

Prevenção de Colisão utilizando Campos Potenciais Artificiais com PSO (CPA+PSO)

Catarina G. B. P. Silva* Emerson V. A. Dias* Josias G. Batista*
Darielson A. Souza** José L. N. Silva* Geraldo L. B. Ramalho*
Marcello A. F. B. Lima*

* *Federal Institute of Ceará CE, (e-mail: catarinagbezerra@gmail.com, emersonverasifce@gmail.com, josiasbatista@ifce.edu.br, leonardo.silva@ifce.edu.br, gramalho@ifce.edu.br, marcello@ifce.edu.br).*

** *Department of Electrical Engineering, Federal University of Ceará, CE (e-mail: darielson@dee.ufc.br)*

Abstract: The use of robotics in the industry has grown a lot in recent years, making production systems more and more efficient. However, some limitations can delay the production process causing losses to production, such as the robot stops which can be caused by several factors, such as accidents, collisions of manipulator robots with operators or other equipment. The main contribution of this research is to improve the Artificial Potential Fields (APF) algorithm using Particle Swarm Optimization (PSO) through the optimization of APF parameters (APF+PSO) to avoid collisions. We present as results the trajectories in the Cartesian space and trajectories of the joints of the SCARA manipulator (Selective Compliance Assembly Robot Arm), as well as the torque of the joints calculated from the dynamic model, the computational cost of the APF+PSO algorithm is also presented to find the parameters of the APF algorithm and generate the collision avoidance trajectories.

Resumo: A utilização da robótica na indústria tem crescido muito nos últimos anos, tornando os sistemas de produção cada vez mais eficientes. No entanto, algumas limitações podem atrasar o processo produtivo causando prejuízos à produção, como as paradas do robô que podem ser causadas por vários fatores, como acidentes, colisões de robôs manipuladores com operadores ou outros equipamentos. A principal contribuição desta pesquisa é melhorar o algoritmo de Campos Potenciais Artificiais (*Artificial Potential Field - APF*) utilizando Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*) por meio da otimização dos parâmetros do APF (APF+PSO) para evitar colisões. Apresentamos como resultados as trajetórias no espaço cartesiano e trajetórias das juntas do manipulador SCARA (*Selective Compliance Assembly Robot Arm*), bem como o torque das juntas calculados a partir do modelo dinâmico, também é apresentado o custo computacional do algoritmo APF+PSO para encontrar os parâmetros do algoritmo APF e gerar as trajetórias livres de colisão.

Keywords: Trajectory generation; Artificial Potential Fields, Collision avoidance; Particle Swarm Optimization; SCARA manipulator.

Palavras-chaves: Geração de trajetória; Campos Potenciais Artificiais; Prevenção de colisão; Otimização por Enxame de Partículas, Manipulador SCARA.

1. INTRODUÇÃO

As mais diversas atividades atuais passaram a ser auxiliadas ou mesmo realizadas inteiramente por robôs. Com a crescente demanda de produção mundial, a robótica vem encontrando cada vez mais espaço nos processos industriais e, assim, foram desenvolvidos robôs para assistir ou mesmo substituir o homem em determinadas tarefas com eficiência e segurança (Dias et al., 2021).

O anseio da indústria é a interação homem-máquina, ou seja, o operador e a máquina trabalhando em conjunto (Dias et al., 2021). Todavia, as possibilidades de acidentes, principalmente, os colisivos, não podem ser desprezadas. A

fim de evitar essas colisões e traçar a melhor rota possível, faz-se uso de algoritmos de prevenção de colisão.

Neste trabalho é analisado o algoritmo de Campos Potenciais Artificiais (*Artificial Potential Field - APF*), que permite o planejamento de trajetórias em espaços desconhecidos para o robô (Pinto Jr et al., 2019). Com o objetivo de aprimorar o resultado encontrado com APF podemos usar metaheurísticas como Otimização por Enxame de Partículas (*Particle Swarm Optimization - PSO*). Metaheurísticas são métodos que resolvem de maneira genérica problemas de otimização. São usados de modo geral para a melhoria de uma função ou um algoritmo para os quais não se conhece os parâmetros eficientes (Engelbrecht, 2007), (Batista et al., 2020b).

