

Self-driving Paths for Automobile Urban Parking^{*}

Renan P. Vieira^{*} Eduardo V. Argento^{**}
Téo C. Revoredo, IEEE Member^{***}

^{*} Programa de Pós-Graduação em Engenharia Eletrônica,
Departamento de Engenharia Eletrônica e de Telecomunicações,
Universidade do Estado do Rio de Janeiro, RJ (e-mail:
renan.porto.vieira@gmail.com).

^{**} Departamento de Engenharia Elétrica, Pontifícia Universidade
Católica do Rio de Janeiro, RJ (e-mail:
eduardoverasargento@gmail.com).

^{***} Departamento de Engenharia Eletrônica e de Telecomunicações,
Universidade do Estado do Rio de Janeiro, RJ (e-mail:
teorevoredodo@uerj.br).

Abstract: This text discusses autonomous vehicles for parking in smart cities. Three path generation methods are implemented to avoid obstacles: geometric shapes, polynomial parameterization, and genetic algorithm. Tracking uses a pure pursuit algorithm. The results in a simulated 3D environment show the effectiveness of the algorithms. For example, for parallel front-in parking, the method based on genetic algorithm was the fastest (17.59 s) and shortest (1.768 m). For reverse perpendicular parking, the method with geometric shapes performed best (16.39 s and 1.645 m). This provides a basis for further investigations into autonomous parking.

Resumo: Este texto discute veículos autônomos para estacionamento em cidades inteligentes. Três métodos de geração de caminhos são implementados para evitar obstáculos: formas geométricas, parametrização polinomial e algoritmo genético. O rastreamento usa um algoritmo de perseguição pura. Os resultados em ambiente 3D simulado mostram a eficácia dos algoritmos. Por exemplo, para estacionamento paralelo de frente, o método baseado em algoritmo genético foi o mais rápido (17,59 s) e curto (1,768 m). Já para perpendicular de ré, o método com formas geométricas foi o melhor (16,39 s e 1,645 m). Isso fornece uma base para mais investigações sobre estacionamento autônomo.

Keywords: Car-like mobile robots; Motion planning; autonomous parking; Genetic algorithm.

Palavras-chaves: Robôs móveis semelhantes a carros; Planejamento de movimento; Estacionamento autônomo; Algoritmo genético.

1. INTRODUÇÃO

Veículos leves de passeio foram por muito tempo considerados o reino dos engenheiros mecânicos. Entretanto, avanços em tecnologia da informação, robótica e automóveis estão transformando o carro tradicional em uma máquina inteligente capaz de interagir com outros veículos e a infraestrutura das cidades, além de prover entretenimento aos usuários. Tais veículos interpretam o ambiente ao redor, tomam decisões rápidas e oportunas, navegam sem intervenção humana, mantêm padrões de mobilidade seguros e executam diferentes tipos de manobras (Rashid et al., 2019). O estacionamento é um dos maiores desafios nas grandes cidades modernas devido ao aumento do número de veículos e da densidade populacional, além

^{*} O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores agradecem, também, à Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ) pelos seus suportes financeiros no desenvolvimento deste trabalho.

da necessidade de manter distanciamento suficiente entre veículos estacionados. A capacidade de estacionamento autônomo alivia esses problemas, economizando espaço em estacionamentos (Hussain and Zeadally, 2019).

Nesse cenário, muitas técnicas vêm sendo investigadas para o planejamento de caminhos para estacionar carros autônomos. D. Wang et al. (Wang et al., 2011) usa um método de dois arcos para modelar o caminho de estacionamento paralelo e otimizar seu comprimento total, entretanto, o caminho é limitado a dois seguimentos. E. Argento et al. (Argento et al., 2020) combina uma técnica de interpolação para gerar caminhos suaves com um algoritmo de busca por força bruta. Todavia, a abordagem oferece apenas soluções sub ótimas determinadas pelo intervalo de busca, embora abordagens deste tipo garantam alta exatidão de rastreamento enquanto evitam modos naturais da estrutura mecânica ou do sistema de servocontrole (Erkorkmaz and Altintas, 2001).

Na busca por otimizar a solução de problemas de movimentação autônoma, dá-se destaque à utilização de inteligência

artificial (IA) (Ni et al., 2018). Entre as técnicas utilizadas, os algoritmos genéticos (AG) estão entre as mais promissoras sendo passíveis de aplicação direta no processo de planejamento. R. Vieira et al. (Vieira et al., 2022) aplica AGs para otimizar caminhos gerados por interpolações que levam o veículo a efetuar o estacionamento paralelo.

