problema muito comum para este tipo de método. Já para aplicação com PRM temos um ambiente usando ocupação binária para identificar os obstáculos e buscar um caminho mais curto até o objetivo. Para ambos planejamento o controle *Pure Pursuit* será usado com função de manter o veículo em seu caminho planejado

mantendo os limites físicos do modelo bicicleta e limitando a velocidade linear e angular. Será visto também o planejamento de trajetória usando PRM considerando pose inicial e final para que o veículo seja capaz de realizar o estacionamento paralelo.

II. MODELO CINEMÁTICO

Neste trabalho será desenvolvido um planejamento de trajetórias para robôs móveis do tipo *car-like*. O modelo cinemático do carro é normalmente baseado no modelo de Ackerman [12], onde o sistema de esterçamento do veículo permite que a roda interna gire com um ângulo maior que a roda externa à curva, para evitar derrapagens. Para simplificar iremos usar o modelo de bicicleta [13], onde as rodas estão centradas no eixo do veículo e assumir que não há escorregamento e nem deslizamento das rodas, mais a garantia que em baixas velocidades nos quais a dinâmica é pouco relevante [14]. Logo podemos determinar as equações que determinam o modelo cinemático:

$$\phi = \frac{\phi_1 + \phi_2}{2}$$

$$\begin{cases} \dot{x} = V \cos \theta \\ \dot{y} = V \sin \theta \\ \dot{\theta} = \left(\frac{\tan \phi}{L} \right) \end{cases}$$

Este modelo representa um veículo com dois eixos separados por uma distância L . O estado do veículo é definido como um vetor de três elementos, $[x, y, \theta]$, com uma posição $[x, y]$ global, especificada em metros, e um ângulo de direção do veículo, θ , especificado em radianos. A roda dianteira pode ser girada com o ângulo de esterçamento ϕ . O rumo do veículo, θ , é definido no centro do eixo traseiro. V representa a velocidade longitudinal, enquanto que o vetor $[\dot{x}, \dot{y}, \dot{\theta}]$ representa as velocidades lineares e angular do carro [13]. A Figura 1 e a Tabela I exibem o modelo cinemático do veículo e seus respectivos parâmetros.

Tabela I: Parâmetros da cinemática do veículo

Parâmetros	Descrição
ϕ	Ângulo de esterçamento do modelo bicicleta
ϕ_1	Ângulo de esterçamento da roda 1
ϕ_2	Ângulo de esterçamento da roda 2
D	Distância entre as rodas do eixo dianteiro
A	Comprimento do veículo
B	Largura do veículo
f	Distância entre o eixo traseiro e a traseira do veículo
l	Distância entre o eixo dianteiro e a dianteira do veículo
L	Distância entre os eixos do veículo
R	Raio de curvatura instantâneo
θ	Ângulo do veículo em relação a eixo x
S	Distância percorrida
V	Velocidade
P	Ponto de referência entre o eixo traseiro

A. Controle Pure Pursuit

Com a finalidade manter o veículo em sua trajetória respeitando suas limitações de velocidade linear e angular é usado neste trabalho um controlador *Pure Pursuit*, no qual é

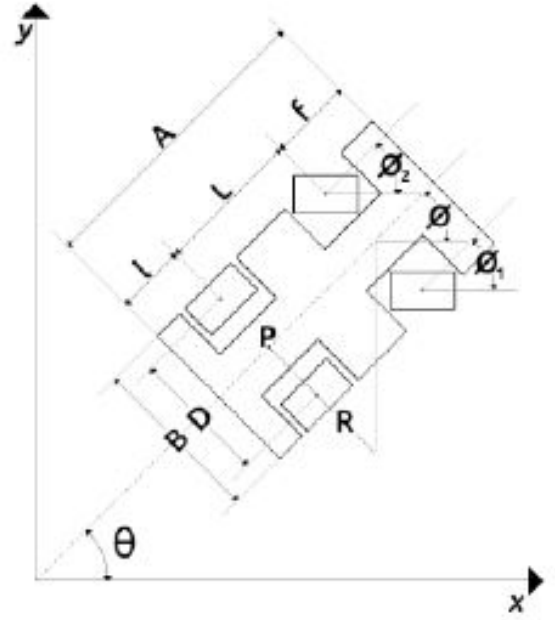


Figura 1: Modelo cinemático do veículo *car-like*

um controlador de rastreamento de caminho geométrico. Um controlador de rastreamento de caminho geométrico é qualquer controlador que rastreia um caminho de referência usando apenas a geometria da cinemática do veículo e o caminho de referência. O controlador *Pure Pursuit* usa um ponto de observação que é uma distância fixa no caminho de referência à frente do veículo, como segue. O veículo precisa prosseguir até esse ponto usando um ângulo de direção que precisamos calcular. Neste método, o centro do eixo traseiro é usado como ponto de referência no veículo. O controlador *Pure Pursuit* é um controle simples. Ele ignora as forças dinâmicas nos veículos e assume que as rodas não escorregam. Além disso, se ele estiver ajustado para baixa velocidade, o controlador seria perigosamente agressivo em altas velocidades. Uma melhoria é variar a distância de previsão com base na velocidade do veículo [15].