O objetivo deste trabalho é apresentar o algoritmo de APF, na geração de trajetória livre de colisão de um manipulador SCARA (*Selective Compliance Assembly Robot Arm*) com PSO (APF+PSO). Essa metaheurística será utilizada para otimizar os parâmetros de repulsão do campo potencial do ponto inicial e dos obstáculos e o parâmetro de atração do ponto final que estão relacionados diretamente com a intensidade da força resultante do campo potencial. A motivação para a realização deste trabalho é a demora para conseguir encontrar os parâmetros de forma empírica tornando o processo dispendioso (Batista et al., 2018), (Batista et al., 2020b). Este artigo traz o algoritmo de APF clássico melhorado com PSO, no sentido de determinar os parâmetros de forma otimizada e rápida a partir de uma função custo, que é a soma das forças do algoritmo de APF.

2. FUNDAMENTAÇÃO TEÓRICA

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.

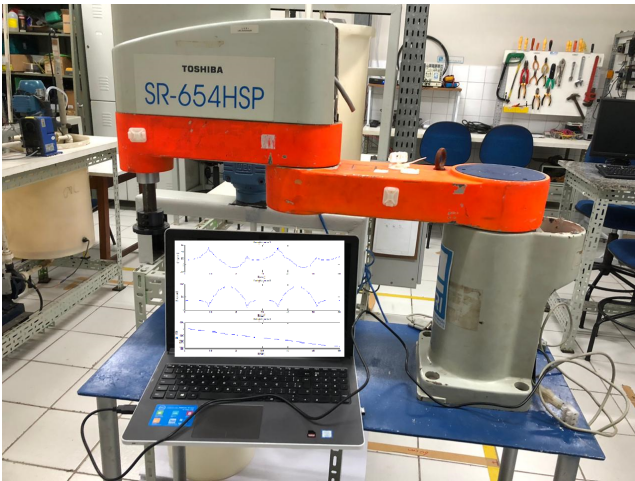


Figura 1. Manipulador robótico SCARA.

2.2 Cinemática inversa

Uma maneira de resolver o problema da cinemática inversa do robô é usando o método geométrico (Batista et al., 2020b), (Batista et al., 2020a). Partindo de métodos geométricos e aplicando algumas transformações trigonométricas encontra-se as equações da cinemática inversa que são dadas por:

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

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

em que $L_1 = 0,35$ e $L_2 = 0,30$, e são os valores dos comprimentos de cada junta do manipulador.

As equações (1) e (2) são as soluções do problema da cinemática inversa do manipulador SCARA são usadas para realizar a geração de trajetória do manipulador.

2.3 Dinâmica de um manipulador industrial

Aqui as equações dinâmicas de movimento para o manipulador são apresentadas utilizando a representação de Lagrange para o movimento que é dada por (Batista et al., 2020a), (Batista et al., 2018):

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta) = \tau, \quad (3)$$

em que $C(\theta, \dot{\theta}) \in \mathbb{R}^n$ é a matriz que descreve as forças centrífugas e de Coriolis e $G(\theta) = \frac{\partial g}{\partial \theta} \in \mathbb{R}^n$ é o vetor de gravidade. O valor adotado para a aceleração da gravidade aqui foi $g = 9,8 \text{ m/s}^2$ (Batista et al., 2018), (Batista et al., 2016).

Aplicando a formulação de Lagrange ao sistema será possível obter as equações da dinâmica para as juntas 1 e 2 do manipulador. Assim, obteremos a primeira equação de movimento que descreve o torque da junta 1, que será:

$$\tau_1 = [(m_1 + m_2)L_1^2 + m_2L_2^2 + 2m_2L_1L_2C_2]\ddot{\theta}_1 + [m_2L_2^2 + m_2L_1L_2C_2]\ddot{\theta}_2 - 2m_2L_1L_2S_2\dot{\theta}_1\dot{\theta}_2 - m_2L_1L_2S_2\dot{\theta}_2^2 + m_2gL_2C_{12} + (m_1 + m_2)gL_1C_1 \quad (4)$$

Podemos obter a primeira equação de movimento que descreve o torque da junta 2, que será:

$$\tau_2 = (m_2L_2^2 + m_2L_1L_2C_2)\ddot{\theta}_1 + m_2L_2^2\ddot{\theta}_2 + m_2L_1L_2S_2\dot{\theta}_1^2 + m_2gL_2C_{12} \quad (5)$$