Este trabalho tem por objetivo propor e investigar soluções para o planejamento de caminhos para o estacionamento de carros autônomos. Assumindo uma pose inicial conhecida e uma pose final pré-definida dentro de uma vaga, os algoritmos devem especificar um caminho realizável que leve o veículo da pose inicial à pose final, desviando-o de eventuais obstáculos. Uma vez definido o caminho, ele é efetuado por meio de um controlador de perseguição pura (Coulter, 1992). Dessa forma, este trabalho contribui com a comparação das características de diferentes planejadores de caminhos validados por um controlador de alta performance em três situações de estacionamento distintas.

Três métodos distintos são aplicados:

Sequência de Movimentos (MSM). Com as dimensões do veículo, da vaga e dos obstáculos conhecidos, define-se uma sequência empírica de movimentos para o estacionamento, emulando a forma convencional que um humano faria e servindo de referência para outros métodos;

Parametrização Espacial (MPE). O planejamento é feito pela parametrização de caminhos no espaço, especificando um movimento coordenado, contínuo e suave entre as posições inicial e final (Argento et al., 2020).

Algoritmo Genético (AG). O caminho é definido usando uma técnica de busca de inteligência computacional denominada algoritmo genético.

2. FORMULAÇÃO DO PROBLEMA

2.1 Modelo cinemático do veículo

O modelo cinemático utilizado é tal como apresentado em Vieira et al. (2022). As equações de movimento são obtidas com base na geometria de Ackerman (Figura 1), assumindo que as rodas dianteiras do CLMR giram de forma ligeiramente diferencial e, portanto, que o centro de rotação instantâneo do veículo pode ser computado por meios exclusivamente cinemáticos, uma aproximação suficiente para movimentação em baixas velocidades.

Assumindo-se terreno plano, o espaço de configurações do veículo é composto pelas variáveis de posição e orientação globais, (x, y, θ) , enquanto o ângulo de esterçamento (ϕ) e a velocidade (V) do mesmo são as variáveis do espaço de atuação. O mapeamento deste último para o espaço de configurações é obtido a partir das equações de Fresnel.

Os valores utilizados para parametrizar o modelo utilizado neste trabalho são apresentados na Tabela 1.

2.2 Modelagem do espaço de navegação

O veículo navega em um espaço modelado com um mapa geométrico (Thrun, 2003), que inclui obstáculos imóveis. Um caminho é projetado e avaliado em quatro zonas de detecção para verificar se está livre de obstáculos (Argento et al., 2020; Vieira et al., 2022).

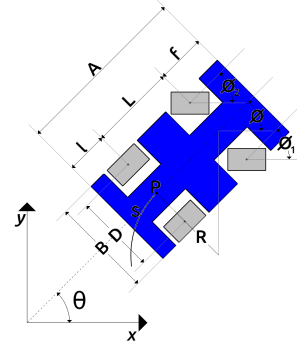


Figura 1. Modelo cinemático do veículo.

Tabela 1. Parâmetros do modelo

| Parâmetro | Descrição |
|--------------|---|
| ϕ | Ângulo de esterçamento do veículo |
| $\dot{\phi}$ | Velocidade angular de esterçamento |
| $\phi_{1,2}$ | Ângulo de esterçamento da roda 1 e 2 |
| A | Comprimento do veículo |
| B | Largura do veículo |
| D | Distância entre as rodas do eixo dianteiro |
| 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 |
| P | Ponto de referência entre eixo traseiro |
| R | Raio de curvatura instantâneo |
| θ | Ângulo do veículo em relação ao eixo x |
| V | Velocidade linear |

3. PLANEJAMENTO DE CAMINHOS

O planejamento de caminho consiste em levar um sistema de uma configuração inicial para uma configuração final. É um problema importante no desenvolvimento de robôs autônomos, envolvendo raciocínio automatizado, percepção e controle. O objetivo é conduzir o veículo de forma suave e contínua, evitando obstáculos durante as manobras de estacionamento.

