

Trajectory Planning For Car-like Robots Through Curve Parametrization And Genetic Algorithm Optimization With Applications To Autonomous Parking

Renan P. Vieira, Eduardo V. Argento and Téo C. Revoredo, IEEE Member

Resumo—Parallel parking a car is a difficult task and may be frustrating and stressful for the driver, while commonly causes traffic jam. One way to mitigate such negative effects is to provide vehicles with self-driving capabilities. As a cornerstone of a mobile robot's ability to move autonomously stands trajectory planning, which despite many works in the literature, is still considered an open problem especially with regards to nonholonomic vehicles such as car-like robots. Based on this scenario, this work presents a trajectory planning algorithm to parallel park car-like mobile robots based on polynomial parametrization and genetic algorithm optimization. The aim is to define a law of motion to lead the vehicle from an initial pose near a parking space to a final pose within the latter in a smooth way, with no interruption and avoiding any obstacles in the way. Simulation results in 3D sim-to-real type environment are presented to validate the feasibility of the proposed algorithm, which lay the foundation to broader studies.

Index Terms—Car-like mobile robots, Trajectory generation, Parallel parking, Genetic algorithm.

I. INTRODUÇÃO

Estacionar um carro nem sempre é uma tarefa fácil e com frequência torna-se algo estressante e frustrante para o condutor. Em grandes centros urbanos, motoristas podem levar até 8 minutos para estacionar seu veículo, contribuindo de 30 a 40% para o congestionamento do tráfego [1]. Não por acaso, sistemas de estacionamento autônomo vêm sendo implementados em veículos comerciais por diversos fabricantes. Nesses sistemas, conhecidos como manobristas automatizados (do inglês *Automated Valet Parking*), o veículo desloca-se de maneira autônoma de uma pose inicial a uma pose final especificada dentro do espaço de estacionamento disponível [2]. Trata-se de um caso particular de planejamento e seguimento de trajetórias aplicado a um sistema com restrições não-holonômicas, o que insere dificuldades na solução do problema e na implementação de controladores automáticos eficientes [3], [4]. Sob essa óptica, pode-se afirmar que é um problema ainda em aberto e muitos estudos vêm sendo realizados para desenvolver algoritmos eficazes e ótimos [5]–[7].

Renan P. Vieira, Universidade do Estado do Rio de Janeiro, RJ, Brasil, e-mail: renan.porto.vieira@gmail.com

Eduardo V. Argento, Pontifícia Universidade Católica do Rio de Janeiro, RJ, Brasil e-mail: eduardoverasargento@gmail.com

Téo C. Revoredo, Universidade do Estado do Rio de Janeiro, RJ, Brasil e-mail: teorevoredodo@uerj.br

Manuscript received Month Day, Year; revised Month day, Year.

Nesse contexto, muitas técnicas vêm sendo investigadas para planejar trajetórias a serem executadas de maneira autônoma por robôs móveis do tipo carro (CLMR, do inglês *car-like mobile robots*), dentre as quais citam-se a interpolação [8], [9] e os algoritmos de inteligência artificial (IA) [10], [11], aplicados tanto de maneira individual quanto conjunta.

D. Kim et al. [12] usa uma curva circular-linear para projetar um caminho de estacionamento, porém a descontinuidade na curvatura aumenta a dificuldade do seu rastreamento. K. Demirli et al. [13] usa uma função polinomial de quinta ordem para projetar a manobra de estacionamento, garantido a continuidade da curvatura do caminho. No entanto, é necessário evitar a colisão entre o veículo e os objetos ao redor através de um controlador baseado em sensores.

D. Wang et al. [14] usa um método de dois arcos para modelar o caminho de estacionamento paralelo e otimizar seu comprimento total, entretanto, a trajetória é limitada a dois seguimentos existindo um ponto de parada ao longo do caminho. Em K. Erkorkmaz et. al. [15], o autor combina interpolação de funções de terceira ordem com otimização por enxame de partículas para gerar trajetórias contínuas e suaves que garantem alta exatidão de rastreamento enquanto evitam modos naturais da estrutura mecânica ou do sistema de servo-controle.

Entre as técnicas de IA aplicadas ao problema de movimentação autônoma de robôs móveis, os algoritmos genéticos (AG) estão entre as mais promissoras, sendo passíveis de aplicação direta no processo de planejamento e seguimento de trajetórias [16], [17] ou de combinação com outras classes de algoritmos para melhorar sua eficiência [18].

Este trabalho visa contribuir para o problema de planejamento de trajetórias para CLMRs, propondo um algoritmo partindo do nível 2 de condução autônoma [19] cujo propósito é conduzir o veículo de maneira suave e contínua de uma pose inicial a uma pose final, desviar de eventuais obstáculos pelo caminho e realizar uma trajetória de menor comprimento, respeitando os limites do veículo. A proposta combina as vantagens da interpolação polinomial com a capacidade de otimização dos algoritmos genéticos para gerar trajetórias ininterruptas, exequíveis e otimizadas com relação à distância percorrida, contribuindo especialmente em termos de implementação na associação das diferentes técnicas. Como foco de aplicação, aborda-se o estacionamento paralelo autônomo e apresenta-se um exemplo simples de estacionamento em uma

garagem com obstáculos no caminho, onde o veículo deve se dirigir próximo à vaga e efetuar o estacionamento paralelo. Os resultados apresentados são emulados em plataforma de simulação física que confere maior realismo e permite avaliar de maneira efetiva o desempenho da implementação proposta, provendo as bases para melhorias futuras.

O trabalho é organizado da seguinte maneira: a Seção II apresenta a modelagem cinemática do veículo, a Seção III discorre sobre o modelo utilizado para representar o ambiente onde o veículo pode se deslocar, incluindo a representação dos obstáculos. Na Seção IV é apresentado o algoritmo proposto para o planejamento de trajetórias e a Seção V apresenta resultados de validação computacional para a movimentação no ambiente, bem como para o estacionamento do veículo. Na Seção VI são realizadas considerações finais sobre o trabalho.