III. CAMPOS POTENCIAIS ARTIFICIAIS

Campos potenciais é um dos métodos mais usados para planejamento de trajetória global. A ideia principal deste método é imaginar que todos os obstáculos possam gerar forças potenciais repulsivas contra o veículo, enquanto que seu destino ofereça força atrativa ao robô. O método de campo potencial trata o robô representado como um ponto na configuração espaço como uma partícula sob a influência de um campo potencial artificial cujo local as variações devem refletir a “estrutura” do espaço livre [2]. O campo potencial pode ser imaginado como uma partícula carregada navegando através de um campo magnético ou uma bola de gude rolando colina abaixo. A ideia básica é que o comportamento exibido pela partícula. A função potencial é normalmente definido sobre o espaço livre como a soma de um potencial atraente

puxando o robô em direção à configuração do objetivo e um potencial repulsivo empurrando o robô para longe dos obstáculos, na qual pode ser definida como:

$$\mathbf{F}(\mathbf{q}) = \mathbf{F}_{at}(\mathbf{q}) + \sum_{i=1}^N \mathbf{F}_{rep_i}(\mathbf{q}) = -\nabla \dot{\mathbf{q}}$$

Onde as forças de atração e repulsão são dadas pelo cálculo do gradiente das velocidades lineares e angulares do modelo cinemático do veículo:

$$\nabla \dot{\mathbf{q}} = \begin{bmatrix} \frac{\partial \dot{\mathbf{q}}}{\partial x} & \frac{\partial \dot{\mathbf{q}}}{\partial y} \end{bmatrix}^T = \begin{bmatrix} F_x & F_y \end{bmatrix}^T \in R^2$$

A trajetória do veículo é gerada continuamente para qualquer posição (x, y) do robô seguindo a direção da descida mais íngreme no campo potencial (descida do gradiente), uma vez que é a direção da rede de força naquele ponto. Isso pode ser considerado um problema de otimização que procura o ponto de potencial mínimo. Supondo que o destino esteja localizado em (x_f, y_f) , a posição atual do veículo esteja em $(x(t), y(t))$ e que o potencial de atração pode ser dado de modo simplificado como:

$$F_{at}(\dot{\mathbf{q}}) = \mathbf{K}_a * \mathbf{q}$$

onde a distância $q(t) = \sqrt{(x(t) - x_f)^2 + (y(t) - y_f)^2}$. Logo aplicando o gradiente temos:

$$\mathbf{F}_{at_x}(t) = -\frac{\partial \dot{\mathbf{q}}}{\partial x} = \mathbf{K}_a \frac{(x_f - x(t))}{q(t)}$$

$$\mathbf{F}_{at_y}(t) = -\frac{\partial \dot{\mathbf{q}}}{\partial y} = \mathbf{K}_a \frac{(y_f - y(t))}{q(t)}$$

Para forças potenciais repulsivas de cada obstáculo supondo a mesma posição $(x(t), y(t))$ e posições de cada obstáculo (x_{obs}, y_{obs}) teremos:

$$F_{rep}(\dot{\mathbf{q}}) = \frac{\mathbf{K}_r}{q}$$

$$\mathbf{F}_{rep_x}(t) = -\frac{\partial \dot{\mathbf{q}}}{\partial x} = \frac{-\mathbf{K}_r(x_o b - x(t))}{q(t)^3}$$

$$\mathbf{F}_{rep_y}(t) = -\frac{\partial \dot{\mathbf{q}}}{\partial y} = \frac{-\mathbf{K}_r(y_o b - y(t))}{q(t)^3}$$

Portanto, a força total das componentes que afetam o veículo são:

$$F_{T(x,y)} = F_{at(x,y)} + \sum_{i=1}^N F_{rep_i(x,y)}$$

onde N é a quantidade de objetos no ambiente. O ângulo direcional do veículo é dado por:

$$\theta = \tan^{-1} \left(\frac{F_y}{F_x} \right)$$

1) *Problemas associados a campos potenciais:* Pode-se ver a partir da representação do campo potencial que o robô seguirá a direção da descida do gradiente, que é a direção da força resultante. Entretanto existem alguns problemas que podem impedir que o robô alcance o alvo. A soma das forças é igual a zero em qualquer ponto (x, y) onde o potencial líquido tem um local mínimo ou onde a derivada da descida do gradiente é igual a zero. Nestes pontos, o veículo irá parar mesmo não sendo o destino final. Esse problema é chamado de mínimo local pois o algoritmo entende que aquele ponto em que a derivada é igual a zero seja o mínimo global da função, mas normalmente a função pode apresentar alguns pontos de mínimo, sendo assim difícil para o algoritmo discernir os pontos de mínimo local e global. Outra situação em que ocorre problemas é quando têm-se muitos obstáculos próximos uns dos outros ou com passagens estreitas. O veículo tentará passar entre os obstáculos devido à força atrativa em direção ao alvo, mas não será capaz de passar entre os obstáculos. Tais problemas podem ser resolvidos adicionando uma força imaginária que resulta em pequenos movimentos que podem ajudar a sair dos mínimos locais. Alternativamente, pode-se modificar a altura ou dimensão dos campos de potencial de obstáculo, ou alterar o tipo de força usada para representá-los, ou use um controlador. Neste trabalho foi utilizado obstáculos imaginários nos pontos em que o veículo permanecer em mínimos locais, aplicando assim uma forças repulsivas suficientes para que o carro possa sair do mínimo local e com isso continuar seu percurso até o destino [16].

IV. Probabilistic Roadmaps

Probabilistic Roadmaps é um algoritmo de planejamento de trajetória no qual possui o objetivo de determinar um caminho entre uma configuração inicial do robô e uma configuração final, evitando colisões. Um exemplo de algoritmo de mapa aleatório probabilístico que explora caminhos viáveis em torno de uma série de obstáculos poligonais. A ideia básica por trás do PRM é pegar amostras aleatórias do espaço de configuração do robô, também conhecido como nós, testando-as para ver se estão no espaço livre e usar um planejador local para tentar conectar essas configurações a outras configurações próximas. Após as configurações de início e objetivo serem adicionadas o algoritmo interliga esses nós fazendo uma busca por um caminho que leve o veículo da posição inicial até a final [17].

