

# Prevenção de Colisão de um Manipulador SCARA utilizando Campos Potenciais Artificiais e Caminhos Probabilísticos \*

Emerson V. A. Dias \* Josias G. Batista \*  
 Geraldo L. B. Ramalho \* José L. N. Silva \*\*  
 Darielson A. Souza \*\*

\* LARI - Laboratório de Automação e Redes Industriais, IFCE -  
 Instituto Federal de Educação Ciência e Tecnologia do Ceará -  
 Fortaleza, CE (e-mail: emersonverasifce@gmail.com/  
 josiasbatista@ifce.edu.br/ gramalho@ifce.edu.br).

\*\* UFC - Universidade Federal do Ceará - Fortaleza, CE  
 (leonardo.silva@ifce.edu.br/ darielson@dee.ufc.br)

**Abstract:** This paper compares the Probabilistic Roadmap (PRM) and Artificial Potential Fields (CPA) algorithms, implemented in a SCARA manipulator (Selective Compliance Assembly Robot Arm). The two algorithms are used in collision avoidance path planning and are compared by computational cost and the shortest collision avoidance path. The results show the trajectories generated by the algorithms in the Cartesian space and also the trajectories of each joint of the manipulator, calculated from the inverse kinematics. Speeds and accelerations for trajectories are also shown. In the implementation, the CPA algorithm proved to be more efficient.

**Resumo:** Este artigo compara os algoritmos *Probabilistic Roadmap* (PRM) e Campos Potenciais Artificiais (CPA), implementados em um manipulador SCARA (*Selective Compliance Assembly Robot Arm*). Os dois algoritmos são utilizados no planejamento de trajetória livre de colisão e são comparados pelo custo computacional e pela menor trajetória livre de colisão. Nos resultados são apresentados as trajetórias geradas pelos algoritmos no espaço cartesiano e também as trajetórias de cada junta do manipulador, calculadas a partir da cinemática inversa. Também são apresentadas as velocidade e acelerações para as trajetórias. Na implementação o algoritmo CPA se mostrou mais eficiente.

**Keywords:** Trajectory Planning, Artificial Potential Fields, Probabilistic Roadmap, Collision Avoidance Path.

**Palavras-chaves:** Planejamento de Trajetória, Campos Potenciais Artificiais, *Probabilistic Roadmap*, Trajetória Livre de Colisão .

## 1. INTRODUÇÃO

A robótica foi introduzida no mundo com o objetivo de automatizar processos industriais, isto é, facilitar o trabalho humano. Com a evolução tecnológica o uso da robótica no meio industrial aumentou significativamente e, assim, foram criados diversos tipos de robôs com a finalidade de auxiliar ou até substituir o homem em determinadas tarefas (Souza, 2008). Sob tal ótica, presume-se que os robôs podem realizar uma tarefa com inteligência suficiente para classificar qual a ação necessária a ser escolhida, por exemplo, evitar uma colisão.

A indústria almeja a interação homem-máquina, ou seja, o homem e a máquina trabalhando juntos (Guerin et al., 2019). No entanto, os movimentos pré-determinados que a máquina pode executar podem ser perigosos, visto que

existem possibilidades de ocorrerem acidentes. Sob esse viés, o objetivo deste trabalho é aplicar métodos em que a máquina possa trabalhar sem que ocorra a colisão com qualquer obstáculo.

O setor industrial utiliza artifícios para que não ocorram acidentes no trabalho, por exemplo, isolar uma área para que o robô seja capaz de trabalhar, e quando um operador entrar no espaço de trabalho do robô, ele possa ser desligado. Entretanto, é mais eficiente programar a máquina para compreender a existência de um operador no local. Diante disso, o robô adota um comportamento no qual seja seguro, como diminuir a velocidade do seu movimento ou calcular uma nova trajetória. Vale salientar que a utilização dessa prática descarta a necessidade de desligar o equipamento melhorando, desse modo, a eficiência e o controle de desempenho da produção (Wisskirchen et al., 2017).

A formação de uma trajetória eficaz para robôs vem sendo estudada desde a década de 60, com a finalidade

\* Os autores agradecem ao Programa Institucional de Bolsas de Iniciação Científica - PIBIC IFCE/CNPq/FUNCAP, Edital N° 2/2019 - PRPI/Reitoria - IFCE, pela bolsa de auxílio..

de conseguir com o robô exerça a sua funcionalidade de forma correta (Adôrno et al., 2006). Diante disso, para que o robô se locomova com êxito em circunstâncias propensas a obstáculos, o planejamento de movimento é encontrar um caminho livre de colisão da posição inicial à posição final (Mohanta and Keshari, 2019). Para a elaboração eficaz de rotas é necessário o uso de algoritmos, os quais possibilitarão a leitura da área de trabalho da máquina e a identificação dos obstáculos (Batista et al., 2020b).

A dificuldade de encontrar um algoritmo que proporcione que o robô se locomova do ponto  $q_{inicial}$  ao ponto  $q_{final}$ , pode ser solucionada por algoritmos como Campo Potencial Artificial (CPA) e *Probabilistic Roadmap* (PRM). O método de campo potencial artificial é aplicado para encontrar os parâmetros de distanciamento e para gerar um caminho viável e seguro (Orozco-Rosas, 2019). O CPA é utilizado por diversos pesquisadores, algumas dessas aplicações são com robôs manipuladores industriais e carros autônomos, para geração de trajetória livre de colisão (Batista et al., 2017).

O algoritmo de *Probabilistic Roadmap* é um método baseado em estratégia de amostragem aleatória que pode resolver o problema de caminhos efetivos que são difíceis de construir com a maioria dos algoritmos no espaço de alta dimensão (Chen, 2019). O PRM é proposto para o planejamento de caminhos e a identificação de obstáculos (Mohanta and Keshari, 2019).