3.1 Sequência de Movimentos (MSM)

Este método busca emular o comportamento de um humano ao realizar manobras para estacionar um automóvel. O caminho é o composto pela junção de curvas S (Roth, 2007), com base nas Equações do modelo cinemático. A Figura 2 apresenta as curvas necessárias para representar as formas de estacionamento paralelo, perpendicular e inclinado, assim como os parâmetros que as definem. Assumindo a entrada do estacionamento como referência do plano cartesiano e partindo dos pontos iniciais (x_i, y_i) e finais (x_f, y_f) e dos pontos intermediários para cálculo do raio de curvatura das curvas que compõem o caminho (x_1, y_1) e (x_2, y_2) , tem-se:

Estacionamento paralelo:

$$R_{\min} = \frac{L}{\tan(|\phi|_{\max})} \quad (1)$$

$$x_2 = x_f \quad (2)$$

$$y_2 = y_f + R_{\min} \quad (3)$$

$$y_1 = y_i - R_{\min} \quad (4)$$

$$x_1 = x_2 - \sqrt{-(y_1 - y_2)^2 + 4R_{\min}^2} \quad (5)$$

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

$$\alpha = \arccos \frac{5R_{\min}^2 - d^2}{4R_{\min}^2} \quad (7)$$

Estacionamento perpendicular:

$$x_2 = x_f - R_{\min} \quad (8)$$

$$y_2 = 0 \quad (9)$$

$$d = \sqrt{(x_i - x_2)^2 + (y_i - y_2)^2} \quad (10)$$

$$\beta = \arctan \frac{|y_i - y_2|}{|x_i - x_2|} + \frac{\pi}{2} \quad (11)$$

$$R = \frac{d^2 - R_{\min}^2}{2R_{\min} + 2d \cos \beta} \quad (12)$$

$$\alpha = \arccos \frac{R^2 + (R + R_{\min})^2 - d^2}{2R(R + R_{\min})} \quad (13)$$

$$x_1 = x_i \quad (14)$$

$$y_1 = y_i + R \quad (15)$$

Estacionamento inclinado:

$$\gamma = \arctan \left(\frac{|y_f|}{|x_f|} \right) + \frac{\pi}{4} \quad (16)$$

$$c_x = \sqrt{x_f^2 + y_f^2} \sin(\gamma) \cos \left(\frac{3\pi}{4} \right) \quad (17)$$

$$c_y = \sqrt{x_f^2 + y_f^2} \sin(\gamma) \sin \left(\frac{3\pi}{4} \right) \quad (18)$$

$$x_2 = R_{\min} \cos \left(\frac{5\pi}{4} \right) + c_x \quad (19)$$

$$y_2 = R_{\min} \sin \left(\frac{5\pi}{4} \right) + c_y \quad (20)$$

$$d = \sqrt{(x_i - x_2)^2 + (y_i - y_2)^2} \quad (21)$$

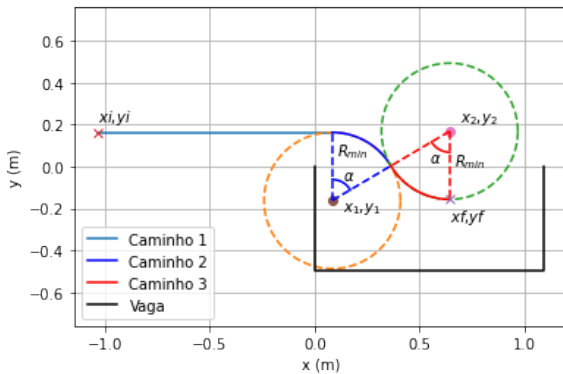
$$\beta = \frac{\pi}{2} - \arctan \frac{|y_i - y_2|}{|x_i - x_2|} \quad (22)$$

$$R = \frac{R_{\min}^2 - d^2}{2R_{\min} - 2d \cos \beta} \quad (23)$$

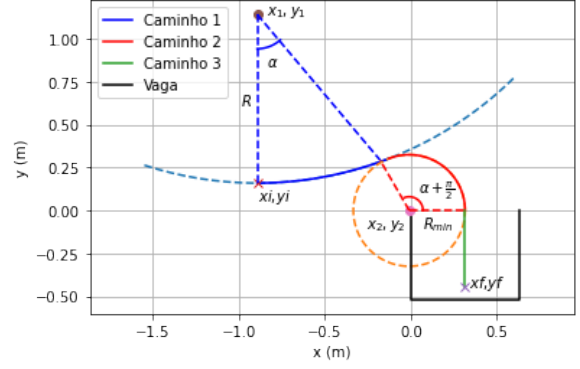
$$\alpha = \arccos \frac{R^2 + (R - R_{\min})^2 - d^2}{2R(R - R_{\min})} \quad (24)$$