O PRM consiste em duas fases: uma fase de construção e uma fase de consulta. Na fase de construção, é construído um *roadmap*, aproximando os movimentos que podem ser feitos no ambiente. Primeiro, uma configuração aleatória é criada. Em seguida, ele é conectado a alguns vizinhos, normalmente os k vizinhos mais próximos ou todos os vizinhos a menos que uma distância predeterminada. Configurações e conexões são adicionadas ao gráfico até que o roteiro seja denso o suficiente. Na fase de consulta, as configurações de início e objetivo são conectadas ao gráfico e o caminho é obtido por uma consulta de caminho mais curto de *Dijkstra*. Os algoritmos 1 e 2 apresentam os pseudo-códigos da fase de construção do mapa e de consulta pelo caminho mais curto respectivamente.

Dadas certas condições relativamente fracas na forma do espaço livre, o PRM é comprovadamente completo, o que significa que à medida que o número de pontos amostrados aumenta sem limites, a probabilidade de que o algoritmo não encontre um caminho se houver se aproxima de zero. A taxa de convergência depende de certas propriedades de visibilidade do espaço livre, onde a visibilidade é determinada pelo planejador local. Grosso modo, se cada ponto puder "ver" uma grande fração do espaço, e também se uma grande fração de cada subconjunto do espaço puder "ver" uma grande fração de seu complemento, o planejador encontrará uma trajetória rapidamente [17].

Algoritmo 1 PRM:Fase de Construção do mapa

Pseudo-Código PRM:Fase de Construção do mapa
Entrada:

n: Número de nós

k: número de vizinhos mais próximos

Saída:PRM $G = (V, E)$

```
{
  V = 0
  E = 0
  enquanto |V| < n faça
  {
    repetir
      q ← Gera aleatoriamente os nós no espaço livre do
      ambiente
    até q estiver livre de colisões
  }
  para todo q ∈ V faça
    Nq ← seleciona os k vizinhos mais próximos de q
    escolhidos de V
    para todo q' ∈ Nq faça
      se (q, q') ∉ E e Δ(q, q') ≠ 0 então
        E ← E ∪ {(q, q')}
```

V. PRM APLICADO A ESTACIONAMENTO PARALELO

Para os planejamentos de trajetória vistos até agora não foram considerados a pose e nem as dimensões do veículo. Os algoritmos calculavam uma trajetória e aplicávamos o modelo cinemático do veículo a um controlador para seguir tais caminhos limitando a velocidade e suas restrições físicas. Contudo, para realização do estacionamento paralelo do veículo é necessário desenvolver um algoritmo que trace uma trajetória considerando o modelo cinemático do veículo, dimensões físicas e a pose θ . O problema é transformado de um sistema $[x, y]$ para $[x, y, \theta]$. Sendo assim será realizada algumas mudanças no algoritmo 2.

O algoritmo foi revisado, impondo um método com restrição não holonômica do veículo relacionando θ com xey . A ideia de alto nível do algoritmo de restrição não holonômica é dada do seguinte modo [18]:

- 1) Observe as configurações próximas na estrutura do corpo do estado do robô

Algoritmo 2 PRM:Fase de busca pelo melhor caminho

Pseudo-Código PRM: Fase de busca pelo melhor caminho
Entrada:

q_i : configuração inicial

q_f : configuração final

Saída:PRM Caminho de q_i até q_f ou erro

```
{
  Nqi ← os k vizinhos mais próximos de qi de V de acordo
  com a distância
  Nqf ← os k vizinhos mais próximos de qf de V de acordo
  com a distância
  V ← {qi} ∪ {qf} ∪ V
  q' ← vizinho mais próximo de qi em Nqi
  repetir
  {
    se Δ(qi, q') ≠ 0 então
      E ← (qi, q') ∪ E
    senão
      defini q' para ser o próximo vizinho mais próximo de
      qi
    até a conexão ser bem sucedida ou Nqi estiver vazio
    repetir se Δ(qf, q') ≠ 0 então
      E ← (qf, q') ∪ E
    senão
      defini q' para ser o próximo vizinho mais próximo de
      qf
  }
  até a conexão ser bem sucedida ou Nqf estiver vazio
  P ← caminho mais curto(qi, qf, G)
  se P ∉ {} então
    retorna
  senão
    retorna falha
}
```

- 2) Com base nas coordenadas euclidianas de outra configuração $(x; y)$, calcule seu ângulo θ permitido usando o modelo não-holonômico
- 3) Se o ângulo do estado amostrado θ estiver dentro da tolerância calculada, em seguida
- 4) conecte o grafo

Isso pode ser descrito matematicamente como segue a seguir. Dado uma configuração de entrada $X_i = [x_i, y_i, \theta_i]^T$ e $X_f = [x_f, y_f, \theta_f]^T$ podemos usar transformação usando deslocamentos planares como segue a seguir

$$X = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} R(-\theta_i) \left(\begin{bmatrix} x_f \\ y_f \end{bmatrix} - \begin{bmatrix} x_i \\ y_i \end{bmatrix} \right) \\ \theta_f - \theta_i \end{bmatrix}$$

Aplica-se então as equações do modelo cinemático visto na seção II. Então, a verificação final é comparar este valor com a amostra θ_0 para ver se esta é uma configuração permitida. Nós podemos fazer isso definindo uma tolerância tol para o planejador de caminho. Finalmente, se a aresta estiver

conectada, podemos pesar as arestas pela distância euclidiana entre a posição dos dois estados, então algoritmo 2 possa encontrar o caminho mais perto.

Para o estacionamento será definido que a velocidade poderá assumir valores entre $v = [-1, 1]m/s$, assim o veículo não irá rápido o bastante para que a roda deslize e permite que o veículo também de ré. Assim como o PRM simples foi utilizado o algoritmo Dijkstra's para poder tentar encontrar o caminho mais curto.

VI. AMBIENTE DE SIMULAÇÃO

Para a simulação do algoritmo é considerado um ambiente com obstáculos, onde se é definido alguns pontos de partida e pontos de chegada. O mapa utilizado neste trabalho consiste em 2 pontos de partida e 3 pontos de destino com 3 obstáculos estáticos poligonais como pode ser visto pela Figura 2. Para cada método de planejamento de trajetória serão criados os mesmos ambientes, porém serão usados abordagens diferentes para a construção de cada um deles.