Substituindo os valores de $L_1 = 0,35 \text{ m}$, $L_2 = 0,30 \text{ m}$, $m_1 = 7,872 \text{ kg}$, $m_2 = 4,277 \text{ kg}$ e $g = 9,8 \text{ m/s}^2$ em (4) e (5), temos:

$$\tau_1 = [1,873 + 0,898C_2]\ddot{\theta}_1 + [0,384 + 0,449C_2]\ddot{\theta}_2 - 0,769S_2\dot{\theta}_1\dot{\theta}_2 - 0,384S_2\dot{\theta}_2^2 + 12,574C_{12} + 41,671C_1, \quad (6)$$

and

$$\tau_2 = [0,384 + 0,449C_2]\ddot{\theta}_1 + 0,384\ddot{\theta}_2 + 0,449S_2\dot{\theta}_1^2 + 12,574C_{12}. \quad (7)$$

2.4 Campos Potenciais Artificiais

No método de Campos Potenciais Artificiais (*Artificial Potential Field - APF*) 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.

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 (8):

$$U(\theta) = U_{atr}(\theta) + U_{rep}(\theta), \quad (8)$$

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

$$F(\theta) = F_{atr}(\theta) + F_{rep}(\theta) \quad (9)$$

com

$$F_{atr}(\theta) = -\nabla[U_{atr}(\theta)] \quad (10)$$

e

$$F_{rep}(\theta) = -\nabla[U_{rep}(\theta)] \quad (11)$$

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).

2.5 Otimização por Enxame de Partículas

Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) é um algoritmo heurístico baseado no comportamento social de um bando de pássaros. Neste algoritmo os indivíduos são chamados de partículas e funcionam como um conjunto de pássaros que buscam por um formato de voo considerando a posição de cada partícula dentro do espaço (Paiva et al., 2017). A posição num instante t é atualizada por $x_i(t)$ e no tempo futuro $t + 1$ será dada por:

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (12)$$

onde $v_i(t)$ é a velocidade (Engelbrecht, 2007). Para esse problema utilizou-se o PSO global (*Global best PSO*), no qual a velocidade da partícula é atualizada por:

$$v_{ij}(t + 1) = v_{ij}(t) + c_1 r_1(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_2(t)[\hat{y}_{ij}(t) - \hat{x}_{ij}(t)] \quad (13)$$

em que $v_{ij}(t)$ é a velocidade da partícula, numa dada dimensão no instante t ; c_1 e c_2 são os parâmetros de aceleração. A informação da melhor partícula é dada por \hat{y}_{ij} e y_{ij} é a melhor posição desde o começo (Engelbrecht, 2007).

3. METODOLOGIA

3.1 Implementação do algoritmo APF

O campo potencial é construído para que o robô seja atraído para a configuração final Q_{final} e seja repellido pelos locais determinados como obstáculos (Spong et al., 2020).

Força de repulsão da posição inicial

Com a definição da posição inicial, uma força de repulsão pode ser gerada na mesma. A função adotada para representar o campo potencial de repulsão da posição inicial foi:

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

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 APF; x_0 e y_0 são os pontos iniciais onde o robô irá iniciar a trajetória.

Força de atração da posição final

A função adotada que representa o campo potencial de atração para posição final foi adotada como:

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

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 APF; x_f e y_f são os pontos finais da trajetória gerada.

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:

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

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. x e y são os pontos de referência para o APF; x_{oi} e y_{oi} são os pontos que definem a posição do obstáculo.

3.2 Algoritmo PSO

Realizou-se várias simulações do algoritmo do PSO, variando o número de iterações (gerações) e também o número de partículas. O melhor resultado do algoritmo foi para 30 partículas e para 50 iterações que apresentou os melhores valores para o APF. Para o algoritmo do PSO foram utilizados os seguintes parâmetros:

- Quantidade de partículas = 30 partículas;
- Parâmetros cognitivos e sociais (taxas de aprendizado - c_1 e c_2) = 2;
- Iterações = 50;
- Fator de inércia (w) = 0,5;
- Geração da população inicial = usado um rand numa equação genérica que se restringe ao intervalo [-1, 1].