Este trabalho tem como objetivo realizar uma comparação entre a implementação dos algoritmos PRM e CPA aplicados a um manipulador SCARA (*Selective Compliance Assembly Robot Arm*) na geração de trajetórias livres de colisão com dois obstáculos estáticos. A comparação entre os dois métodos é demonstrada a partir de um mapa, no qual ambos os algoritmos são submetidos ao procedimento de desafio de identificar os obstáculos e elaborar uma trajetória segura do ponto inicial ao ponto  $q$  final. Nos resultados também são apresentadas as trajetórias, velocidades e acelerações das juntas do manipulador calculadas a partir do modelo da cinemática inversa.

Este artigo está organizado como segue. A seção 2 apresenta as características do manipulador SCARA, bem como a fundamentação teórica sobre CPA e PRM. Na seção 3 é apresentada como os algoritmos foram implementados. A seção 4 apresenta os resultados obtidos e finalmente na seção 5 são apresentadas as conclusões e trabalhos futuros.

## 2. MANIPULADOR SCARA E ALGORITMOS

Nesta seção são apresentados os fundamentos teóricos referentes ao objeto deste artigo. É realizada uma breve descrição do manipulador SCARA (*Selective Compliance Assembly Robot Arm*) e as descrições sobre CPA e PRM.

### 2.1 Manipulador SCARA

O manipulador SCARA utilizado nesse trabalho é um robô de quatro graus de liberdade (GDL), sendo que aqui será trabalhado com dois GDL, a Figura 1 mostra o robô. Por ser um robô tipo SCARA, as duas primeiras juntas, a partir da base, são de rotação em torno de eixos verticais

e trabalhando portanto num plano horizontal ( $xy$ ), como se fosse um robô 2-GDL planar.

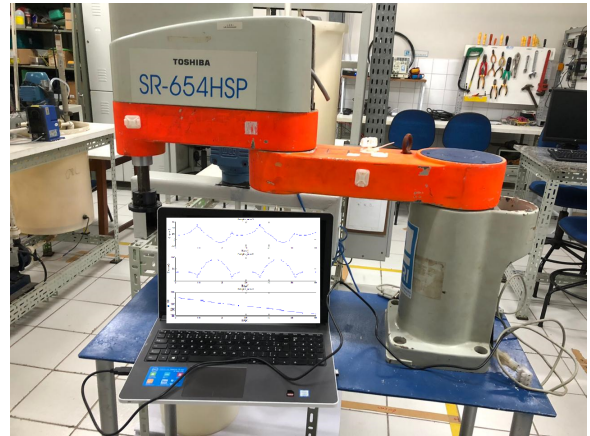


Figura 1. Manipulador robótico SCARA.

Neste trabalho, o manipulador SCARA será considerado como um ponto, cuja posição está localizada no *end-effector*, ou seja será a posição no espaço cartesiano de duas dimensões.

#### 2.1.1 Cinemática direta

Para encontrar o modelo cinemático direto, é utilizada a convenção Denavit-Hartenberg (DH). Portanto, para calcular os parâmetros  $\alpha$ ,  $a$ ,  $d$  e  $\theta$ , foi adotado o sistema de coordenadas do manipulador, apresentado na figura 2, (Hartenberg and Danavit, 1964).

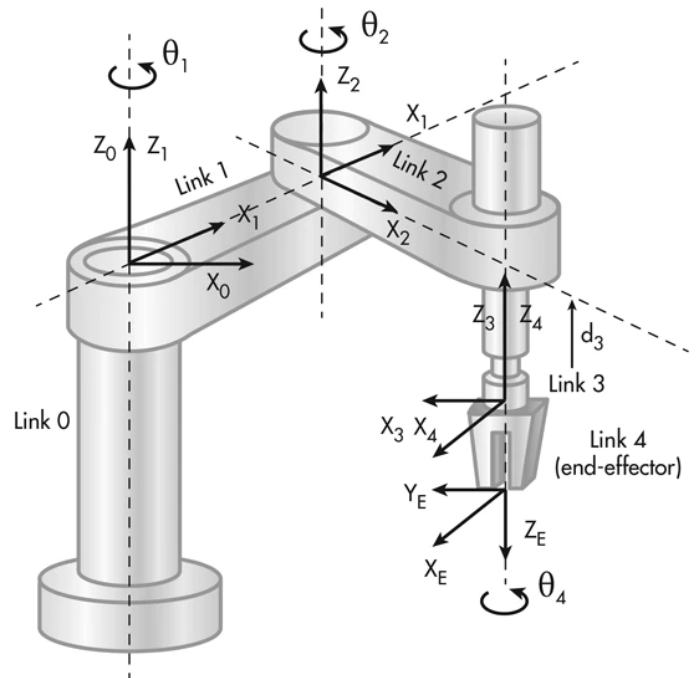


Figura 2. Sistemas de coordenadas do manipulador SCARA, Gonzalez (2017).

A partir dos parâmetros D-H encontra-se as matrizes de transformação homogênea (MTH) dos sistemas de coordenadas que é composta por quatro transformações básicas. A partir MTH e comparando com sua forma geométrica, podemos determinar a posição do elemento final do robô

que pode ser encontrada no espaço cartesiano a partir das coordenadas no espaço das juntas como observado nas equações

$$X = 0,35\cos(\theta_1) + 0,30\cos(\theta_1 + \theta_2) \quad (1)$$

$$Y = 0,35\sin(\theta_1) + 0,30\sin(\theta_1 + \theta_2) \quad (2)$$

As equações (1) e (2) são as soluções para o problema da cinemática de posição direta para o manipulador SCARA.

### 2.1.2 Cinemática inversa

A cinemática inversa resolve a configuração que deve adotar o robô para uma posição e orientação de um ponto conhecido. Uma maneira simples de resolver o problema da cinemática inversa do robô é usando o método geométrico. A partir da cinemática inversa, será possível determinar o valor do ângulo de cada junta, a fim de colocar o manipulador na posição e a partir de então gerar trajetórias no espaço das juntas (Silva et al., 2016), (Batista et al., 2020a).