$$x_1 = x_i \quad (25)$$

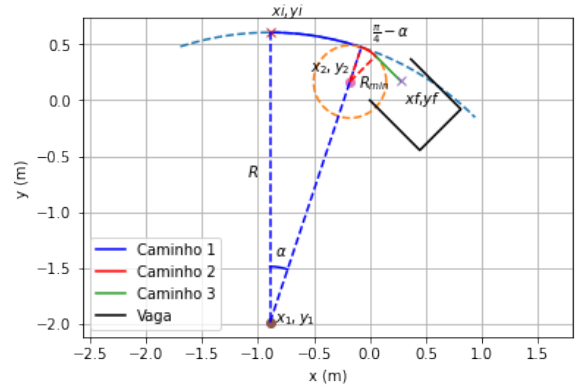
$$y_1 = y_i - R \quad (26)$$



(a) Paralelo.



(b) Perpendicular.



(c) Inclinado.

Figura 2. Composição das curvas para os estacionamentos paralelo, perpendicular e inclinado para método MSM.

3.2 Parametrização Espacial (MPE)

Os caminhos são planejados utilizando polinômios de quinto grau (Demirli and Khoshnejad, 2009) para duas variáveis de uma função espacial

$$f(s) = \sum_{i=0}^5 C_i s^i, \quad (27)$$

na qual s denota a posição do veículo no espaço bidimensional. O caminho é dado por

$$P(s) = (x(s), y(s)) \quad (28)$$

Impõe-se uma lei de movimento como segue:

$$s = s(t) \quad (29)$$

Assim, as velocidade e aceleração do veículo são:

$$\dot{P} = \frac{dP}{ds} \dot{s}(t) \quad (30)$$

$$\ddot{P} = \frac{dP}{ds} \ddot{s}(t) + \frac{d^2P}{ds^2} \dot{s}(t) \quad (31)$$

Os parâmetros de entrada são as posições $(x_0, y_0$ e $x_f, y_f)$ e orientações $(\theta_0$ e $\theta_f)$, iniciais e finais.

Com a parametrização temporal, tem-se:

$$\begin{aligned} \mathbf{x} &= [x_0 \ x_f \ V_s \cos(\theta_0) \ V_s \cos(\theta_f) \ -V_s \sin(\theta_0) \dot{\theta}_0 \ -V_s \sin(\theta_f) \dot{\theta}_f]^\top \\ \mathbf{y} &= [y_0 \ y_f \ V_s \sin(\theta_0) \ V_s \sin(\theta_f) \ V_s \cos(\theta_0) \dot{\theta}_0 \ V_s \cos(\theta_f) \dot{\theta}_f]^\top \\ \mathbf{T} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ k_f^5 & k_f^4 & k_f^3 & k_f^2 & k_f & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5k_f^4 & 4k_f^3 & 3k_f^2 & 2k_f & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20k_f^3 & 12k_f^2 & 6k_f & 2 & 0 & 0 \end{bmatrix} \\ \mathbf{c}_x &= [C_{x_5} \ C_{x_4} \ C_{x_3} \ C_{x_2} \ C_{x_1} \ C_{x_0}]^\top \\ \mathbf{c}_y &= [C_{y_5} \ C_{y_4} \ C_{y_3} \ C_{y_2} \ C_{y_1} \ C_{y_0}]^\top \end{aligned}$$

Os vetores \mathbf{x} e \mathbf{y} contêm os valores de entrada, enquanto a matriz com as condições temporais é denominada \mathbf{T} , e os vetores dos coeficientes de parametrização são denotadas por \mathbf{c}_x (dimensão x) e \mathbf{c}_y (dimensão y). Para obter o caminho desejado a partir das entradas disponíveis, calcula-se:

$$\mathbf{c}_x = \mathbf{T}^{-1} \times \mathbf{x} \quad (32)$$

$$\mathbf{c}_y = \mathbf{T}^{-1} \times \mathbf{y} \quad (33)$$

A escolha dos caminhos factíveis é feita variando-se valores dos parâmetros de projeto. A constante k_f é definida satisfazendo a equação:

$$k_f \geq \sqrt{(x_f - x_0)^2 + (y_f - y_0)^2} \quad (34)$$

A velocidade angular ($\dot{\theta}$) é definida de tal forma que:

$$\frac{\tan(\phi_{\min})}{L} \leq \dot{\theta}_0 \quad , \quad \dot{\theta}_f \leq \frac{\tan(\phi_{\max})}{L} \quad (35)$$

A constante V_s determina o sentido da velocidade do veículo.