A. Ambiente Para Campos Potenciais

Na Seção III observou-se que as forças potenciais descritas são circulares, enquanto que os obstáculos são polígonos convexos. Para resolver esse problema pode-se assumir que cada lado do polígono seja dividido em c círculos idênticos de forma que em cada vértice do polígono há um círculo.

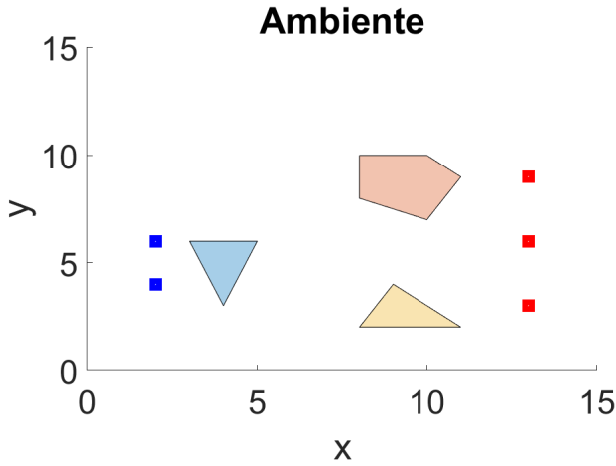


Figura 2: Ambiente de simulação.

A posição desses obstáculos que compõe o polígono são calculados da seguinte forma:

$$x = \frac{mx_2 + nx_1}{m + n}$$

$$y = \frac{my_2 + ny_1}{m + n}$$

onde (x_1, y_1) e (x_2, y_2) são os pontos de 2 vértices que compõe um lado divididos por uma razão de m para n . Após configurado o ambiente de simulação definindo as forças atrativas e repulsivas no sistema é possível visualizar o modelo potencial. A Figura 3 ilustra o modelo potencial artificial em \mathbb{R}^3 .

Ambiente Artificial

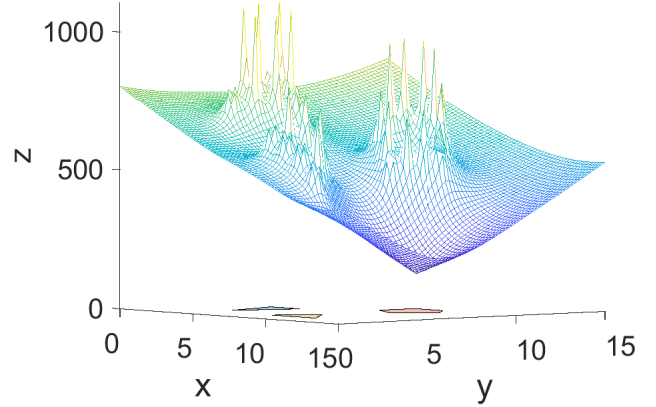


Figura 3: Ambiente criado artificialmente em superfície com campos potenciais

B. Ambiente Para PRM

Um dos procedimentos mais comuns para criar um ambiente para robótica móvel é usando *Occupancy Map*. A ideia básica da *Occupancy Map* é representar um mapa do ambiente como um campo uniformemente espaçado de variáveis aleatórias binárias, cada uma representando a presença de um obstáculo naquele local no ambiente [19]. O algoritmo calculam estimativas posteriores aproximadas para essas variáveis aleatórias. Como as variáveis são binárias teremos que o espaço livre será representado com "0" e os obstáculos com "1". Graficamente o espaço livre representa a área branca e a área preta contém os obstáculos.

VII. EXPERIMENTOS E RESULTADOS

O modelo matemático descrito na Seção III e os algoritmos propostos neste trabalho junto com a simulação do planejamento de trajetória do veículo foi realizada usando o software MATLAB®. Alguns dos parâmetros utilizados no experimento podem ser vistos pela Tabela II.

Tabela II: Parâmetros da simulação do algoritmo

Parâmetros	Descrição
Ambiente	
$[x_0, y_0] = \begin{cases} [1, 3] \\ [1, 6] \end{cases}$	Posições iniciais do veículo
$[x_f, y_f] = \begin{cases} [9, 8] \\ [9, 6] \\ [9, 3] \end{cases}$	Posições finais do veículo
Veículo	
$\phi = \pm \frac{\pi}{4} rad$	Ângulo de esterçamento limite
$\lambda = 0.2$	Passo do veículo
$L = 0, 2m$	Distância entre os eixos do veículo
$V = 0.5m/s$	Velocidade do veículo
Campos Potenciais	
$K_{at} = 1.5$	Coefficiente de atração
$K_{rep} = 2.5$	Coefficiente de repulsão
PRM	
$n = 100$	Número de nós
$k = 20$	Vizinhos mais próximos

A. Campos Potenciais

Após a definição dos parâmetros é necessário definir o ambiente potencial. Para isso o algoritmo calcula para cada ponto do mapa os potenciais de atração dado um destino e potenciais de repulsão dado cada obstáculo. Pela Figura 4 pode-se observar separadamente os potenciais. Realizando apenas a diferença de ambas as matrizes que representam estes potenciais é possível obter o potencial total como pode ser observado na Figura 5. Comparando a Figura 4 e a Figura 5 é possível observar como o ambiente fica distorcido com a presença dos obstáculos e consequentemente com a diferença de ambos potenciais, logo, regiões próximas a esses obstáculos podem vir a formar mínimos locais. Para o algoritmo resolver o problema do mínimo local foi criado artificialmente obstáculos com a ideia de exercer forças repulsivas suficientes para que o veículo saia dessa região caso o algoritmo perceba após algumas iterações que o veículo continua preso nesta região.