II. MODELO CINEMÁTICO DO VEÍCULO

O planejamento das trajetórias a serem desempenhadas pelo veículo é feito com base na sua cinemática, sendo a dinâmica desconsiderada, algo válido em implementações de baixas velocidades tais como o estacionamento paralelo [20]. O veículo é modelado como um corpo rígido contendo quatro rodas em contato com o solo, sendo a sua orientação determinada pelas rodas dianteiras, tal como apresentado na Fig. 1.

As equações cinemáticas são obtidas com base no modelo de bicicleta [21], [22]. O referencial do veículo tem origem no centro do eixo traseiro, ponto este denominado ponto de guiamento e denotado por P na Fig. 1. A escolha deste ponto se dá por serem suas coordenadas cartesianas uma saída diferencialmente plana para o sistema, ou seja, uma vez especificadas estas variáveis e suas derivadas de segunda ordem, todas as outras variáveis de estado e a entrada do sistema são também definidas [23].

O raio de curvatura (R) da trajetória é definido pela Equação (1), na qual L é a distância entre os eixos do veículo e ϕ é o seu ângulo de esterçamento. As especificações destes e dos outros parâmetros do CLMR utilizados neste trabalho são resumidos na Tabela I. Cabe ressaltar que o ângulo de esterçamento (ϕ) é limitado, como mostra a Equação (2).

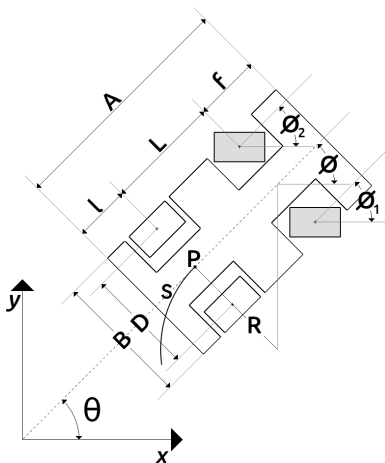


Fig. 1. Modelo cinemático do veículo.

$$R = \frac{L}{\tan(\phi)} \quad (1)$$

$$\phi_{\min} \leq \phi \leq \phi_{\max} \quad (2)$$

TABELA I
PARÂMETROS DA CINEMÁTICA DO VEÍCULO

Parâmetro	Descrição
ϕ	Ângulo de esterçamento do veículo
ϕ_1	Ângulo de esterçamento da roda 1
ϕ_2	Ângulo de esterçamento da roda 2
$\dot{\phi}$	Velocidade angular de esterçamento
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 ao eixo x
V	Velocidade linear
\dot{V}	Aceleração do veículo
P	Ponto de referência entre eixo traseiro

A distância (S) percorrida pelo ponto de referência (P) do veículo e a sua derivada (V) são definidas por (3) e (4).

$$S = R \cdot \theta \quad (3)$$

$$\dot{\theta} = \frac{V}{R} \quad (4)$$

Sendo a decomposição da velocidade, no plano x e y determinadas por (5) e (6).

$$\dot{x}(t) = V \cdot \cos(\theta(t)) \quad (5)$$

$$\dot{y}(t) = V \cdot \sin(\theta(t)) \quad (6)$$

A orientação do veículo (θ) e a sua derivada podem ser expressas em função de x e y , são apontadas nas Equações (7) e (8).

$$\theta(t) = \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right) \quad (7)$$

$$\dot{\theta}(t) = \frac{\dot{x} \cdot \ddot{y} - \ddot{x} \cdot \dot{y}}{\dot{x}^2 + \dot{y}^2} \quad (8)$$

Assim, a velocidade linear do CLMR e o seu ângulo de esterçamento são definidos por (9) e (10).

$$V = \pm \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} \quad (9)$$

$$\phi = \arctan\left(\frac{L\dot{\theta}}{V}\right) \quad (10)$$

Ainda que as rodas sejam situadas em um mesmo eixo, a distância entre elas implica em raios de curvatura diferentes para um determinado ângulo de condução do veículo. Deve-se, portanto, adaptar o modelo cinemático para que represente adequadamente a movimentação do CLMR em curvas, aproximando de maneira exata o seu comportamento real [24], [25]. A solução adotada utiliza a geometria de Ackermann para o sistema de esterçamento [26], a qual especifica ângulos distintos para as rodas dianteiras, ϕ_1 e ϕ_2 , dados pela Equação (11).

$$\phi_1, \phi_2 = \arctan\left(\frac{L}{R \pm \frac{D}{2}}\right) \quad (11)$$

De maneira semelhante, para que o veículo se movimente sem que haja deslizamento das rodas, estas devem ser capazes de assumir velocidades diferentes entre si [27]. As velocidades das rodas traseiras direita e esquerda, V_{td} e V_{te} , respectivamente, e das rodas dianteira direita e esquerda, V_{fd} e V_{fe} , são determinadas pelas Equações (12), (13) e (14).

$$V_{td}, V_{te} = V \cdot \left(1 \pm \frac{D}{2 \cdot R}\right) \quad (12)$$

$$V_{fd} = \frac{V_{td}}{\cos(\phi_1)} \quad (13)$$

$$V_{fe} = \frac{V_{te}}{\cos(\phi_2)} \quad (14)$$

III. MODELAGEM DO AMBIENTE

O ambiente no qual o veículo se movimenta é representado por um mapa geométrico [28], compondo uma abstração de alto nível em uma abordagem de fácil interpretação, implementação computacional simples e representação consistente.

A. Presença de obstáculos fixos

Para detectar obstáculos no ambiente onde o veículo navega, regiões de detecção são definidas com base em quatro pontos de referência [29], tal como exemplificado na Fig. 2. As zonas

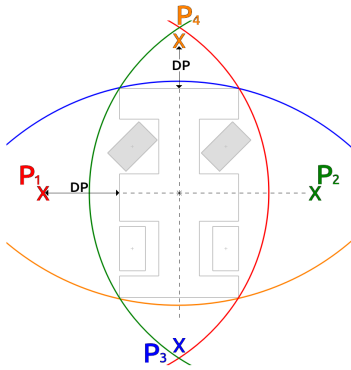


Fig. 2. Zona de detecção de obstáculos.

de detecção são circunferências, centralizadas nos pontos de referência (P_1 , P_2 , P_3 e P_4), cujos raios são definidos por (15) e (16).

$$r_1 = r_2 = \sqrt{(DP + B)^2 + \left(\frac{A}{2}\right)^2} \quad (15)$$

$$r_3 = r_4 = \sqrt{(DP + A)^2 + \left(\frac{B}{2}\right)^2} \quad (16)$$

nas quais A e B denotam o comprimento e a largura do veículo, respectivamente, e DP a distância entre este e ponto de referência correspondente.