A partir das posições e orientações iniciais e finais, gera-se um caminho que leve o veículo de uma a outra, atendendo aos limites definidos para a curva. A cinemática inversa determina, então, a velocidade (V) e o ângulo de esterçamento (ϕ) em função do tempo.

O gerador de caminhos é um programa fundamentado em algoritmo de força bruta. Trata-se de um algoritmo de fácil implementação, que enumera de maneira simples as possibilidades de uma solução e verifica se cada uma satisfaz à solução do problema (Ziviani, 2010). Diante das características do veículo e a presença de obstáculos no ambiente, o algoritmo seleciona o caminho de menor comprimento, caso exista. Para a definição dos campos de busca, a constante k_f tem seu início definido pela Equação (34) em um índice de incremento k_{f_i} ($k_{f_i} = \frac{1}{10}$). As velocidades angulares inicial (θ_0) e final (θ_f), têm seus limites definidos pela Equação (35), em um índice de incremento $\dot{\theta}_i$ ($\dot{\theta}_i = \frac{|\dot{\theta}_{\max}|}{10}$). Por fim, o sentido da velocidade pode assumir dois valores, na qual, $V_s = 1$ e $V_s = -1$ representam caminhos com movimentação para frente e para trás, respectivamente.

3.3 Algoritmo Genético (AG)

Inspirado no método de parametrização espacial, o planejamento de caminhos por AG é realizado utilizando o mesmo polinômio de quinto grau, porém tendo como parâmetros de entrada as posição (x_0 e y_0) e orientação

(θ_0) iniciais, as posição (x_f e y_f) e orientação (θ_f) finais, e a posição de um ponto auxiliar (x_1 e y_1).

$$\begin{aligned} \mathbf{x} &= [x_0 \ x_f \ V_s \cos(\theta_0) \ V_s \cos(\theta_f) \ x_1]^\top \\ \mathbf{y} &= [y_0 \ y_f \ V_s \cos(\theta_0) \ V_s \sin(\theta_f) \ y_1]^\top \\ \mathbf{T} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ k_f^5 & k_f^4 & k_f^3 & k_f^2 & k_f & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5k_f^4 & 4k_f^3 & 3k_f^2 & 2k_f & 1 & 0 \\ k_1^5 & k_1^4 & k_1^3 & k_1^2 & k_1 & 1 \end{bmatrix} \end{aligned}$$

Para variar os caminhos possíveis e encontrar a melhor, variam-se alguns parâmetros de entrada. Cada curva é avaliada e selecionada em função dos limites físicos e estruturais estabelecidos, incluindo o desvio dos eventuais obstáculos. A constante k_f é definida de acordo com a Equação (34). A constante k_1 deve ser maior que zero e diferente de k_f .

O ponto intermediário (x_1, y_1) é definido no intervalo $x_0 < x_1 < x_f$ e $y_0 < y_1 < y_f$ e ocorre entre os pontos inicial e final ($k_1 < k_f$) ou depois do último ($k_1 > k_f$), porém a curva é descrita apenas até o ponto final (x_f e y_f).

A velocidade do veículo pode assumir dois sentidos $V_s = 1$ e $V_s = -1$ denotam movimento para frente e para trás, respectivamente.

O gerador de caminhos utiliza o Algoritmo Genético para Otimização de Funções proposto por R. Houck et al. (Houck et al., 1995). Trata-se de um programa de implementação que, de maneira acessível, diante das condições e limitações das entradas, chamados de indivíduos, apura a curva de melhor aptidão arbitrada dentro das gerações.

Principiando as posições e poses iniciais e finais, o algoritmo exerce a escolha dos parâmetros que moldam o caminho, a saber: o ponto intermediário (x_1, y_1); as constantes k_f e k_1 e o sentido da velocidade (V_s).