Partindo das equações da cinemática direta (1) e (2) e aplicando algumas transformações trigonométricas encontra-se as equações da cinemática inversa que são dadas por:

$$\theta_1 = \tan^{-1} \left[ \frac{P_y(L_1 + L_2\cos(\theta_2)) - P_x L_2 \sin(\theta_2)}{P_x(L_1 + L_2\cos(\theta_2)) - P_y L_2 \sin(\theta_2)} \right] \quad (3)$$

$$\theta_2 = \cos^{-1} \left( \frac{P_x^2 + P_y^2 - L_1^2 - L_2^2}{2L_1 L_2} \right) \quad (4)$$

onde  $L_1 = 0,35$  e  $L_2 = 0,30$ , e são os valores dos comprimentos de cada junta do manipulador como mostrado na figura 2.

As equações (3) e (4) são as soluções do problema da cinemática inversa do manipulador SCARA são usadas para realizar a geração de trajetória do manipulador.

### 2.2 Algoritmo de Campos Potenciais Artificiais

O método de CPA foi utilizado por Khatib (1986) em robôs manipuladores com um ponto alvo e com obstáculos estáticos e em movimento. Neste método, considera-se um ponto inicial,  $q_{inicial}$ ; uma posição de um ponto que se move num campo de forças; e um ponto final,  $q_{final}$ , que fornece uma força atrativa e os obstáculos, que fornecem forças repulsivas. No método de campos potenciais artificiais o manipulador se move em um campo (vetorial) de forças  $F(\theta)$ , derivado de um campo (escalar) potencial  $U(\theta)$  (Khatib, 1986), (Spong et al., 2020).

$$F(x) = \nabla[U(\theta)] \quad (5)$$

Apesar deste método ter sido inicialmente utilizado em robôs manipuladores, atualmente, a maioria das aplicações são na robótica móvel (Batista et al., 2018).

A função potencial  $U(\theta)$  é tipicamente definida no espaço livre como a combinação linear de um campo atrativo,  $U_{atr}(\theta)$ , que puxa o robô para a posição alvo, e de um potencial repulsivo,  $U_{rep}(\theta)$ , que o empurra para longe dos obstáculos (Latombe, 2012). Assim, considerando a presença de um único obstáculo, define-se as coordenadas da posição do obstáculo como sendo  $\theta_{obs} = (\theta_{1obs}, \theta_{2obs})$ , a posição do alvo por  $\theta_{alvo} = (\theta_{1alvo}, \theta_{2alvo})$ , e a posição do centro do robô por  $\theta = (\theta_1, \theta_2)$  (no caso de um robô móvel)

ou origem do sistema de coordenadas da ferramenta (no caso de um manipulador), a trajetória gerada é função do seguinte campo de potencial artificial, dado por (6):

$$U(\theta) = U_{atr}(\theta) + U_{rep}(\theta), \quad (6)$$

em que  $U(\theta)$  é o potencial resultante do campo artificial,  $U_{atr}(\theta)$  é o potencial atrativo produzido pela posição do alvo,  $\theta_{alvo}$ , no centro do robô,  $\theta$ , e  $U_{rep}(\theta)$  é o potencial repulsivo induzido pelo obstáculo em  $\theta$ . O vetor que representa a força que deve ser aplicada no centro do robô móvel ou ferramenta de um manipulador é dado por:

$$F(\theta) = F_{atr}(\theta) + F_{rep}(\theta) \quad (7)$$

com

$$F_{atr}(\theta) = -\nabla[U_{atr}(\theta)] \quad (8)$$

e

$$F_{rep}(\theta) = -\nabla[U_{rep}(\theta)] \quad (9)$$

em que  $F(\theta)$  é a força resultante,  $F_{atr}(\theta)$  é uma força atrativa que guia o centro do robô ao alvo, e  $F_{rep}(\theta)$  é uma força que induz uma repulsão artificial da superfície do obstáculo produzida por  $U_{rep}(\theta)$  (Sciavicco and Siciliano, 2012).

### 2.3 Método Probabilistic Roadmap

O método de *Probabilistic Roadmap* fornece o planejamento de movimento para encontrar um caminho livre de colisão. É utilizado com êxito em robôs móveis na presença de obstáculos (Mohanta and Keshari, 2019). Essa metodologia é extremamente eficiente, devido à capacidade de projetar caminhos rapidamente e à prevalência do trajeto mais curto. Neste método, inicialmente é gerada uma amostra aleatória do espaço de configuração usando uma distribuição de probabilidade uniforme. Em sequência o algoritmo testa a amostra quanto a colisão, se a qual não for detectada, será gerada a trajetória do ponto  $q_{inicial}$  ao  $q_{final}$  (Siciliano et al., 2011).

O PRM gera aleatoriamente um conjunto de configurações, os quais são representados como nós. Em seguida o planejador conecta esses nós até ser elaborado um caminho mais eficiente (Spong et al., 2020). Na literatura direcionada ao PRM, essas são as principais funções utilizadas para determinar a melhor ligação entre os nós criados, a partir da distância calculada:

$$\|q' - q\| = \left[ \sum_{i=1}^n (q'_i - q_i)^2 \right]^{\frac{1}{2}}, \quad (10)$$

$$\max_n |q'_i - q_i|, \quad (11)$$

$$\left[ \sum p \in A \|p(q') - p(q)\|^2 \right]^{\frac{1}{2}}, \quad (12)$$

$$\max_{p \in A} \|p(q') - p(q)\|. \quad (13)$$

Para o algoritmo identificar o caminho mais curto são executados cálculos de distância entre os nós. A equação (11) é a mais usada para essa finalidade (Spong et al., 2020).