As coordenadas no plano (x e y) dos pontos de referência são dadas pela Equação (17).

$$P_n = (P_{nx}, P_{ny}), \quad n = 1, 2, 3, 4 \quad (17)$$

Dessa forma, a distância entre qualquer obstáculo \mathbf{O} , onde $o \in \mathbf{O}$, $o = (o_x, o_y)$, e os pontos de referência de detecção (P_n) pode ser calculada através da Equação (18).

$$d_{P_n o} = \sqrt{(P_{nx} - o_x)^2 + (P_{ny} - o_y)^2} \quad (18)$$

$$n = 1, 2, 3, 4$$

Portanto, para que seja livre de obstáculos, uma trajetória deve satisfazer a Equação (19).

$$d_{P_n o} > r_n, \quad n = 1, 2, 3, 4 \quad (19)$$

A zona de detecção, a qual o veículo pertence, é formada pela interseção das quatro circunferências que, para que sejam tangentes ao veículo, devem assumir raios de dimensões excessivas em função de DP , como mostra a Fig. 3. Por isso, o valor de DP adotado é de 10^9 m.

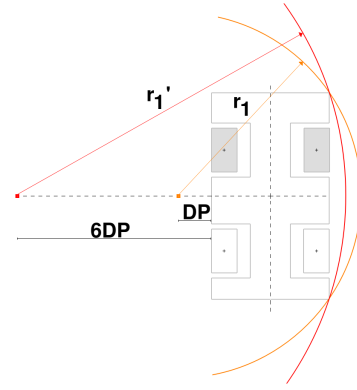


Fig. 3. Circunferência de detecção tangente ao veículo em função de DP .

IV. PLANEJAMENTO DE TRAJETÓRIAS

O objetivo do planejamento de trajetórias é determinar uma rota exequível no plano (x e y) para mover o veículo de uma pose inicial a uma pose final, evitando colisões com obstáculos presentes no ambiente e de forma contínua. Para tal, propõe-se aqui a utilização de funções polinomiais de quinta ordem [13], tal como definidos em (20) e (21), sendo C_{x_i} e C_{y_i} os seus coeficientes.

$$x(t) = \sum_{i=0}^5 C_{x_i} t^i \quad (20)$$

$$y(t) = \sum_{i=0}^5 C_{y_i} t^i \quad (21)$$

Tem-se então a posição do veículo no espaço bidimensional, $P(t)$, definida por (22).

$$P(t) = (x(t), y(t)) \quad (22)$$

No entanto, o sistema apresenta inconsistência de primeira ordem nas condições de contorno (iniciais e finais) [30]. Isso ocorre, pois a velocidade do veículo é nula nesses pontos, isto é, $\dot{x} = 0$ e $\dot{y} = 0$. Portanto, dadas as Equações (7), (9) e (10), é notório que os valores de orientação (θ), velocidade (V) e ângulo de esterçamento (ϕ) do veículo são indefinidos

nestes pontos. Com isso, não é possível especificar o conjunto completo de sinais de controle e os estados das trajetórias. Para evitar esses pontos onde as trajetórias são indefinidas, propõe-se a parametrização de trajetórias de forma que a descrição do caminho seja independente da informação do tempo [31]. Impõe-se, então, uma lei de movimento $s = s(t)$. Desse modo, a trajetória é descrita em termos desses parâmetros como $x(s)$ e $y(s)$. Logo $\dot{x}(t)$ e $\dot{y}(t)$ são preciso por (23) e (24).

$$\dot{x}(t) = \frac{dx(s)}{ds} \cdot \frac{ds}{dt} \quad (23)$$

$$\dot{y}(t) = \frac{dy(s)}{ds} \cdot \frac{ds}{dt} \quad (24)$$

Define-se s como uma variável normalizada em uma função polinomial no tempo que assume valores no intervalo $[0,1]$. O tempo (t), por sua vez, assume valores no intervalo $[0,t_f]$. O mapeamento entre s e t é representado através das Equações (25) e (26).

$$t = 0 \rightarrow s = 0 \quad (25)$$

$$t = t_f \rightarrow s = 1 \quad (26)$$

Assume-se s também como um polinômio de quinto grau, satisfazendo as condições iniciais ($t = 0$) e finais ($t = t_f$) de posição (s), velocidade (\dot{s}) e aceleração (\ddot{s}), da seguinte maneira:

$$s(0) = 0, s(t_f) = 1$$

$$\dot{s}(0) = 0, \dot{s}(t_f) = 0$$

$$\ddot{s}(0) = 0, \ddot{s}(t_f) = 0$$

Assim, considerando a descrição matricial do sistema, tem-se:

$$\mathbf{s} = [s(0) \quad s(t_f) \quad \dot{s}(0) \quad \dot{s}(t_f) \quad \ddot{s}(0) \quad \ddot{s}(t_f)]^T \quad (27)$$

$$\mathbf{T}_s = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ t_f^5 & t_f^4 & t_f^3 & t_f^2 & t_f^1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 5t_f^4 & 4t_f^3 & 3t_f^2 & 2t_f^1 & 1 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 20t_f^3 & 12t_f^2 & 6t_f & 2 & 0 & 0 \end{bmatrix} \quad (28)$$

$$\mathbf{c}_s = [C_{s_5} \quad C_{s_4} \quad C_{s_3} \quad C_{s_2} \quad C_{s_1} \quad C_{s_0}]^T \quad (29)$$

O vetor \mathbf{s} (Equação (27)) contém as condições iniciais e finais de posição, velocidade e aceleração, a matriz \mathbf{T}_s (Equação (28)) é composta pelas condições temporais e o vetor \mathbf{c}_s (Equação (29)) compreende os coeficientes do polinômio. Portanto, pode-se descrever $s(t)$ como mostra a Equação (30).