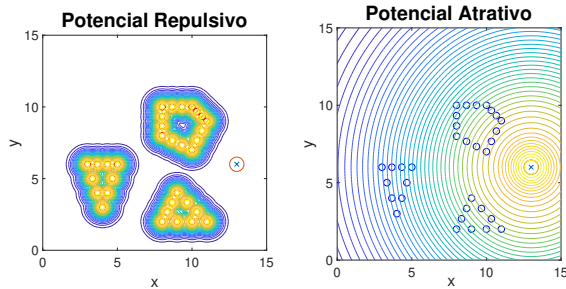


Figura 4: Potenciais de repulsão e atração distribuídos separadamente no mapa

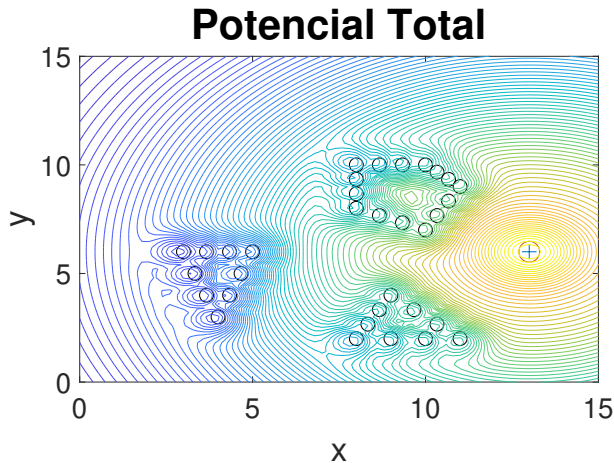


Figura 5: Potencial total do ambiente

Posteriormente maximiza-se a função de custo para o algoritmo saber qual caminho tomar após cada iteração atualizando sempre os parâmetros cinemáticos do veículo a cada iteração. Por fim, a Figura 6 ilustra a efetividade do algoritmo ao achar para todas as configurações de partida e chegada escolhidas uma trajetória sem colisões. Em alguns casos podemos ver que

o algoritmo precisou usar o recurso de obstáculo imaginário para poder sair de um mínimo local e por conseguinte alcançar o objetivo. Um dos problemas encontrados usando este método foi a dificuldade do veículo passar por caminhos estreitos. As forças repulsivas entre dois objetos acabam obstruindo a passagem mesmo que haja espaço físico para o veículo atravessar o percurso. Pela Figura 7 é possível observar que o algoritmo acaba encontrando apenas caminhos que contornem os obstáculos, o que em muitas das vezes não é o desejado.

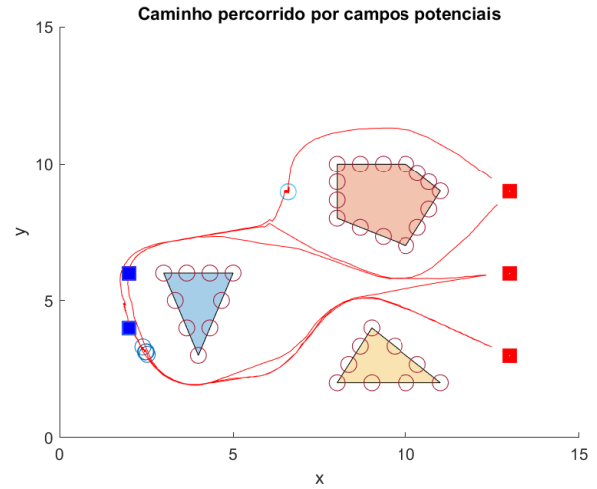


Figura 6: Trajetória do veículo para o ambiente apresentado usando campos potenciais

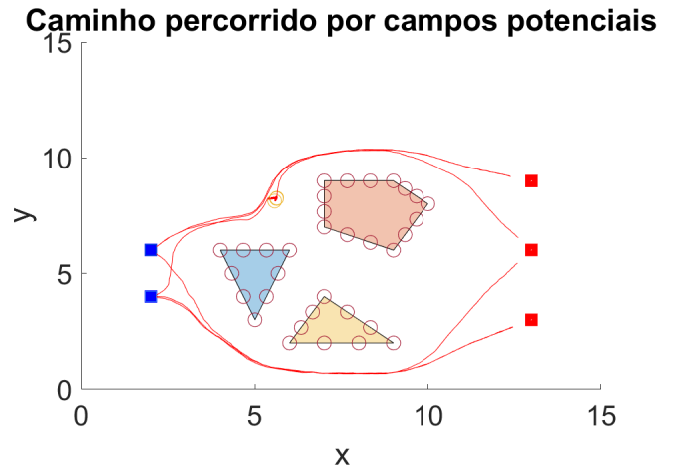


Figura 7: Trajetória do veículo para caminhos estreitos usando campos potenciais

Após obtermos os caminhos calculados pelo algoritmo iremos aplicar o nosso modelo de veículo para seguir esse caminho com a cinemática do modelo bicicleta obedecendo suas restrições. A figura 8 ilustra este resultado de forma satisfatória

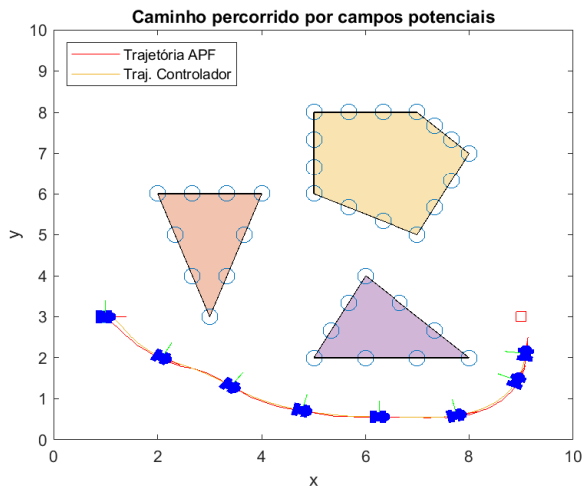


Figura 8: Trajetória do veículo usando o modelo cinemático por campos potenciais

B. PRM

Definidos os hiper-parâmetros é possível construir o ambiente. Como foi mencionado na subseção VI-B o mapa é criado usando ocupação binária. Os mesmos polígonos usados para campos artificiais são usados neste método também. Esse procedimento de criação de mapa é útil para este método pois quando o algoritmo PRM distribuir os nós aleatoriamente pelo mapa garante que todos os nós ocupem o espaço livre do ambiente. Pela figura 9 é possível visualizar a distribuição dos nós interligados com $n = 50$ e $k = 20$ pelo mapa definido pelo algoritmo 1.