O robô possui  $n$  articulações, que estão inclusas nas equações (10) e (11). Para descobrir a distância entre os nós aleatórios setados no mapa probabilístico, é executada

a diferença entre as especificações no eixo de  $q$  e de  $q'$ , que são configurações correspondentes a nós diferentes que estão situados no mapa. A variável  $q_i$  se refere à configuração da  $i$ -ésima articulação e  $p(q)$  declara o ponto de referência da área de trabalho  $p$  de um conjunto  $A$  de pontos de referência do robô da configuração  $q$  (Spong et al., 2020).

### 3. METODOLOGIA

Nesta seção são apresentadas as implementações dos algoritmos CPA e PRM, bem como a cena utilizada para a geração de caminhos livres de colisões.

#### 3.1 Implementação do algoritmo CPA

O método de colisão livre a partir de um campo potencial artificial é baseado em um campo de forças em determinada trajetória que influenciará no trajeto do robô. Como uma partícula pontual, o robô sofre influência pelo campo potencial artificial. O campo é construído para que o robô seja atraído para a configuração final  $Q_{final}$  e seja repelido pelos locais determinados como obstáculos (Spong et al., 2020).

A geração de trajetória usando campos potenciais é um método que pode ser utilizado para planejamento global *off-line*, quando o ambiente do robô é conhecido à priori, ou em planejamento local *on-line* quando o ambiente é desconhecido e a presença dos obstáculos vai sendo detectada pelos sensores montados no robô (Chen et al., 2009).

##### 3.1.1 Força de repulsão da posição inicial

Com a definição da posição inicial, uma força de repulsão pode ser gerada na mesma. É uma maneira de garantir que a inércia do movimento será vencida e também que não haverá um mínimo local na posição inicial. A função adotada para representar o campo potencial de repulsão da posição inicial foi (LaValle, 2006):

$$U_r(x, y) = \frac{K_r}{\sqrt{(x - x_0)^2 + (y - y_0)^2}}, \quad (14)$$

em que  $K_r$  é o parâmetro da intensidade desta repulsão na posição inicial, ou seja,  $K_r$  é o parâmetro que define a intensidade do campo de repulsão para a posição inicial.  $x$  e  $y$  são os pontos de referência para o CPA;  $x_0$  e  $y_0$  são os pontos iniciais onde o robô irá iniciar a trajetória.

##### 3.1.2 Força de atração da posição final

Com a definição da posição final, uma força de atração pode ser gerada na própria posição inicial. A função adotada que representa o campo potencial de atração para posição final foi adotada como (Batista et al., 2018):

$$U_a(x, y) = \frac{1}{2}K_a[(x - x_f)^2 + (y - y_f)^2], \quad (15)$$

em que  $K_a$  é um parâmetro da intensidade de atração à posição final, ou seja,  $K_a$  é o parâmetro que define a intensidade do campo de atração para posição final.  $x$  e  $y$  são os pontos de referência para o CPA;  $x_f$  e  $y_f$  são os pontos finais da trajetória gerada.

##### 3.1.3 Força de repulsão do obstáculo

Com a definição dos obstáculos, ou seja, as coordenadas que representam a localização dos centros dos obstáculos

$(x_{oi}, y_{oi})$ , são geradas as forças de repulsão. A função que representa a repulsão devido à presença de um obstáculo foi adotada como Batista et al. (2018):

$$U_o(x, y) = \frac{K_o}{\sqrt{(x - x_{oi})^2 + (y - y_{oi})^2}}, \quad (16)$$

em que  $K_o$  é um parâmetro que controla a intensidade do campo potencial de repulsão aos obstáculos, ou seja,  $K_o$  define a intensidade deste campo de repulsão (Batista et al., 2018).  $x$  e  $y$  são os pontos de referência para o CPA;  $x_{oi}$  e  $y_{oi}$  são os pontos que definem a posição do obstáculo.

#### 3.2 Implementação do algoritmo PRM

A lógica de *Probabilistic Roadmap* é fundamentada em analisar anteriormente a trajetória do robô a partir de um mapa pré-determinado, portanto, esse conhecimento do local que o robô irá transitar é possibilitado que ele calcule a rota que irá seguir. Diante disso, o manipulador SCARA calculará o trajeto com eficácia, visando o distanciamento de todos os obstáculos ao longo do caminho.

No PRM o mapa descrito para a máquina será analisado, e verificado, para que seja identificado onde estão localizados os obstáculos e as rotas de acesso livre. O método é baseado na plotagem aleatória de pontos imaginários. Estes pontos serão fixados em locais que foram determinados como livres, ou seja, sem obstáculos. A trajetória elaborada por esse algoritmo preza não só pelo o movimento livre de colisões, mas também pelo caminho mais curto (Siciliano et al., 2011).

Para o planejamento de trajetória com o PRM são necessários os seguintes passos:

- (1) O caminho é um gráfico  $G(V, E)$ ;
- (2) A configuração do robô  $q \rightarrow Q_{free}$  é um vértice;
- (3) A borda  $(q_1, q_2)$  implica um caminho livre de colisão entre essas configurações do robô;
- (4) É necessária uma métrica para  $d(q_1, q_2)$  (por exemplo, distância euclidiana);
- (5) Utilização de amostragem grossa dos nós e fina das bordas;
- (6) Resultado: um caminho em  $Q_{free}$ .

O pseudo-código do algoritmo PRM é apresentado a seguir.

#### 3.3 Cenário utilizado

Para a geração de trajetória do manipulador SCARA, livre de colisão, foi considerado o espaço cartesiano  $(XY)$ . A partir dos pontos obtidos pelos algoritmos de CPA e PRM, resolve-se a cinemática inversa, e encontram-se as posições de cada junta do manipulador. Estes pontos podem ser aplicados em qualquer manipulador que tenha o espaço de trabalho compatível ou em um robô móvel. O caminho gerado deve evitar a colisão de dois obstáculos circulares, mostrados na Figura 3, com tamanhos iguais cujo raio é igual a 0,2 m.