Cada caminho gerado é submetido a função objetivo (f_o) definida pela Equação (36), e o AG busca o menor valor possível.

$$f_o = \sqrt{(|S|)^2 + (|\phi|_{\max})^2}, \quad (36)$$

na qual:

$$|S| = \int (\dot{x}^2 + \dot{y}^2) dt \quad (37)$$

Para tratar outras restrições, como evitar obstáculos, são utilizadas técnicas de penalização (Reeves, 1997), isto é, caso alguma outra restrição não seja respeitada, o algoritmo suprime o resultado natural de algum parâmetro e o substitui por um valor inadaptado. Neste caso, o comprimento do caminho ($|S|$) assume um valor cem vezes maior.

Ao decorrer de todas as gerações, se a curva escolhida atender aos requisitos, a mesma se torna válida.

4. CONTROLADOR DE PERSEGUIÇÃO PURA (PP)

O controlador de perseguição pura é um algoritmo de rastreamento de caminho, baseado na geometria da cinemática do veículo (Coulter, 1992). Adotando como referência

o modelo cinemático tipo bicicleta e dado um Ponto de Meta (P_m), definem-se respectivamente l_d e α como a distância e a inclinação entre ele e o veículo, como mostra a Figura 3.

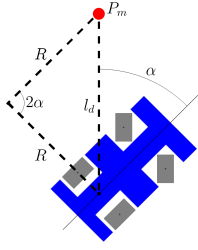


Figura 3. Geometria do controlador de perseguição pura.

Definindo-se o raio de curvatura (R), chega-se ao triângulo isósceles presente na Figura 3 e tem-se a seguinte relação:

$$R = \frac{l_d}{2\text{sen}(\alpha)} \quad (38)$$

Correlacionando com o modelo cinemático, o ângulo de direção (ϕ) pode ser calculado, como:

$$\phi = \arctan\left(\frac{2L\text{sen}(\alpha)}{l_d}\right), \quad (39)$$

na qual L é a distância entre os eixos do veículo.

O objetivo é controlar o veículo em todos os pontos do caminho de forma sequencial, partindo de um ponto inicial até o último ponto da rota. No entanto, como normalmente esta sequência predefinida é descrita por pontos muito próximos entre si, é comum haver dificuldade em alcançar o ponto seguinte em casos de curvas mais acentuadas ou ao se recuperar de uma perturbação, quando o ângulo final do veículo tende a não ser muito próximo da inclinação natural do caminho. Pelo exposto, busca-se adaptar o algoritmo para torná-lo mais robusto. Implementa-se uma Distância de Antecipação (L_{DA}), definida como a distância mínima necessária para que o algoritmo possa se antecipar para o próximo ponto de meta e, com isso, suavizar a curva necessária e estabilizar a rota.

5. RESULTADOS

Os resultados apresentados consideram os parâmetros que constam na Tabela 2. As vagas têm suas dimensões definidas em função das características do veículo (Vorobieva et al., 2015) e seus dados são apresentados na Tabela 3. A abordagem é baseada no ambiente de simulação PyBullet (Coumans and Bai, 2016), um módulo Python focado em simulação física de sistemas que aproxima a emulação computacional de resultados experimentais (Michalik and Janota, 2020).

Na Figura 4 pode-se observar o resultado do estacionamento paralelo de frente, pelo método MPE simulado no ambiente PyBullet com o controlador Pure Pursuit, e nas Figuras 5 e 6 podem ser observados os caminhos para todas as formas de estacionamento consideradas. A Tabela 4 apresenta a distância percorrida e o tempo efetuado para cada tipo manobra, através de cada método. Em mais

Tabela 2. Parâmetros do veículo

| Parâmetro | Valor |
|-----------------------|----------------------|
| ϕ | $-45^\circ/45^\circ$ |
| $ \dot{\phi} _{\max}$ | 1 rad/s |
| A | 475 mm |
| B | 290 mm |
| f | 100 mm |
| l | 50 mm |
| L | 325 mm |
| $ V _{\max}$ | 0,1 m/s |
| $ a _{\max}$ | 0,5 m/s ² |

Tabela 3. Dados das Vagas

| Tipo de vaga | Comprimento | Largura |
|----------------------------|-------------|----------|
| Paralelo de frente | 1091,8 mm | 495,6 mm |
| Paralelo de ré | 729,9 mm | 335,1 mm |
| Perpendicular de frente/ré | 631,7 mm | 519,2 mm |
| Inclinado de frente/ré | 631,7 mm | 519,2 mm |

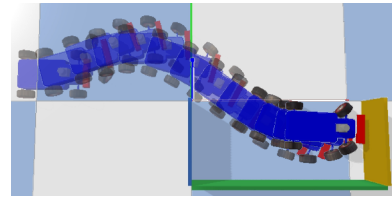


Figura 4. Estacionamento paralelo de frente, planejado pelo método MPE e seguido pelo controlador de perseguição pura no ambiente de simulação PyBullet.