$$s(t) = \sum_{i=0}^5 C_{s_i} t^i \quad (30)$$

sendo as suas constantes (C_{s_i}) definidas pela Equação (31):

$$\mathbf{c}_s = \mathbf{T}_s^{-1} \mathbf{s} \quad (31)$$

Uma vez estabelecida a variável normalizada ($s(t)$), pode-se definir x e y como polinômios de quinta ordem em função de s , como é exposto pelas Equações (32) e (33).

$$x(s) = \sum_{i=0}^5 C_{x_i} s^i \quad (32)$$

$$y(s) = \sum_{i=0}^5 C_{y_i} s^i \quad (33)$$

Assim, submetendo-os às condições iniciais ($x(0)$, $y(0)$, $\theta(0)$), e finais ($x(1)$, $y(1)$, $\theta(1)$) e determinando as suas constantes (C_{x_i} e C_{y_i}), é então possível determinar a trajetória utilizando o mesmo método matricial.

Define-se a matriz \mathbf{T} pelas condições temporais em função de $s(t)$, como mostra a Equação (34), os vetores \mathbf{c}_x e \mathbf{c}_y , respectivamente, das constantes C_{x_i} e C_{y_i} , como mostram as Equações (35) e (36).

$$\mathbf{T} = \begin{bmatrix} s(0)^5 & s(0)^4 & s(0)^3 & s(0)^2 & s(0)^1 & 1 \\ s(t_f)^5 & s(t_f)^4 & s(t_f)^3 & s(t_f)^2 & s(t_f)^1 & 1 \\ 5s(0)^4 & 4s(0)^3 & 3s(0)^2 & 2s(0) & 1 & 0 \\ 5s(t_f)^4 & 4s(t_f)^3 & 3s(t_f)^2 & 2s(t_f) & 1 & 0 \end{bmatrix} \quad (34)$$

$$\mathbf{c}_x = [C_{x_5} \quad C_{x_4} \quad C_{x_3} \quad C_{x_2} \quad C_{x_1} \quad C_{x_0}]^T \quad (35)$$

$$\mathbf{c}_y = [C_{y_5} \quad C_{y_4} \quad C_{y_3} \quad C_{y_2} \quad C_{y_1} \quad C_{y_0}]^T \quad (36)$$

Para compor os vetores \mathbf{x} e \mathbf{y} , com as condições iniciais e finais, atribuem-se as derivadas de primeira ordem, como mostram as Equações (37), (38), (39) e (40).

$$\frac{dx(0)}{ds} = V_s k_0 \cos(\theta(0)) \quad (37)$$

$$\frac{dy(0)}{ds} = V_s k_0 \sin(\theta(0)) \quad (38)$$

$$\frac{dx(1)}{ds} = V_s k_1 \cos(\theta(1)) \quad (39)$$

$$\frac{dy(1)}{ds} = V_s k_1 \sin(\theta(1)) \quad (40)$$

onde $x(0)$, $y(0)$ e $\theta(0)$ correspondem às condições iniciais e $x(1)$, $y(1)$ e $\theta(1)$ às condições finais de posição e orientação. V_s indica o sentido do veículo, assumindo $V_s = 1$ para frente ou $V_s = -1$ para ré. Por último, considerando arbitrariamente duas constantes k_0 e k_1 e partindo-se da condição $k_0 = k_1 = 1$, pode-se então aplicar a metodologia à problemas de otimização [32].

Sendo assim, os vetores \mathbf{x} e \mathbf{y} são definidos por meio das Equações (41) e (42).

$$\mathbf{x} = \left[x(0) \quad x(1) \quad \frac{dx(0)}{ds} \quad \frac{dx(1)}{ds} \right]^T \quad (41)$$

$$\mathbf{y} = \left[y(0) \quad y(1) \quad \frac{dy(0)}{ds} \quad \frac{dy(1)}{ds} \right]^T \quad (42)$$

Por fim, as constantes C_{x_i} e C_{y_i} , das Equações (32) e (33), são definidas por (43) e (44).

$$\mathbf{c}_x = \mathbf{T}^{-1} \mathbf{x} \quad (43)$$

$$\mathbf{c}_y = \mathbf{T}^{-1} \mathbf{y} \quad (44)$$

A. Implementação computacional

A determinação das diferentes trajetórias, entre as possibilidades que a parametrização temporal provê, é realizada computacionalmente por um algoritmo denominado Gerador de Trajeto. Tendo como informações de entrada as posições e orientações iniciais e finais desejadas e as restrições impostas pela cinemática do veículo e pelo ambiente, o Gerador de Trajeto vale-se de um algoritmo genético (AG), associado à interpolação polinomial, para escolher a trajetória dentro dos critérios pré-estabelecidos.

Algoritmos Genéticos são métodos de otimização e busca inspirados nos mecanismos de evolução de populações de seres vivos [33]. São caracterizados por utilizar o princípio da seleção natural e sobrevivência do mais apto, declarado em 1859 pelo naturalista e fisiologista inglês Charles Darwin [34]. Esta classe de algoritmo permite a utilização de diferentes representações das variáveis de entrada, tais como reais e binárias, assim como diferentes níveis de precisão ou até mesmo restrições de acordo com a natureza do problema, formando os chamados "cromossomas" que baseiam um indivíduo. Um grupo de indivíduos, por sua vez, é conhecido como população.

Seguindo a inspiração citada anteriormente, o algoritmo busca soluções variando suas entradas de acordo com dois parâmetros de diferentes objetivos. Para valorizar a exploração do espaço de busca são aplicadas mutações individuais nos cromossomas, enquanto se busca unir as melhores características e obter melhores avaliações por meio do cruzamento entre eles. Por fim, a seleção natural conhecida pela teoria Darwiniana é representada por uma função responsável por selecionar os indivíduos que compõe a população seguinte.