O cenário é formado pela posição inicial do  $(0, 4; 0, 7)$ , posição final  $(0, 4; 0, 1)$  do manipulador e o obstáculo (círculos), no espaço cartesiano. Para os obstáculos utilizou-se as posições  $(0, 3; 0, 4)$  para o obstáculo 1 (que se encontra mais à esquerda), e  $(0, 5; 0, 4)$  para o obstáculo 2 (que

---

**Algoritmo Probabilistic Roadmap.**

---

**Entrada:**n: número de nós de entrada do *roadmap*

k: número de vizinhos para cada configuração

**Saída:**A *roadmap*  $G = (V, E)$ 

---

1:  $V \leftarrow \emptyset$ 2:  $E \leftarrow \emptyset$ 3: **enquanto**  $|V| < n$  **faça**4: **repete**5:  $q \leftarrow$  a random configuração em  $Q$ 6: **até quando**  $q$  for livre de colisão7:  $V \leftarrow V \cup \{q\}$ 8: **fim enquanto**9: **para todo**  $q \in V$  **faça**10:  $N_q \leftarrow k$  vizinhos de  $q$  escolhido a partir de  $V$  de acordo com a distância11: **para todo**  $q' \in N_q$  **faça**12: **se**  $(q, q') \notin E$  e  $\Delta(q, q') \neq null$  **então**13:  $E \leftarrow E \cup \{(q, q')\}$ 14: **fim se**15: **fim para**16: **fim para**

---

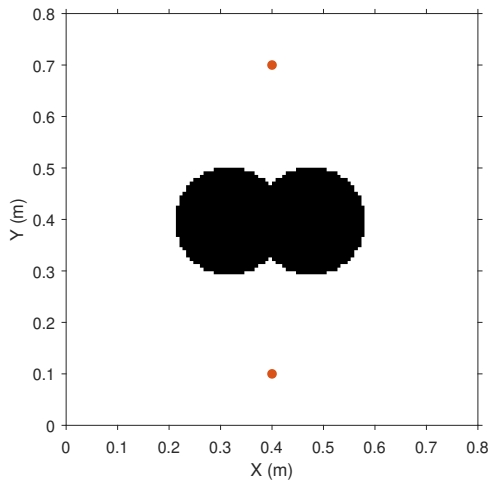


Figura 3. Cenário utilizado para a implementação dos algoritmos.

se encontra mais à esquerda) na cena, posições essas que definem o centro dos mesmos no espaço cartesiano.

É importante destacar o fato de que a partir das técnicas utilizadas, o obstáculo aqui apresentado tem formato circular. Isto reduz a geração de mínimos locais no momento do planejamento de caminhos para o algoritmo CPA, conforme indica (Volpe and Khosla, 1990), além do fato de tornar o sistema capaz de analisar obstáculos de vários formatos como triangular, retangular e trapezoidal que também reduzem a geração de mínimos locais no planejamento de caminhos.

São utilizados dois critérios de comparação dos algoritmos: o tempo de processamento e o comprimento da trajetória. Os valores médios desses critérios são calculados para 20 repetições da simulação. As simulações são realizadas em um computador com um processador Core i7 - 7ª Geração, com velocidade de processamento de 2,90 GHz e com memória RAM de 8,00 GB. Os algoritmos foram imple-

mentados na linguagem M-código com algumas funções específicas para a linguagem.

## 4. RESULTADOS

Nesta seção são apresentados os resultados dos algoritmos de CPA e PRM, bem como as considerações finais sobre os resultados. São apresentados duas situações para cada algoritmo, um resultado com colisão e sem colisão. Também são apresentadas as trajetórias, velocidades e acelerações das juntas para as situações dos algoritmos sem colisão.

### 4.1 Resultados do CPA

A Figura 4 apresenta o caminho livre de colisão gerado pelo algoritmo de CPA. Os parâmetros utilizados para a geração deste caminho foram:  $K_r = 25$ ,  $K_a = 30$  e  $K_o = 4, 5$ , os quais foram encontrados empiricamente. Na Figura 5 é mostrada a superfície potencial para a cena apresentada na Figura 4.

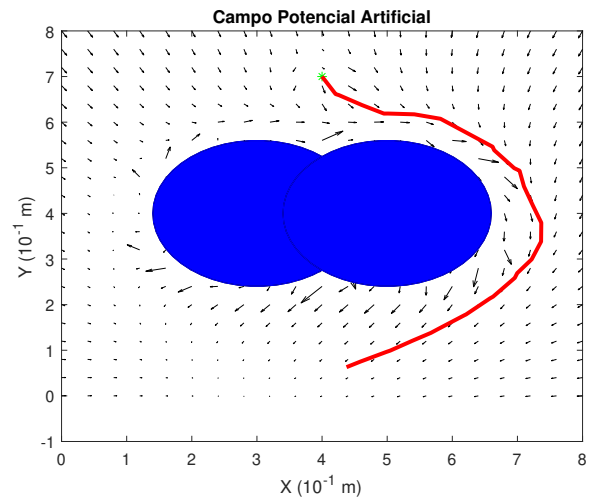


Figura 4. Caminho livre de colisão no espaço cartesiano gerado pelo algoritmo CPA.