Tabela 4. Resultados

| Tipo de vaga | MSM | MPE | AG |
|-------------------------|-------------------|--------------------------|--------------------------|
| Paralelo de frente | 1,793 m - 17,87 s | 1,830 m - 18,20 s | 1,768 m - 17,59 s |
| Paralelo de ré | 0,834 m - 8,41 s | 0,805 m - 8,09 s | 0,802 m - 8,07 s |
| Perpendicular de frente | 1,446 m - 14,49 s | 1,426 m - 14,28 s | 1,467 m - 14,71 s |
| Perpendicular de ré | 1,868 m - 18,65 s | 1,679 m - 16,75 s | 1,645 m - 16,39 s |
| Inclinado de frente | 1,298 m - 12,89 s | 1,255 m - 12,48 s | 1,283 m - 12,74 s |
| Inclinado de ré | 1,652 m - 16,35 s | 1,607 m - 15,95 s | 1,655 m - 16,37 s |

detalhes, o ¹ [link](https://www.youtube.com/watch?v=3AZmfy57NqA) apresenta o vídeo da execução de cada método para todos os tipos de estacionamento.

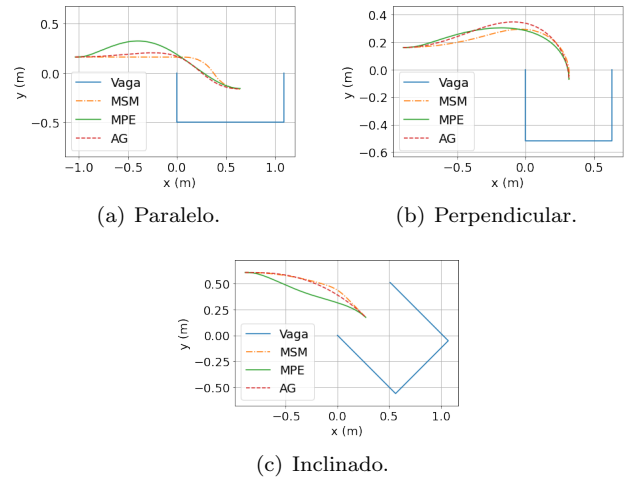


Figura 5. Comparação entre os caminhos planejados por cada método para o estacionamento de frente.

Observa-se que todos os métodos conseguem efetuar diferentes tipos de estacionamento de forma contínua e su-

¹ <https://www.youtube.com/watch?v=3AZmfy57NqA>

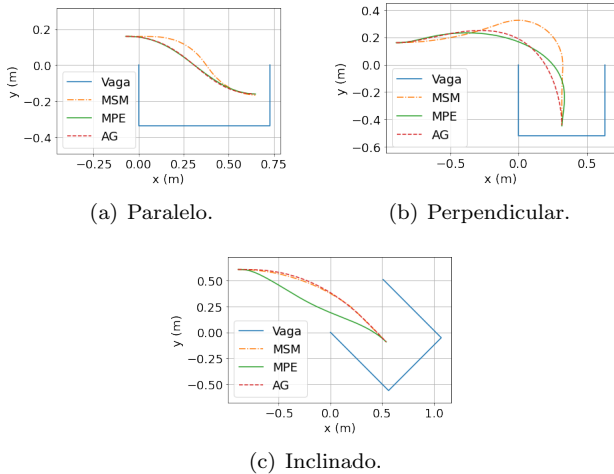


Figura 6. Comparação entre os caminhos planejados por cada método para o estacionamento de ré.

ave, desviando dos obstáculos definidos pelas vagas. Os métodos que utilizam referências polinomiais e algoritmos de busca tiveram o melhor desempenho no que tange aos critérios de menor distância e tempo de execução dos caminhos. O MPE se destaca em 50% das situações avaliadas, enquanto o AG desempenhou melhor nas outras situações. Como esperado, ambos superam o MSM, demonstrando, portanto, sua superioridade com relação à condução humana convencional ao realizar manobras de estacionamento. Outro destaque dado aos algoritmos MPE e AG são suas aplicabilidades, pois uma única abordagem foi suficiente para ter êxito em diferentes cenários.