O pseudocódigo para o algoritmo PSO, é apresentado a seguir.

Algoritmo PSO

```

1: inicialize a nuvem de partículas;
2: repita
3: para  $i = 1$  até  $m$ 
4: se  $f(x_i) < f(p_i)$  então
5:  $p_i = x_i$ ;
6: se  $f(x_i) < f(g)$  então
7:  $g = x_i$ ;
8: fim se
9: fim se
10: para  $j = 1$  até  $n$ 
11:  $r_1 = rand()$ ,  $r_2 = rand()$ ;
12:  $v_{ij} = wv_{ij} + c_1 r_1 (p_i - x_{ij}) + c_2 r_2 (g_j - x_{ij})$ ;
13: fim para
14:  $x_i = x_i + v_i$ ;
15: fim para
16: até satisfazer o critério de parada
    
```

O critério de parada utilizado no algoritmo PSO foi o número de gerações do algoritmo.

3.3 APF+PSO

A ação total do campo potencial, para o manipulador, é obtida pela soma de cada força em sua respectiva direção (Batista et al., 2020b), ou seja

$$F(x, y) = F_{xya}(x, y) + F_{xyr}(x, y) + F_{xyo}(x, y). \quad (17)$$

A equação (17) foi utilizada como função de avaliação (*fitness*) para o algoritmo do PSO. A equação (17) foi minimizada para encontrar a melhor força do campo potencial e a partir daí serem determinados os valores dos parâmetros K_r , K_a e K_o .

3.4 Cenário utilizado

Para a geração de trajetória do manipulador SCARA, livre de colisão, foi considerado o espaço cartesiano (XY). A partir dos pontos obtidos pelo algoritmo APF+PSO, resolve-se a cinemática inversa, e encontram-se as posições de cada junta do manipulador. O caminho gerado deve evitar a colisão de dois obstáculos mostrados na Figura 2, com tamanhos iguais cujo raio é igual a $0,2\text{ m}$.

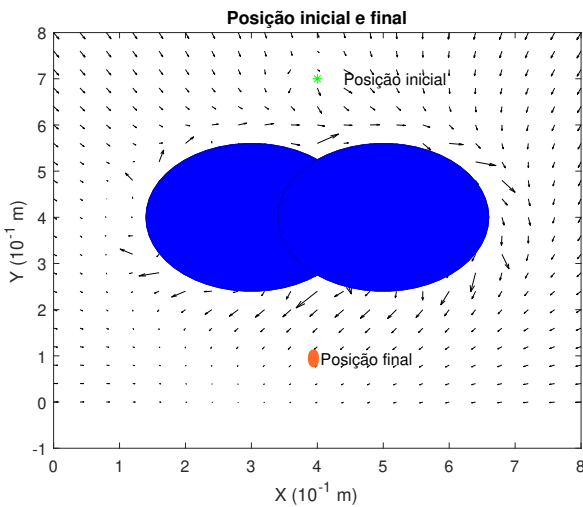


Figura 2. Cenário utilizado para a implementação do algoritmo APF+PSO.

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

Um ponto importante com relação à metodologia adotada está no fato de que a partir das técnicas utilizadas, o obstáculo aqui apresentado tem o formato de um círculo, reduzindo a geração de mínimos locais no momento do planejamento de caminhos para o algoritmo de APF, além do fato de tornar o sistema capaz de analisar obstáculos de vários formatos como triangular, retangular e trapezoidal.

As simulações do algoritmo (20 repetições) foram realizadas em um computador com um processador Core i7 - 7^a Geração, com velocidade de processamento de $2,90\text{ GHz}$ e com memória RAM de $8,00\text{ GB}$. Os algoritmos foram implementados na linguagem M-código com algumas funções específicas para a linguagem.

4. RESULTADOS

4.1 Resultados do APF+PSO

Na Figura 3, a seguir, tem-se o gráfico das aptidões e gerações do PSO, onde foi utilizado no algoritmo 30 partículas e 50 iterações, tem-se uma convergência entre a melhor partícula, a solução e a média de todas as partículas.