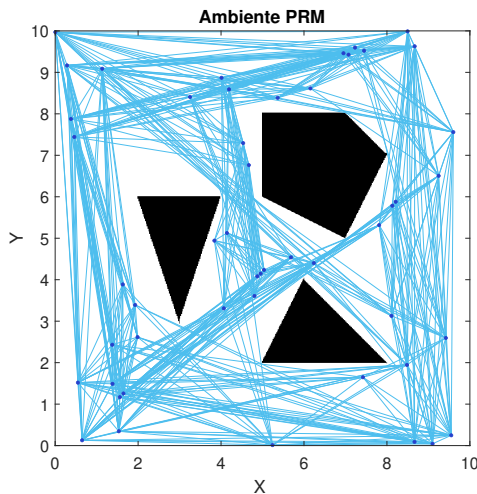


Figura 9: Ambiente de simulação PRM para $n = 50$ e $k = 20$

Com base neste mapa o algoritmo 2 ficará responsável, para dado um ponto inicial e de destino do veículo, interligar os nós que compõe o caminho mais curto até o objetivo. O que determina o caminho mais curto são os parâmetros passados ao algoritmo. Pela figura 12a e 12b ilustra os caminhos escolhidos

para um mapa com 50 e 100 nós respectivamente. Nota-se que com uma maior quantidade de nós a complexidade do mapa aumenta e com isso rotas mais curtas são traçadas. Entretanto, tais rotas tendem a passar muito próximas aos obstáculos, podendo haver colisão quando um modelo real do veículo for usado. Podemos observar que com um mapa com menor complexidade possui uma menor probabilidade de ocorrer colisões. Outro problema é o uso de um baixo valor de k . Usando poucos vizinhos próximos é possível que o mapa não consiga ficar totalmente interligado, não sendo possível achar um caminho, como é ilustrado pela Figura 10.

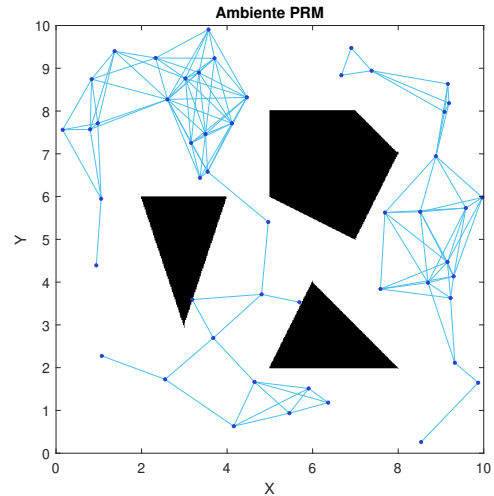


Figura 10: Ambiente PRM com $n = 50$ e $k = 2$

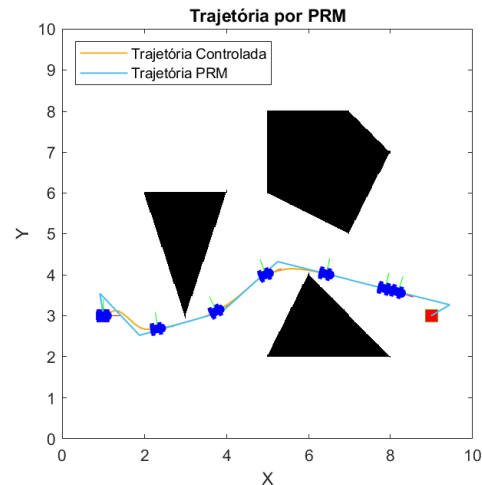
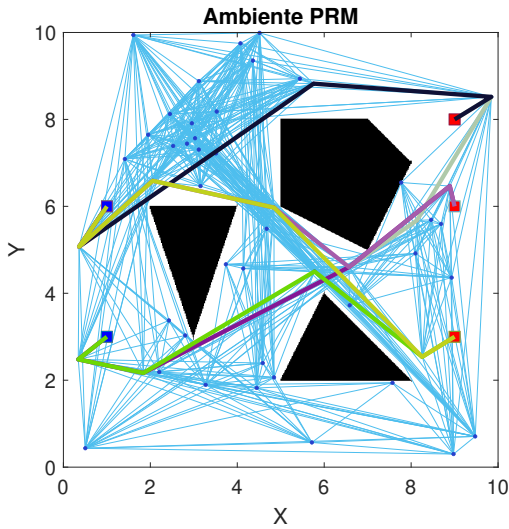
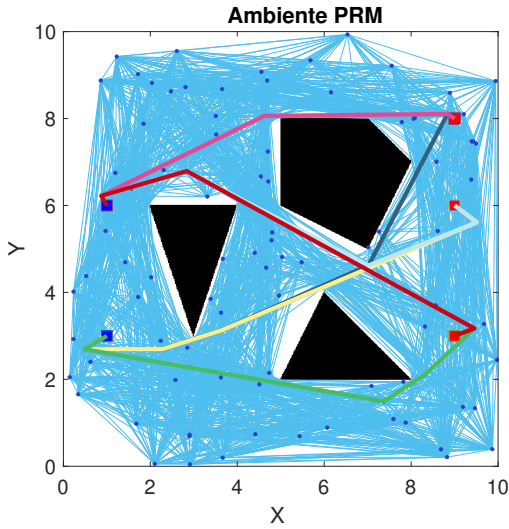


Figura 11: Trajetória do veículo usando PRM com trajetória calculada e trajetória pós controle

Igualmente ao que foi feito para campos artificiais, iremos aplicar o modelo cinemático do veículo ao caminho calculado pelo método PRM, ilustrado pela figura 13 e figura 11. Podemos observar que a trajetória dada pelo controlador considera as limitações do veículo e ajusta o caminho usando



(a) Trajetória do veículo usando PRM com $k = 20$ e $n = 50$



(b) Trajetória do veículo usando PRM com $k = 20$ e $n = 100$

Figura 12: Trajetória calculada pelo algoritmo PRM

a cinemática do modelo bicicleta controlando sua velocidade. Para todos pontos iniciais e finais o veículo chegou ao destino.