Dentre as diversas implementações possíveis para a escolha dos mais aptos, o Gerador de Trajeto aqui implementado utiliza a proposta para otimização de funções feita por R. Houck et al. [35]. Trata-se de um algoritmo que avalia um grupo (população) de informações de entrada (indivíduos) e, com base nas restrições impostas, apura a curva de melhor aptidão, arbitrada dentro das gerações. No presente trabalho, além das constantes k_0 , k_1 , outro parâmetro de entrada importante é o sentido da velocidade, uma vez que o CLMR pode se movimentar para frente e para trás. Assim, o sentido da velocidade do veículo é dado por V_s , e assume valores 1 para movimentação para a frente e -1 para marcha a ré.

Diante da variação dos parâmetros que geram diferentes formas de trajetórias (k_0 , k_1 e V_s), cada trajetória é submetida à função objetivo (f_o), na qual o algoritmo genético deve encontrar o seu menor valor possível. Dadas as características do veículo, além de relacionar o comprimento ($|S|$) da trajetória, a função objetivo também considera o valor absoluto máximo do ângulo de esterçamento ($|\phi|_{\max}$) efetuado. Define-se, dessa maneira, a função objetivo pela Equação (45).

$$f_o = \sqrt{(|S|)^2 + (|\phi|_{\max})^2} \quad (45)$$

na qual:

$$|S| = \int (\dot{x}^2 + \dot{y}^2) dt \quad (46)$$

Para tratar outras restrições, tais como evitar obstáculos, são utilizadas técnicas de penalização [36], 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. Nesta implementação, o comprimento da trajetória ($|S|$) assume um valor cem vezes maior.

Uma vez definida a trajetória, é selecionado o menor tempo para a sua execução em função dos limites de velocidade (V), aceleração (\dot{V}) e variação do ângulo de esterçamento ($\dot{\phi}$) do CLMR. Para sintetizar o funcionamento do algoritmo, a Fig. 4 apresenta o fluxograma do Gerador de Trajeto.

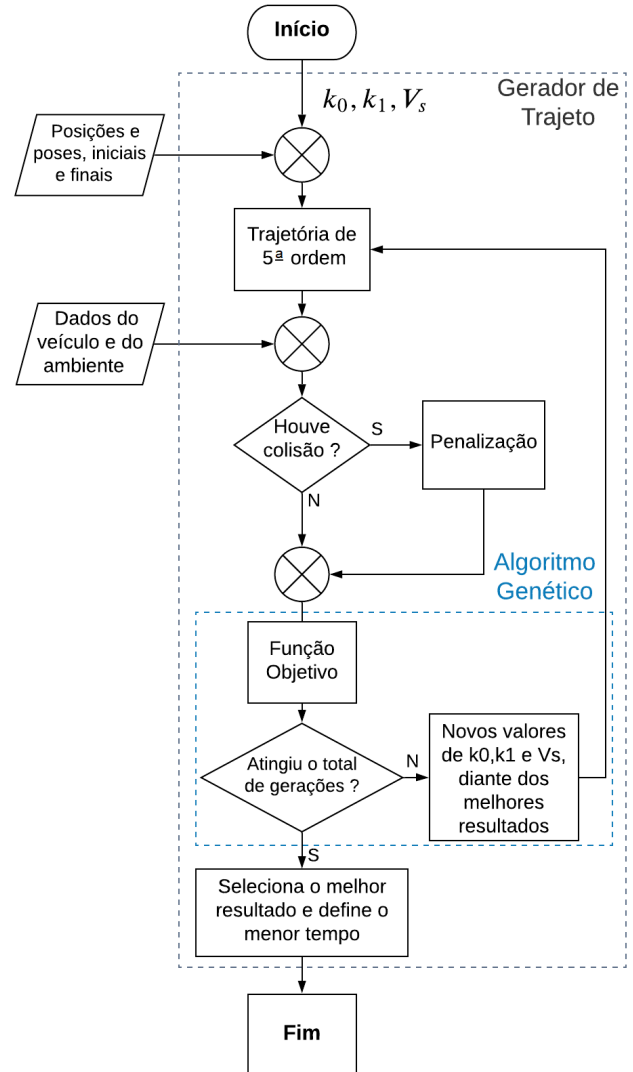


Fig. 4. Fluxograma do Gerador de Trajeto.

V. RESULTADOS

Para os resultados apresentados a seguir, as características consideradas para o CLMR são aquelas que constam na Tabela II. A Tabela III apresenta as especificações do AG aplicado. A abordagem é baseada no ambiente de simulação PyBullet [37]. Trata-se de um módulo Python para simulação física para robótica, jogos, efeitos visuais e aprendizagem de máquinas, com foco na transferência da simulação para o robô

real [38]. O ambiente permite uma variedade de simulações, incluindo, dinâmica, cinemáticas direta e inversa, e detecção de colisão.

TABELA II
PARÂMETROS DO VEÍCULO

Parâmetro	Valor
L	0,325 m
B	0,29 m
f	0,1 m
l	0,05 m
A	0,475 m
ϕ	$-33^\circ/33^\circ$
$ \dot{\phi} _{\max}$	1 rad/s
$ V _{\max}$	1 m/s
$ \dot{V} _{\max}$	0,5 m/s ²

TABELA III
PARÂMETROS DO AG

Parâmetro	Especificações
Representação	Binária
Domínio k_0	[1, 50]
Domínio k_1	[1, 50]
Domínio V_s	{-1, 1}
ϵ (Precisão)	10^{-8}
Aptidão	Avaliação
População	50
Gerações	100
Taxa de cruzamento	60%
Taxa de Mutação	4%
Função de Seleção	Seleção por roleta

Estacionar um CLMR paralelamente significa manobrá-lo para estacionar em posição paralela à estrada, em linha com outros veículos. Os resultados que seguem focam neste tipo de aplicação. A pose inicial do veículo é definida próxima a um espaço de estacionamento disponível e a pose final dentro da vaga, no extremo oposto, de tal maneira a prover o maior espaço de manobra.

A Fig. 5 descreve o arranjo para o estacionamento paralelo de frente e de ré. Os pontos I_f e I_r são as posições de partida dos modos de estacionamento de frente e de ré, respectivamente, descritos em função das dimensões do veículo. L é a distância entre eixos, l a distância entre o eixo traseiro e a traseira, e f é a extensão entre o eixo dianteiro e a frente. A posição final na vaga é estabelecida como aquela em que o veículo fica totalmente em seu interior e deixando o maior espaço disponível possível para manobras, com uma margem de segurança de 2 cm para executar as manobras.