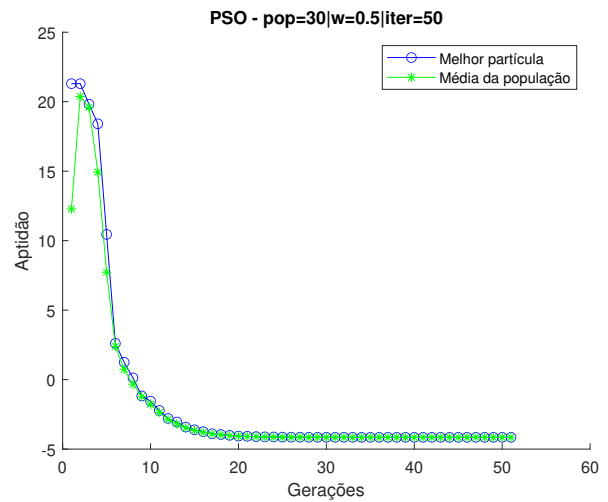


Figura 3. Gráfico do PSO convergindo para 30 partículas e 50 iterações.

Os parâmetros do campo potencial encontrados na simulação do algoritmo do PSO para 30 partículas e 50 iterações, estão apresentados na Tabela 1 a seguir.

Tabela 1. Valores dos parâmetros do APF encontrados pelo PSO para 30 partículas e 50 iterações.

Parâmetro	K_r	K_a	K_o
Intensidade	25,13	30,05	4,52

A Figura 4 apresenta as linhas do campo potencial artificial e a trajetória livre de colisão, no espaço cartesiano gerados pelo algoritmo APF+PSO utilizando os parâmetros da Tabela 1, obtidos a partir das localizações dos obstáculos.

Na Figura 5 é apresentada a superfície resultante do campo potencial para a cena mostrada na Figura 4.

4.2 Trajetórias das juntas e torques

As figuras 6, 7 e 8 mostram as trajetórias, velocidades e acelerações, respectivamente, das juntas 1 e 2, do manipulador sem colisão, apresentada na Figura 6.

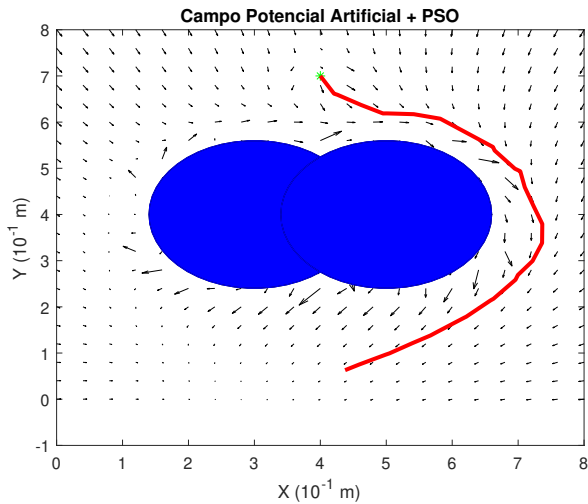


Figura 4. Linhas do campo potencial e trajetória realizada pelo algoritmo de APF+PSO para os parâmetros da Tabela 1.

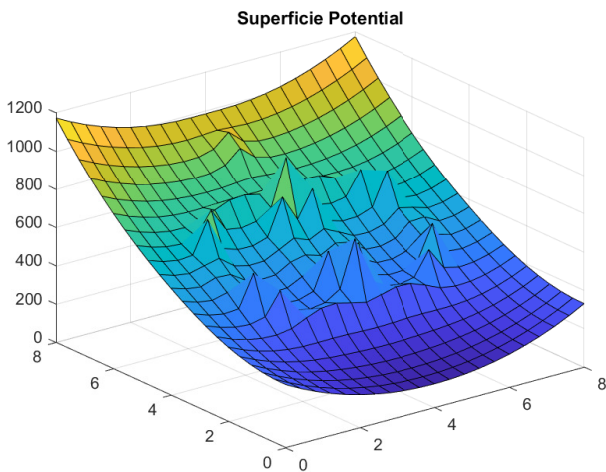


Figura 5. Superfície resultante do algoritmo de APF da cena da Figura 4.

A Figura 9 apresenta os torques das juntas 1 e 2 do manipulador calculados a partir do modelo dinâmico e as trajetórias (Figura 6), velocidades (Figura 7) e acelerações (Figura 8) anteriormente apresentadas.