C. Comparação de Resultados

Ambos os modelos foram capazes de achar uma trajetória evitando os obstáculos até chegar ao destino. Contudo, quando comparamos dois modelos precisamos considerar outros fatores que determinam a qualidade de cada modelo. Uma dessas medidas é o tempo que levou para o veículo sair do ponto de partida até ao destino. Pela tabela III é possível visualizar a média e o desvio padrão do tempo gasto pelo veículo até cumprir o seu objetivo. Ambos os planejamentos utilizaram em seu controlador a velocidade limite de 0,5 m/s, logo o tempo e a distância são diretamente proporcionais e indica que o menor tempo tem grande chance de ser também a

Tabela III: Tempo estimado para trajetórias por APF e PRM

início - fim	$tempo_{APF}(s)$	$tempo_{PRM}(s)$
i1-f1	$25,32 \pm 1,08$	$19,36 \pm 0,48$
i1-f2	$26,56 \pm 1,16$	$16,48 \pm 0,18$
i1-f3	$19,56 \pm 0,17$	$16,32 \pm 0,33$
i2-f1	$18,16 \pm 0,09$	$16,52 \pm 0,98$
i2-f2	$21,6 \pm 0,2$	$16,28 \pm 0,23$
i2-f3	$24,88 \pm 0,11$	$16,88 \pm 0,94$

menor distância. Percebe-se que para todas as combinações de configurações iniciais e finais o algoritmo PRM obteve melhor desempenho em comparação com o planejamento por campos potenciais. Isso pode ser explicado devido ao método APF ter dificuldades de passar por obstáculos muito próximos um do outro, enquanto que o PRM não apresentou dificuldades para o ambiente deste trabalho. Além do tempo e da distância tem ainda o problema do mínimo local, que apesar do veículo conseguir sair com o uso de obstáculos imaginários há desvios de sua trajetória original, aumentando assim ainda mais o tempo e a distância. Em relação ao planejamento de trajetória o método usado pelo APF apresenta curvas suaves e quando foi aplicado o controlador a este modelo de trajetória quase não percebeu diferença entre os dois caminhos. Enquanto o PRM pode apresentar mudanças bruscas na trajetória que mesmo usando um controlador para limitar a velocidade e limitações físicas do veículo é possível que esta suavização da curva passe rente ao algum obstáculo, havendo assim colisão. Uma possível solução é aumentando consideravelmente a quantidade de nós no mapa, assim os pontos escolhido são cada vez mais próximos e assim suaviza a curva da trajetória.

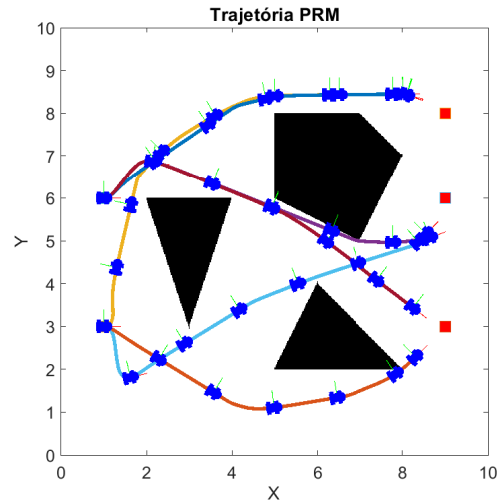


Figura 13: Trajetória do veículo usando o modelo cinemático por PRM

D. Estacionamento Paralelo

Para o estacionamento paralelo serão consideradas duas situações: A que a dianteira do carro está mais próxima da vaga e a que a traseira está mais próxima da vaga. Os obstáculo

são gerados da mesma forma usando Ocupação binária. Será considerada a dimensão do veículo também, então o algoritmo que distribui os pontos aleatoriamente pelo mapa irá considerar que os obstáculos são maiores do que realmente são. Outra consideração que deve ser levada em conta é quantidade de nós utilizados. Um número pequeno de nó irá poder causar oscilações no veículo. Com isso a ideia é considerar o mapa com uma alta resolução. Para estas aplicações foram gerados 10.000 nós. Pelas figuras 14 e 15 podemos ver o mapa que simula uma vaga para o estacionamento paralelo. Os pontos em azul representam os nós distribuídos no ambiente livre e a figura representa o carro através de poses ao longo do tempo. Os dois pontos amarelos simbolizam os faróis do carro e consequentemente a frente do veículo. É possível observar que para o estacionamento de ré foi mais simples para o veículo fazer e que demorou menos tempo do que o estacionamento de frente. Nota-se que para o estacionamento de frente o veículo precisou manobrar o carro algumas vezes e o próprio algoritmo achou um caminho para que precise passar o limite da vaga para poder estacionar. Se a vaga fosse maior o veículo seria capaz de entrar de frente direto, mas devido a sua construção física de que apenas as rodas da frente fazem o ângulo de estercamento dificulta o veículo fazer este tipo de estacionamento. O que é bem comum na prática. Normalmente o condutor ultrapassa os limites da vaga para poder começar a executar o estacionamento de ré. Além disso, o tempo do estacionamento de ré foi duas vezes menor que o estacionamento começando de frente.