As dimensões das vagas são definidas em função do comprimento (A) e da largura (B) do veículo [10], [13]. Para o estacionamento paralelo de frente, os valores de largura e comprimento da vaga são, respectivamente, $1,7B$ e $3A$. De forma análoga, para o estacionamento paralelo de ré a vaga apresenta largura igual a $1,3B$ e comprimento de $1,85A$.

A Fig. 6 e a Tabela IV apresentam os resultados obtidos para o estacionamento paralelo de frente. Para o estacionamento paralelo de ré, os resultados são apresentados por meio da Fig. 7 e da Tabela IV.

Pode-se notar que as trajetórias são executadas com sucesso, o veículo é conduzido sem interrupções de movimento, do início ao fim, alcançando uma pose final adequada e desviando dos obstáculos definidos pelas vagas. Diante dos dados e das

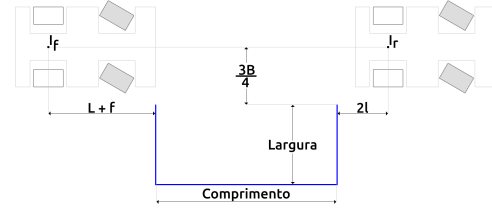
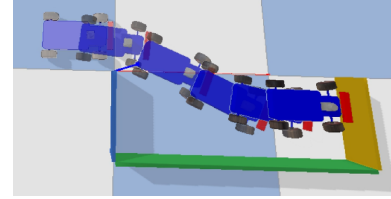
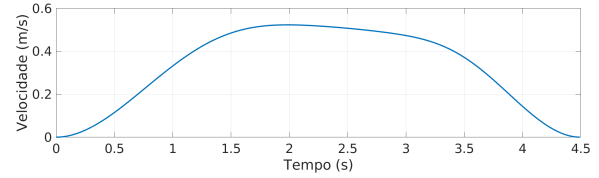


Fig. 5. Posições iniciais.

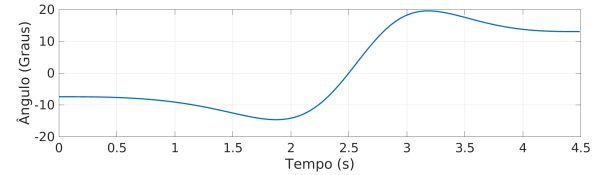
Figuras 6b, 6c, 7b e 7c, é evidente que as trajetórias respeitam os limites do veículo.



(a) Poses ao longo do tempo



(b) Velocidade



(c) Ângulo de esterçamento (ϕ)

Fig. 6. Estacionamento paralelo de frente.

TABELA IV
DADOS DA TRAJETÓRIA DO ESTACIONAMENTO PARALELO

Parâmetro	Frente	Ré
k_0	1,782	1,00
k_1	2,316	1,834
V_s	1	-1
$ \phi _{\max}$	$19,561^\circ$	$31,944^\circ$
Comprimento da trajetória	1,476 m	1,013 m
Tempo	4,495 s	5,900 s
$ V _{\max}$	0,523 m/s	0,288 m/s
$ \dot{V} _{\max}$	0,5 m/s ²	0,227 m/s ²
$ \phi _{\max}$	0,816 rad/s	1,000 rad/s
Largura da vaga	0,493 m	0,377 m
Comprimento da vaga	1,425 m	0,879 m

A. Aplicação

O planejamento de trajetórias para estacionamento de veículos é de grande valia para a otimização de alocação de vagas em ambientes conhecidos, como garagens de shoppings ou supermercados. Como exemplo, considera-se uma garagem na qual há duas colunas estruturais no caminho entre a entrada e vaga de estacionamento disponível. Portanto, o algoritmo

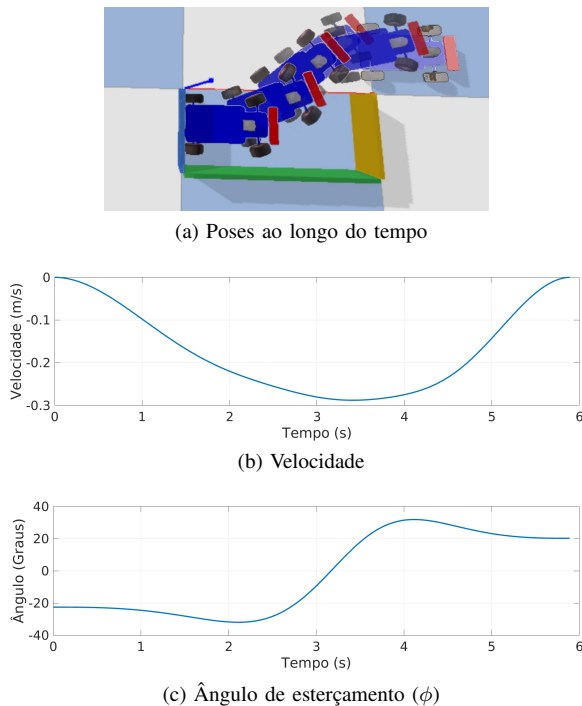


Fig. 7. Estacionamento paralelo de ré.

de planejamento de trajetórias deve ser utilizado em duas partes. A primeira trajetória tem de levar o veículo da entrada da garagem, desviando dos eventuais obstáculos, até o local adequado para efetuar o estacionamento, realizado por meio da segunda trajetória. Em uma aplicação de tempo real, o mapa da garagem deve estar disponível e as características do veículo devem ser previamente conhecidas (podem, por exemplo, serem medidas na entrada da garagem). A Fig. 8 apresenta o resultado da emulação da condição descrita, contendo a planta da garagem com suas limitações e colunas, para que o veículo seja direcionado à vaga adequada. Nota-se que o CLMR é estacionado corretamente, desempenhando trajetórias contínuas e suaves a destinos específicos e sem manobras desnecessárias que levariam a espera de outros carros na fila. O