6. CONCLUSÕES

Neste trabalho, são comparados três planejadores de caminhos para estacionamento autônomo de carros, usando o modelo geométrico de Ackermann para evitar derrapagens. Os resultados da simulação computacional em um ambiente físico robótico demonstraram a viabilidade dos algoritmos propostos e a superioridade dos caminhos de movimento contínuo em relação aos movimentos humanos usuais ao estacionar. Dentre os algoritmos avaliados, MPE e AG tiveram os melhores desempenhos. Propõe-se investigações futuras sobre o desempenho destes métodos para diferentes tamanhos de vagas, desenvolvimento de um modelo dinâmico e outras técnicas de otimização, além da validação experimental.

AGRADECIMENTOS

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores também agradecem à Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro - FAPERJ - pelo seu apoio no desenvolvimento deste trabalho.

REFERÊNCIAS

Argento, E.V., Vieira, R.P., and Revoredo, T.C. (2020). Estacionamento paralelo autônomo de veículos leves de passeio. *Anais da Sociedade Brasileira de Automática*, 2(1).

- Coulter, R.C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical report, Carnegie-Mellon UNIV Pittsburgh PA Robotics INST.
- Coumans, E. and Bai, Y. (2016). Pybullet, a python module for physics simulation for games, robotics and machine learning. URL <http://pybullet.org>.
- Demirli, K. and Khoshnejad, M. (2009). Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller. *Fuzzy Sets and Systems*, 160(19), 2876–2891. doi:10.1016/j.fss.2009.01.019.
- Erkorkmaz, K. and Altintas, Y. (2001). High speed CNC system design. part i: jerk limited trajectory generation and quintic spline interpolation. *International Journal of Machine Tools and Manufacture*, 41(9), 1323–1345. doi:10.1016/S0890-6955(01)00002-5.
- Houck, C.R., Joines, J., and Kay, M.G. (1995). A genetic algorithm for function optimization: a matlab implementation. *Ncsu-ie tr*, 95(09), 1–10.
- Hussain, R. and Zeadally, S. (2019). Autonomous cars: Research results, issues, and future challenges. *IEEE Communications Surveys & Tutorials*, 21(2), 1275–1313. doi:10.1109/comst.2018.2869360.
- Michalik, R. and Janota, A. (2020). The PyBullet module-based approach to control the collaborative YuMi robot. In *2020 ELEKTRO*. IEEE. doi:10.1109/elektro49696.2020.9130233.
- Ni, S., Liu, Z., Cai, Y., and Wang, X. (2018). Modelling of ship's trajectory planning in collision situations by hybrid genetic algorithm. *Polish Maritime Research*, 25(3), 14–25. doi:10.2478/pomr-2018-0092.
- Rashid, M.M., Rahman, M.M., Islam, M.R., Alwahedy, O.N., and Abdullahi, A. (2019). Autonomous 4wd smart car parallel self-parking system by using fuzzy logic controller. *American International Journal of Sciences and Engineering Research*, 2(2), 1–31.
- Reeves, C.R. (1997). Genetic algorithms for the operations researcher. *INFORMS journal on computing*, 9(3), 231–250.
- Roth, J. (2007). *Einparken-ein ideales thema für experimentelle geometrie und projektarbeit*. Universitätsbibliothek Dortmund.
- Thrun, S. (2003). Robotic mapping: a survey. *Exploring Artificial Intelligence in the NewMillenium*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1–35.
- Vieira, R., Argento, E., and Revoredo, T. (2022). Trajectory planning for car-like robots through curve parametrization and genetic algorithm optimization with applications to autonomous parking. *IEEE Latin America Transactions*, 20(2), 309–316. doi:10.1109/tla.2022.9661471.
- Vorobieva, H., Glaser, S., Minoiu-Enache, N., and Mammari, S. (2015). Automatic parallel parking in tiny spots: Path planning and control. *IEEE Transactions on Intelligent Transportation Systems*, 16(1), 396–410.
- Wang, D., Liang, H., Mei, T., and Zhu, H. (2011). Research on self-parking path planning algorithms. In *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*. IEEE. doi:10.1109/icves.2011.5983825.
- Ziviani, N. (2010). Projeto de algoritmos com implementações em pascal e c. São Paulo: Pioneira Thomson Learning, 3ª Edição.