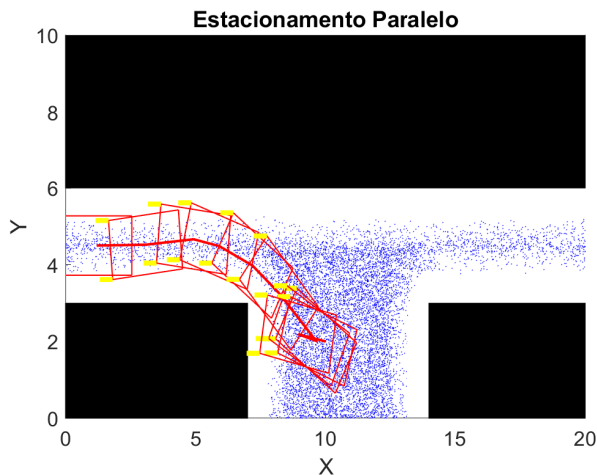


Figura 14: Estacionamento de Ré por PRM

VIII. CONCLUSÃO

Neste trabalho propõe-se o uso do algoritmo de campos potenciais artificiais com solução para o mínimo local e PRM(Probabilistic Roadmap) para o veículo do tipo *car-like* inspirado na cinemática do modelo bicicleta. O propósito deste método consiste em poder fazer o cálculo de uma trajetória para o veículo sem que haja colisões durante o percurso. Com base nos resultados apresentados vimos que ambos os

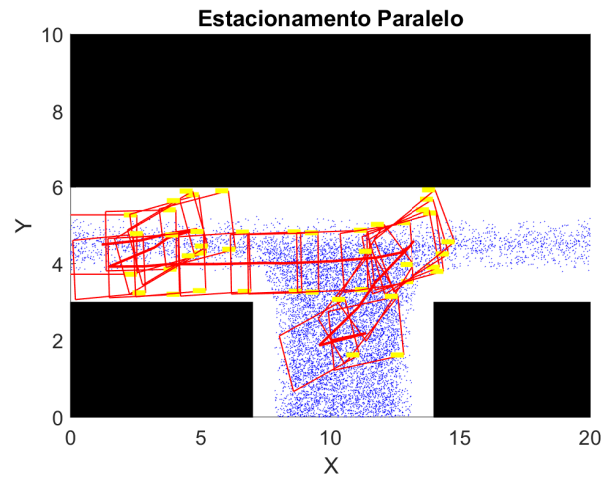


Figura 15: Estacionamento de frente por PRM

algoritmos foram capazes de traçar trajetórias ao redor dos obstáculos mesmo ao passar por mínimos locais. Contudo quando foi apresentado ambientes estreitos ao algoritmo de campos potenciais só foi capaz de traçar trajetórias contornando externamente os obstáculos. Em contrapartida o algoritmo PRM lidou melhor com caminhos estreitos procurando sempre o caminho mais curto, apesar de apresentar trajetórias que passem mais próximos aos obstáculos comparando com o outro método. Em relação ao tempo que o veículo chegou ao destino foi visto que para PRM o carro chegou antes do outro modelo para todas as posições iniciais e finais configuradas para o planejamento. Por fim, foi aplicado o algoritmo PRM modificado, considerando pose inicial e final para a realização de estacionamento. Para ambos as situações o algoritmo foi capaz de achar um caminho que seja possível realizar o estacionamento. Como trabalhos futuros deseja-se estudar maneiras de unificar os dois métodos e analisar seu desempenho e estudar outros ambientes e métodos globais e locais para solução de trajetórias para veículos autônomos.

REFERÊNCIAS

- [1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.
- [2] G. R. Petry, "Navegação de um robô móvel em ambiente semi-estruturado," 2019.
- [3] R. M. Pinto, "Geração e controle de trajetória ponto a ponto para veículos agrícolas de grande porte," Ph.D. dissertation, Universidade de São Paulo.
- [4] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.
- [5] Y. K. Hwang and N. Ahuja, "Gross motion planning—a survey," *ACM Computing Surveys (CSUR)*, vol. 24, no. 3, pp. 219–291, 1992.
- [6] O. Souissi, R. Benatallah, D. Duvivier, A. Artiba, N. Belanger, and P. Feyzeau, "Path planning: A 2013 survey," in *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*. IEEE, 2013, pp. 1–8.
- [7] E. Masehian and D. Sedighzadeh, "Classic and heuristic approaches in robot motion planning—a chronological review," *World Academy of Science, Engineering and Technology*, vol. 23, no. 5, pp. 101–106, 2007.
- [8] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

- [9] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.
- [10] A. M. Ladd and L. E. Kavraki, "Measure theoretic analysis of probabilistic path planning," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 229–242, 2004.
- [11] A. Rajvanshi, S. Islam, H. Majid, I. Atawi, M. Biglerbegian, and S. Mahmud, "An efficient potential-function based path-planning algorithm for mobile robots in dynamic environments with moving targets," *British Journal of Applied Science and Technology*, vol. 9, no. 6, pp. 534–550, 2015.
- [12] T. D. Gillespie, *Fundamentals of vehicle dynamics*. Society of automotive engineers Warrendale, PA, 1992, vol. 400.
- [13] B. C. Pinheiro *et al.*, "Sistema de controle tempo real embarcado para automação de manobra de estacionamento," 2009.
- [14] C. Zhu and R. Rajamani, "Global positioning system-based vehicle control for automated parking," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 220, no. 1, pp. 37–52, 2006.
- [15] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.
- [16] P. E. Stingu and F. L. Lewis, "Motion path planning for mobile robots," *target*, vol. 10, p. 10, 2007.
- [17] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [18] H. J. Suh, J. Deacon, and Q. Wang, "A fast prm planner for car-like vehicles."
- [19] S. Thrun, W. Burgard, D. Fox *et al.*, "Probabilistic robotics, vol. 1," 2005.