4.3 Discussões

Uma comparação com relação ao tempo médio de processamento com 20 repetições foi realizada afim de verificar o tempo de cada algoritmo. A tabela 2 apresenta o tempo de processamento do algoritmo APF+PSO para encontrar os parâmetros do APF e gerar a trajetória livre de colisão.

Com a aplicação do algoritmo de PSO consegue-se uma melhoria do algoritmo de APF, pois os parâmetros do APF são muitas vezes adotados de forma empírica, tornando o processo dispendioso. Com a utilização do algoritmo de PSO conseguiu-se aqui uma melhoria do algoritmo de APF para gerar trajetórias livres de colisão, pois os parâmetros são encontrados de forma otimizada. Os

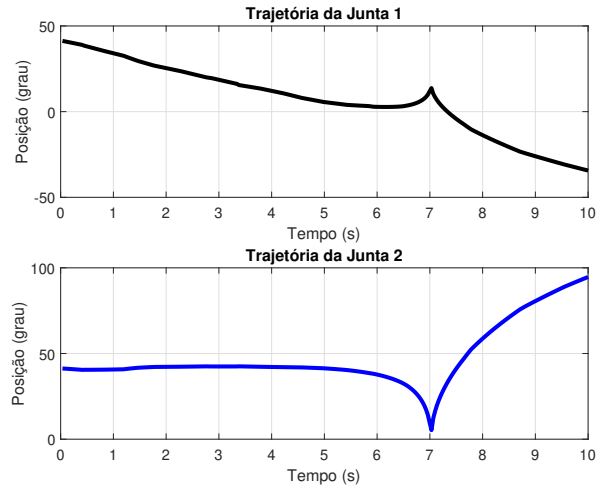


Figura 6. Trajetórias das juntas 1 e 2 a partir do caminho gerado na Figura 4.

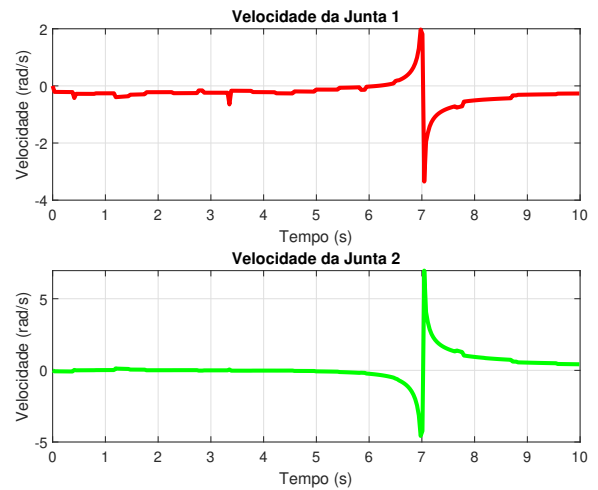


Figura 7. Velocidades das juntas 1 e 2 para a trajetória apresentada na Figura 6.

Tabela 2. Tempo médio de processamento do algoritmo APF+PSO para a geração da trajetória livre de colisão.

Algoritmo	Proc. Médio (s)
PSO	2,5791
APF	0,6636
APF+PSO	3,2427

algoritmos apresentam baixo tempo de processamento e precisão dos pontos das trajetórias alcançadas.

5. CONCLUSÃO

Pode-se concluir que os resultados do método de APF+PSO, apresenta baixo tempo de processamento do algoritmo, permitindo aplicações em ambientes dinâmicos para trajetórias livres de colisão. O método de APF requer ajustes de vários parâmetros para que a trajetória gerada seja satisfatória. Como mostrado neste trabalho, através da seleção adequada dos parâmetros K_a , K_r e K_o pode-se atingir a posição e otimizar o caminho percorrendo uma

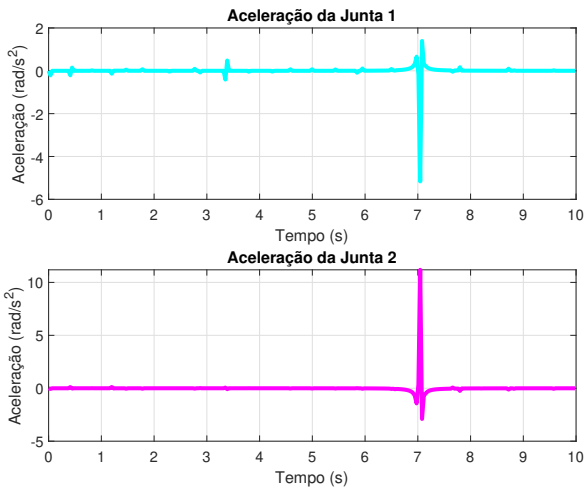


Figura 8. Acelerações das juntas 1 e 2 para as trajetórias da Figura 6.