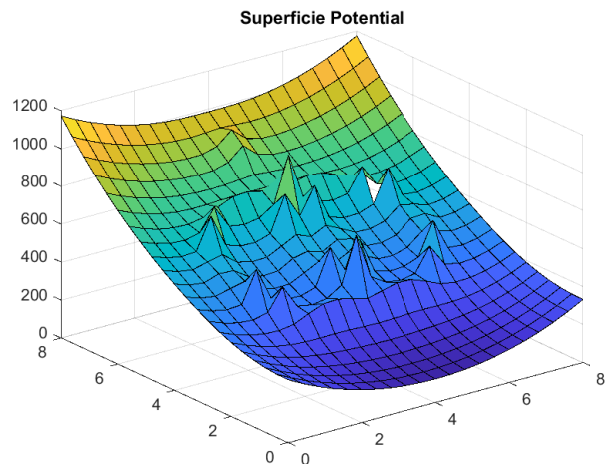


Figura 5. Superfície resultante do algoritmo de CPA da cena da Figura 4

Na Figura 6 é apresentada uma situação onde ocorreu uma colisão com os obstáculos no algoritmo de CPA, os parâmetros do algoritmos utilizados foram:  $K_r = 25$ ,  $K_a = 30$  e  $K_o = 1, 0$ .

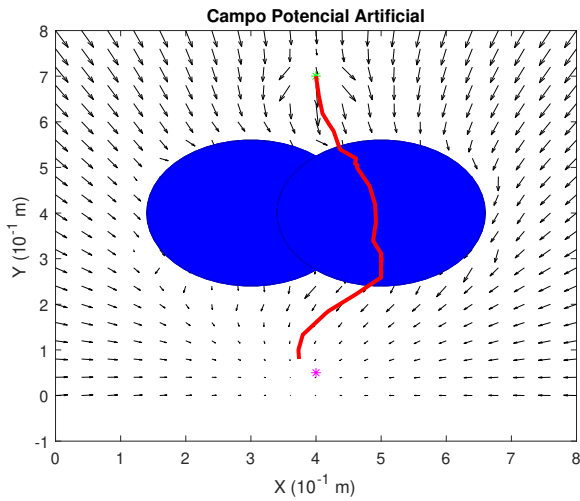


Figura 6. Caminho com colisão no espaço cartesiano gerado pelo algoritmo CPA.

As figuras 7, 8 e 9 apresentam as trajetórias, velocidades e acelerações, respectivamente, das juntas 1 e 2, do manipulador para a situação sem colisão, apresentada na Figura 4.

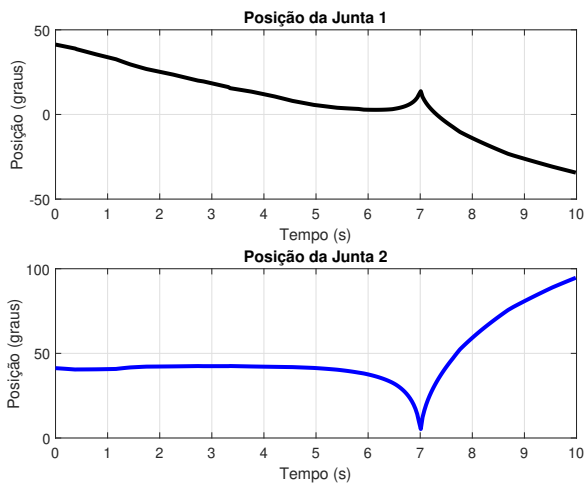


Figura 7. Trajetórias das juntas do manipulador SCARA geradas a partir do CPA.

#### 4.2 Resultados do PRM

Seguindo a mesma ideia do CPA, apresenta-se aqui os resultados para o algoritmo PRM.

As figuras 10 e 11 apresentam os caminhos no espaço cartesiano para 100 e 1.000 pontos utilizados no algoritmo PRM, respectivamente. Observa-se que o algoritmo PRM com 1.000 pontos gera um caminho com colisão, pois os pontos ficaram muito próximos dos obstáculos, e algoritmo gera o menor caminho.

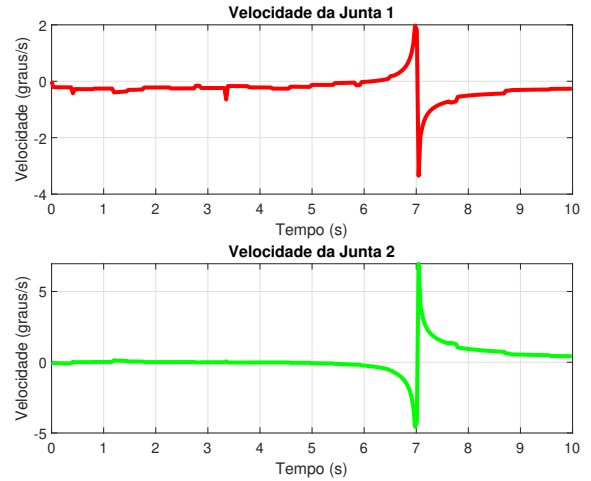


Figura 8. Velocidades das juntas do manipulador SCARA geradas a partir do CPA.

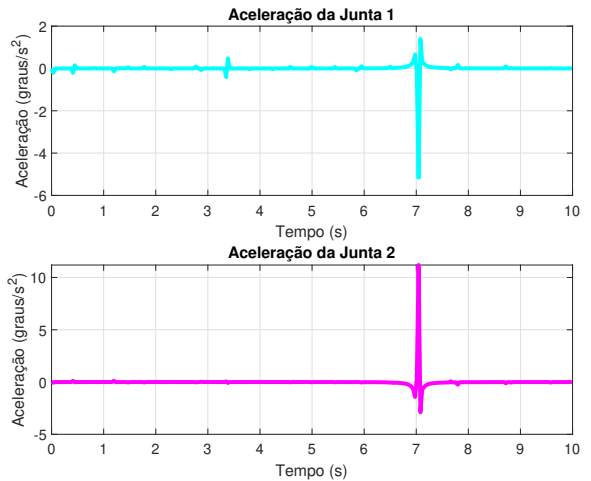


Figura 9. Acelerações das juntas do manipulador SCARA geradas a partir do CPA.