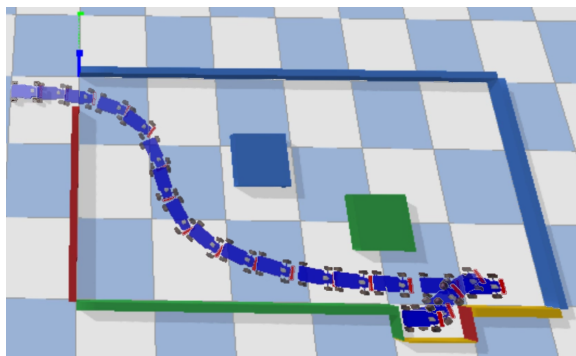


Fig. 8. Estacionamento em uma garagem com obstáculos.

exemplo apresentado tem por objetivo salienta a utilidade da realização do planejamento de trajetórias para estacionamento de veículos do tipo carro em aplicações do cotidiano das cidades. Para um maior refinamento, e objetivando otimizar

o tempo gasto em manobras e busca de vagas em estacionamento, outros parâmetros podem ser considerados. Ademais, considerações sobre o consumo de combustível, o incômodo sonoro gerado, o tráfego de pessoas e a sua segurança, etc. podem ser consideradas na formulação de um problema de otimização para a entrada, o estacionamento e a saída de veículos de um espaço de estacionamento.

VI. CONCLUSÕES

Este artigo apresenta um planejador de trajetórias que realiza manobras de estacionamento paralelo de veículos do tipo carro de maneira contínua e suave, evitando-se obstáculos pelo caminho e respeitando os limites do automóvel. A modelagem é baseada no modelo geométrico de Ackermann para evitar derrapagem, o planejamento de trajetórias é baseado na combinação entre interpolação polinomial e um algoritmo genético de otimização, por meio de uma função objetivo combinada com uma técnica de penalização. Como exemplo de aplicação, é emulado um exemplo simples de estacionamento em uma garagem com obstáculos no caminho, em que o veículo deve se dirigir próximo à vaga e efetuar o estacionamento. A proposta prova-se viável e útil para desenvolvimentos futuros para o planejamento de trajetórias para veículos do tipo carro.

Cabe ressaltar que o processo de aplicação do algoritmo genético é quase exclusivamente empírico, demandando experiência para definir seus parâmetros. Além disso, sua execução pode demandar um tempo considerável. Com isso em mente, a realização de emulações em plataformas de simulação física tal como a realizada neste trabalho conferem maior realismo e permitem avaliar de maneira efetiva o desempenho da implementação proposta. Recomenda-se que investigações futuras incluam o detalhamento na utilização do modelo dinâmico e a comparação de outras técnicas de otimização com o algoritmo aqui proposto, bem como com outras soluções propostas na literatura, além de validação experimental.

AGRADECIMENTOS

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) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelos seus suportes financeiros no desenvolvimento deste trabalho.

REFERÊNCIAS

- [1] M. M. Rashid, M. M. Rahman, M. R. Islam, O. N. Alwahedy, and A. Abdullahi, "Autonomous 4wd smart car parallel self-parking system by using fuzzy logic controller," *American International Journal of Sciences and Engineering Research*, vol. 2, no. 2, pp. 1–31, 2019.
- [2] D. C. Conner, H. Kress-Gazit, H. Choset, A. A. Rizzi, and G. J. Pappas, "Valet parking without a valet," in *2007 IEEE/RSJ international conference on intelligent robots and systems*, pp. 572–577, IEEE, 2007.
- [3] B. Li and Z. Shao, "Simultaneous dynamic optimization: A trajectory planning method for nonholonomic car-like robots," *Advances in Engineering Software*, 09 2015.
- [4] N. A. Martins, E. S. Elyoussef, D. W. Bertol, E. R. De Pieri, U. F. Moreno, and E. d. B. Castelan, "Trajectory tracking of a nonholonomic mobile robot with kinematic disturbances: A variable structure control design," *IEEE Latin America Transactions*, vol. 9, no. 3, pp. 276–283, 2011.