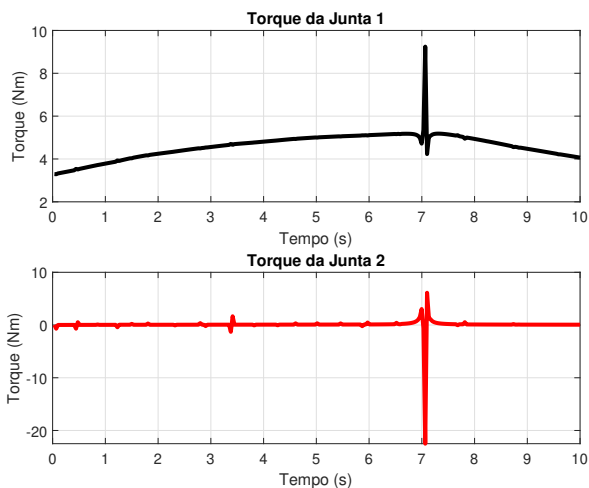


Figura 9. Torque das juntas 1 e 2 calculadas a partir das trajetórias, velocidades e acelerações (figuras 6, 7 e 8).

distância mínima. Conclui-se aqui que o algoritmo proposto pode ser utilizado de forma satisfatória na aplicação aqui apresentada.

Este trabalho representou uma contribuição pelo fato de aplicar uma técnica consagrada na literatura, APF melhorado com PSO para geração de trajetória de manipuladores industriais. Os autores estão pesquisando outras formas de implementação melhoradas do algoritmo de APF, com sistemas mais robustos não lineares como Redes Neurais Artificiais, Mínimos Quadrados, Filtro de Kalman.

AGRADECIMENTOS

Os autores agradecem ao Programa Institucional de Bolsas de Iniciação Científica - PIBIC IFCE / CNPq / FUNCAP, Edital N^o 1/2020 - PRPI / Reitoria - IFCE, pelo auxílio-bolsa.

REFERÊNCIAS

Batista, J., Souza, D., dos Reis, L., Barbosa, A., and Araújo, R. (2020a). Dynamic model and inverse kine-

matic 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? *ICINCO 2018*.

Batista, J.G., da Silva, J.L., Pereira, N.S., et al. (2016). Modelagem matemática e simulação computacional da dinâmica de um robô scara. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 4(1).

Dias, E.V., Silva, C.G., Batista, J.G., Ramalho, G.L., Costa, J.R., Silva, J.L., and Souza, D.A. (2021). Prevenção de colisão de um manipulador scara utilizando campos potenciais artificiais e caminhos probabilísticos. *Brazilian Journal of Development*, 7(1), 11252–11270.

Engelbrecht, A.P. (2007). *Computational intelligence: an introduction*. John Wiley & Sons.

Latombe, J. (2012). *Robot Motion Planning*. The Springer International Series in Engineering and Computer Science. Springer US.

Paiva, F.A.P., Costa, J.A.F., and Silva, C.R.M. (2017). A serendipity-based approach to enhance particle swarm optimization using scout particles. *IEEE Latin America Transactions*, 15(6), 1101–1112.

Pinto Jr, U.d.M., Carvalho, M.P., and CONCEICAO, A.G.S. (2019). Campos potenciais artificiais aplicado ao planejamento de trajetórias do braço robótico jaco. *Anais da Sociedade Brasileira de Automática*, 1(1).

Sciavicco, L. and Siciliano, B. (2012). *Modelling and control of robot manipulators*. Springer Science & Business Media.

Spong, M.W., Hutchinson, S., and Vidyasagar, M. (2020). *Robot modeling and control*. John Wiley & Sons.