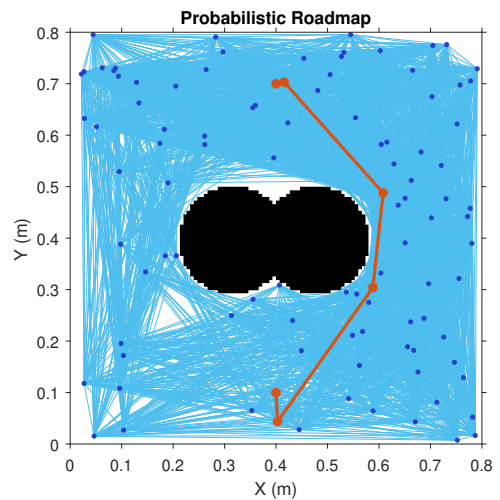


Figura 10. Caminho no espaço cartesiano para 100 pontos do PRM.



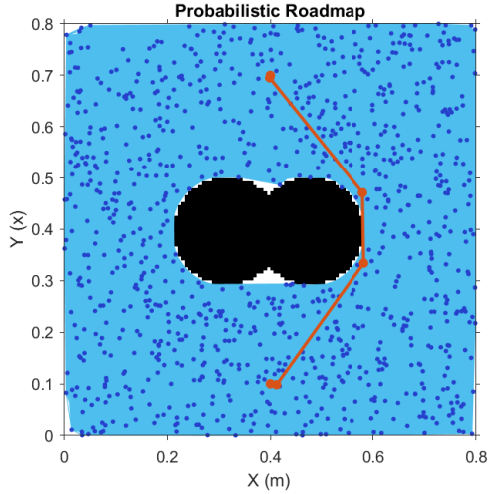


Figura 11. Caminho no espaço cartesiano para 1000 pontos do PRM.

As figuras 12, 13 e 14 apresentam as trajetórias, velocidades e acelerações, respectivamente, das juntas 1 e 2, do manipulador para a situação sem colisão, apresentada na Figura 10.

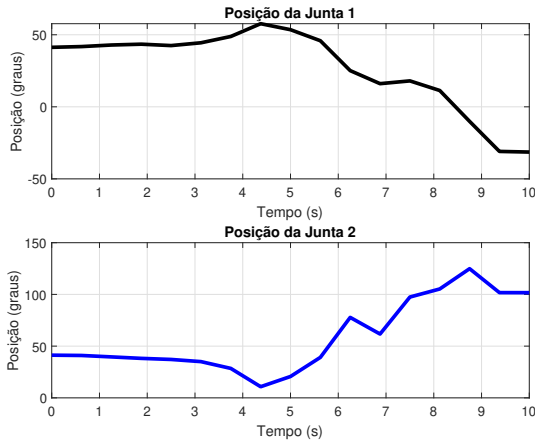


Figura 12. Trajetórias das juntas do manipulador SCARA geradas a partir do PRM.

### 4.3 Discussões

Realizou-se uma comparação mais detalhada entre os algoritmos. Verificou-se o tempo de processamento médio para 20 repetições da simulação e o comprimento das trajetórias geradas sem colisão por cada algoritmo. Essa comparação é apresentada na Tabela 1, para as situações sem colisão de ambos os algoritmos.

Tabela 1. Tempo de processamento médio e comprimento das trajetórias gerados pelos algoritmos CPA e PRM (100 pontos), sem colisão.

| Algoritmo | Tempo de Proc. Médio (s) | Trajetoária (m) |
|-----------|--------------------------|-----------------|
| CPA       | 0,65429                  | 0,96375         |
| PRM       | 1.02448                  | 1.06421         |

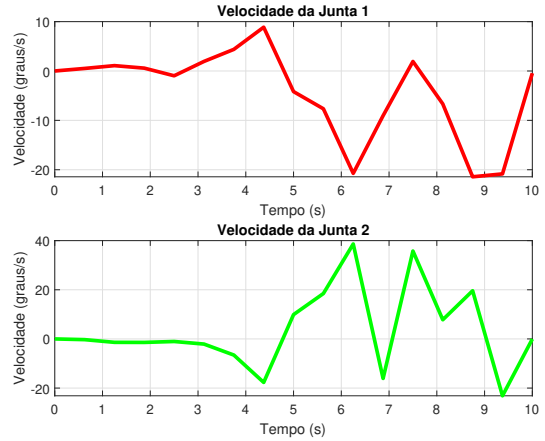


Figura 13. Velocidades das juntas do manipulador SCARA geradas a partir do PRM.

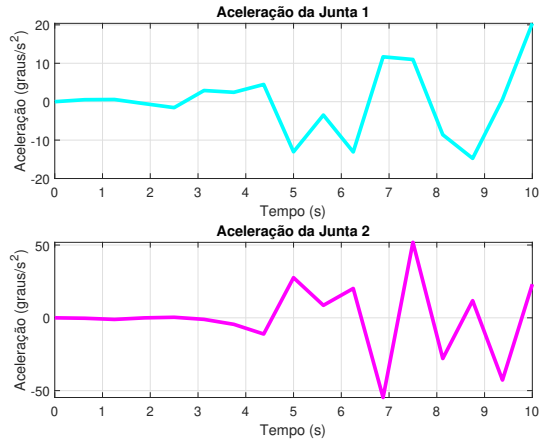


Figura 14. Acelerações das juntas do manipulador SCARA geradas a partir do PRM.

Na Tabela 2 são apresentados os resultados com as colisões, do tempo de processamento médio e comprimento das trajetórias para os dois algoritmos.

Tabela 2. Tempo de processamento médio e comprimento das trajetórias gerados pelos algoritmos CPA e PRM (1.000 pontos), com colisão.

| Algoritmo | Tempo de Proc. Médio (s) | Trajetoária (m) |
|-----------|--------------------------|-----------------|
| CPA       | 0,66882                  | 0,69980         |
| PRM       | 21,68804                 | 0,73122         |

O algoritmo de CPA obteve um menor tempo para ser executado e gerou uma trajetória mais otimizada que o PRM, como pode ser observado na Tabela 1. A importância de se conhecer o tempo gasto para execução do algoritmo é fundamental para uma implementação *on-line*, ou seja, em tempo real, para verificar se o algoritmo consegue realizar a prevenção de obstáculos em movimento (obstáculos dinâmicos). Este tempo é necessário para determinar o valor da velocidade máxima que o algoritmo pode ser implementado e evitar a colisão do manipulador com os obstáculos em movimento.