- [5] D. Perez Morales, O. Kermorgant, S. Dominguez Quijada, and P. Martinet, "Laser-based control law for autonomous parallel and perpendicular parking," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pp. 64–71, 2018.
- [6] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 4327–4332, 2018.
- [7] S. Upadhyay and A. Ratnoo, "A point-to-ray framework for generating smooth parallel parking maneuvers," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1268–1275, 2018.
- [8] J. Song, W. Zhang, X. Wu, H. Cao, Q. Gao, and S. Luo, "Laser-based SLAM automatic parallel parking path planning and tracking for passenger vehicle," *IET Intelligent Transport Systems*, vol. 13, pp. 1557–1568, July 2019.
- [9] M. Ye, X. Ji, and Y. Zhao, "A trajectory planning method based on b-spline algorithm for automatic parking systems," in *Proceedings of the 3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)*, Atlantis Press, 2019.
- [10] N. Nakrani and M. Joshi, "An intelligent fuzzy based hybrid approach for parallel parking in dynamic environment," *Procedia Computer Science*, vol. 133, pp. 82–91, 2018.
- [11] E. Ballinas, O. Montiel, O. Castillo, Y. Rubio, and L. T. Aguilar, "Automatic parallel parking algorithm for a carlike robot using fuzzy pd+ i control," *Engineering Letters*, vol. 26, no. 4, 2018.
- [12] D. Kim, W. Chung, and S. Park, "Practical motion planning for car-parking control in narrow environment," *IET Control Theory & Applications*, vol. 4, pp. 129–139, Jan. 2010.
- [13] K. Demirli and M. Khoshnejad, "Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller," *Fuzzy Sets and Systems*, vol. 160, pp. 2876–2891, Oct. 2009.
- [14] D. Wang, H. Liang, T. Mei, and H. Zhu, "Research on self-parking path planning algorithms," in *Proceedings of 2011 IEEE International Conference on Vehicular Electronics and Safety*, IEEE, July 2011.
- [15] K. Erkorkmaz and Y. Altintas, "High speed CNC system design. part i: jerk limited trajectory generation and quintic spline interpolation," *International Journal of Machine Tools and Manufacture*, vol. 41, pp. 1323–1345, July 2001.
- [16] A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Computers & Electrical Engineering*, vol. 38, no. 6, pp. 1564–1572, 2012.
- [17] J. Tu and S. X. Yang, "Genetic algorithm based path planning for a mobile robot," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 1, pp. 1221–1226, IEEE, 2003.
- [18] S. Ni, Z. Liu, Y. Cai, and X. Wang, "Modelling of ship's trajectory planning in collision situations by hybrid genetic algorithm," *Polish Maritime Research*, vol. 25, pp. 14–25, Sept. 2018.
- [19] O.-R. A. D. O. Committee, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles." https://www.sae.org/standards/content/j3016_202104/, 2021. [Online; accessed 24-May-2021].
- [20] C. Zhu and R. Rajamani, "Global positioning system-based vehicle control for automated parking," *I MECH E Part D Journal of Automobile Engineering*, vol. 220, no. 1, pp. 37–52, 2006.
- [21] B. C. Pinheiro, "Sistema de controle tempo real embarcado para automação de manobra de estacionamento," *Dissertação de Mestrado em Automação e Sistemas, Universidade Federal de Santa Catarina*, 2009.
- [22] C. M. Filho and D. F. Wolf, "Dynamic inversion-based control for front wheel drive autonomous ground vehicles near the limits of handling," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, IEEE, Oct. 2014.
- [23] T. C. Revoredo, F. Mora-Camino, and J. Slama, "A two-step approach for the prediction of dynamic aircraft noise impact," *Aerospace Science and Technology*, vol. 59, pp. 122–131, 2016.
- [24] J. Reimpell, H. Stoll, and J. Betzler, *The automotive chassis: engineering principles*. Elsevier, 2 ed., 2001.
- [25] P. Simionescu and M. Smith, "Initial estimates in the design of central-lever steering linkages," *J. Mech. Des.*, vol. 124, no. 4, pp. 646–651, 2002.
- [26] T. D. Gillespie, "Fundamentals of vehicle dynamics," *SAE International*, 1992.
- [27] K. Hartani, M. Bourahla, Y. Miloud, and M. Sekour, "Electronic differential with direct torque fuzzy control for vehicle propulsion system," *Turk J Elec Eng end Comp Sci*, vol. 17, no. 1, 2009.
- [28] S. Thrun, "Robotic mapping: a survey," *Exploring Artificial Intelligence in the NewMillenium, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.*, pp. 1–35, 2003.
- [29] E. V. Argento, R. P. Vieira, and T. C. Revoredo, "Estacionamento paralelo autônomo de veículos leves de passeio," *Anais da Sociedade Brasileira de Automática*, vol. 2, no. 1, 2020.
- [30] J. Lévine, *Analysis and Control of nonlinear systems: A flatness-based approach*. Springer-Verlag Berlin Heidelberg, 2009.
- [31] D. Pedrosa, A. Medeiros, and P. Alsina, "Point-to-point paths generation for wheeled mobile robots," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, IEEE, 2003.
- [32] R. Walambe, N. Agarwal, S. Kale, and V. Joshi, "Optimal trajectory generation for car-type mobile robot using spline interpolation**this work is carried out under the research project grant sanctioned under the WOS-a scheme by department of science and technology (DST), govt. of india.," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 601–606, 2016.
- [33] J. H. Holland et al., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.
- [34] C. Darwin, *A origem das espécies*. EDITORA BIBLIOMUNDI SERVIÇOS DIGITAIS LTDA, 2020.
- [35] C. R. Houck, J. Joines, and M. G. Kay, "A genetic algorithm for function optimization: a matlab implementation," *Ncsu-ie tr*, vol. 95, no. 09, pp. 1–10, 1995.
- [36] C. R. Reeves, "Genetic algorithms for the operations researcher," *INFORMS journal on computing*, vol. 9, no. 3, pp. 231–250, 1997.
- [37] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," *URL http://pybullet.org*, 2016.
- [38] R. Michalik and A. Janota, "The PyBullet module-based approach to control the collaborative YuMi robot," in *2020 ELEKTRO*, IEEE, May 2020.



Renan P. Vieira nasceu no Rio de Janeiro, Brasil, em 1991. Recebeu o diploma em Engenharia Elétrica com ênfase em Sistemas Eletrônicos pela Universidade do Estado do Rio de Janeiro, Rio de Janeiro, Brasil, em 2020. Atualmente está cursando seu Mestrado em Controle e Automação na Universidade do Estado do Rio de Janeiro. Seus interesses de pesquisa incluem modelagem e controle de robôs móveis do tipo carro, veículos elétricos e sistemas inteligentes.



Eduardo V. Argento é formado em Engenharia Elétrica com ênfase em Sistemas Eletrônicos pela Universidade do Estado do Rio de Janeiro no ano de 2020. Atualmente cursa mestrado em Engenharia Elétrica com área de concentração em Métodos de Apoio à Decisão na Pontifícia Universidade Católica do Rio de Janeiro. Suas áreas de interesse são: controle e automação, robótica, especialmente móvel, eletrônica e inteligência computacional.



Téo C. Revoredo possui Graduação em Engenharia Elétrica com ênfase em Sistemas eletrônicos pela Universidade do Estado do Rio de Janeiro (UERJ, 2003), Mestrado em Engenharia Elétrica pelo Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia (COPPE) da Universidade Federal do Rio de Janeiro (UFRJ, 2007) e Doutorado em Engenharia Mecânica pela mesma instituição (2012), com período Sanduíche na École Nationale de l'Aviation Civile (ENAC) na França. É Professor Adjunto do Departamento de Engenharia Eletrônica e Telecomunicações (DETEL) e pesquisador do Programa de Pós-Graduação em Engenharia Eletrônica (PEL) da UERJ. Suas áreas de interesse são: robótica, automação, eletrônica, instrumentação e microgeração de energia com fontes renováveis. É membro do Instituto de Engenheiros Elétricos e Eletrônicos (IEEE) e da sua Sociedade de Automação e Robótica (RAS), sendo orientador do Capítulo Estudantil da RAS na UERJ.