No CPA, se a intensidade do campo potencial aplicado é muito grande a probabilidade de colisão é alta, e a

rota formada pelo robô tende a ter oscilações devido à distância entre os obstáculos e à força do campo. No PRM a determinação de parâmetros deve ser cautelosa, visto que quando se utiliza poucos pontos o algoritmo tende a gerar um caminho mais longo. No entanto, quando é utilizado com muitos pontos a trajetória formada tende a ser menor, aumentando a possibilidade de colisão, pois o manipulador se aproxima muito do obstáculo.

## 5. CONCLUSÕES

Os algoritmos PRM e CPA são eficientes em aplicações onde exigem-se trajetórias livres de colisão, com isso pode-se afirmar que este estudo traz contribuições importantes nesse contexto. Esse artigo comparou a utilização de dois algoritmos para a navegação livre de colisão de um manipulador SCARA. No CPA, a rota determinada para o robô é baseada em um campo de forças que influencia a trajetória do robô, já no PRM o trajeto destinado ao manipulador é baseado em plotagem aleatória de pontos, os quais serviram como referência para a determinação da trajetória.

Na elaboração da trajetória a partir do mapa usado, ambos os algoritmos tiveram baixo custo computacional, para o CPA o tempo foi de 0,67747 s, enquanto que para o PRM foi de 0,98921 s. Os algoritmos ao concluírem a proposta, geraram uma trajetória de 0,9637 m para o CPA e de 1.06421 m para o PRM. Portanto, os dois algoritmos se mostraram satisfatórios na prevenção de colisão para o manipulador em estudo. No entanto, o algoritmo CPA apresentou melhor resultado, mesmo sem otimização dos parâmetros.

Como trabalhos futuros pretende-se: implementar os algoritmos em tempo real (*on-line*); melhorar o CPA, utilizando algoritmos para otimização dos seus parâmetros; implementar o PRM com inteligência computacional e validar os algoritmos no robô.

## REFERÊNCIAS

- Adorno, B.V., Borges, G.A., and por Computador, V. (2006). Um método de planejamento de trajetória para robôs móveis através de passeios aleatórios adaptativos e mapa de rotas. In *XVI Congresso Brasileiro de Automática*, 1–6.
- Batista, J., Souza, D., dos Reis, L., Barbosa, A., and Araújo, R. (2020a). Dynamic model and inverse kinematic identification of a 3-dof manipulator using rlspso. *Sensors*, 20(2), 416.
- Batista, J., Souza, D., Silva, J., Ramos, K., Costa, J., dos Reis, L., and Braga, A. (2020b). Trajectory planning using artificial potential fields with metaheuristics. *IEEE Latin America Transactions*, 18(05), 914–922.
- Batista, J.G., da Silva, J.L., and Thé, G.A. (2018). Can artificial potentials suit for collision avoidance in factory floor?
- Batista, J.G. et al. (2017). Um estudo de caso sobre a interação máquina-máquina na perspectiva dos algoritmos clássicos de geração de caminhos livres de colisão.
- Chen, F., Di, P., Huang, J., Sasaki, H., and Fukuda, T. (2009). Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly. In *2009 International Symposium on Micro-NanoMechatronics and Human Science*, 92–97. IEEE.
- Chen, Jinbao e Zhou, Y.e.G.J.e.D.Y. (2019). Um algoritmo de roteiro probabilístico aprimorado com função de campo potencial para o planejamento de caminhos do quadrotor. 3248–3253.
- Gonzalez, C. (2017). What’s the difference between industrial robots? URL <https://www.machinedesign.com/robotics/what-s-difference-between-industrial-robots/>.
- Guerin, C., Rauffet, P., Chauvin, C., and Martin, E. (2019). Toward production operator 4.0: modelling human-machine cooperation in industry 4.0 with cognitive work analysis. *IFAC-PapersOnLine*.
- Hartenberg, R. and Denavit, J. (1964). *Kinematic synthesis of linkages*. New York: McGraw-Hill.
- Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, 396–404. Springer.
- Latombe, J. (2012). *Robot Motion Planning*. The Springer International Series in Engineering and Computer Science. Springer US.
- LaValle, S.M. (2006). *Planning algorithms*. Cambridge university press.
- Mohanta, J. and Keshari, A. (2019). A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation. *Applied Soft Computing*, 79, 391–409.
- Orozco-Rosas, Ulisses e Montiel, O.e.s.R. (2019). Planejamento do caminho do robô móvel usando o campo de potencial artificial evolutivo da membrana. *Applied Soft Computing*, 77, 236–251.
- Sciavicco, L. and Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2011). *Robotics: Modelling, Planning and Control (Advanced Textbooks in Control and Signal Processing)*. Springer.
- Silva, J.L.N.d. et al. (2016). Projeto e realização de controle fuzzy para manipulador industrial tipo scara e sua avaliação de desempenho à luz da norma iso 9283.
- Souza, S. (2008). Planejamento de trajetória para um robô móvel com duas rodas utilizando um algoritmo a-estrela modificado.
- Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2020). *Robot modeling and control*. John Wiley & Sons.
- Volpe, R. and Khosla, P. (1990). Manipulator control with superquadric artificial potential functions: Theory and experiments. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(6), 1423–1436.
- Wisskirchen, G., Biacabe, B.T., Bormann, U., Muntz, A., Niehaus, G., Soler, G.J., and von Brauchitsch, B. (2017). Artificial intelligence and robotics and their impact on the workplace. *IBA Global Employment Institute*, 2012–2017.