

Trajectory Planning Using Artificial Potential Fields with Metaheuristics

J. Batista, D. Souza, J. Silva, K. Ramos, J. Costa, L. dos Reis and, A. Braga

Abstract—The use of industrial robots has grown over the years, making production systems increasingly efficient. Within this context, some limitations appear that can delay the productive process causing damages to the production. These limitations are robot stops, for example. Stops can be caused by various 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 Field (APF) using Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Differential Evolution (DE) by optimizing the APF parameters in collision avoidance. We present as results: the trajectories generated by a planar manipulator robot; the position errors between the final position and the last position of the generated trajectories; and the computational cost of the PSO, GA and DE algorithms to find the parameters of the APF algorithm.

Index Terms—Trajectory generation, Artificial potential fields, Collision avoidance, Particle swarm optimization, Genetic algorithm, Differential evolution.

I. INTRODUÇÃO

NO intuito de suprir a crescente atividade produtiva mundial, as indústrias dos mais diversos setores têm encontrado na modernização e na automação de seu chão de fábrica importantes aliados. Um dos aliados em especial, têm se tratado da utilização de robôs na realização de tarefas. De fato, a utilização de robôs em tarefas de manipulação teve um crescimento significativo no contexto de produção industrial nos últimos anos[1].

A referida difusão dos robôs em ambiente industrial propiciou, ao longo dos anos, que vários métodos fossem desenvolvidos com o intuito de monitorar e controlar robôs móveis ou robôs manipuladores. Com isto, os mesmos adquiriram a capacidade de operar em ambientes perigosos para os seres humanos como, por exemplo, além da atmosfera terrestre, em explorações aquáticas, no transporte de materiais, entre outros [2].

O problema da prevenção de colisão que se põe é o de não apenas detectar uma “travessia de fronteira” e não apenas tomar a decisão de desligar o robô, mas permitir que o agente de automação possa estar na cena, coexistindo em modo harmônico, evitando a colisão com outros equipamentos ou outros robôs. Isto requer a implementação de um algoritmo que detecte um obstáculo em movimento e, antevendo colisão, evite-a. Para este fim, existem duas abordagens tradicionais

para a geração de caminhos livres de colisão, são elas: os algoritmos que utilizam Campos Potenciais Artificiais - CPA (*Artificial Potential Fields - APF*) [3] e aqueles baseados em Mapas de Rotas Probabilísticos (PRM - *Probabilistic Roadmap Methods*) [4].

Campos Potenciais Artificiais tem sido bastante utilizados por diversos pesquisadores em várias aplicações para geração de trajetória livre de colisão. Dentre as aplicações, pode-se citar: veículos aéreos não tripulados (VANT), robôs móveis, robôs manipuladores industriais, robótica com aplicação na medicina, carros autônomos, dentre outras [5], [6].

CPA podem ser melhorados com a utilização de outros algoritmos, como a utilização de metaheurísticas para otimizar as constantes desse algoritmo. Metaheurísticas são métodos heurísticos para resolver de forma genérica problemas de otimização. São geralmente aplicadas para otimizar uma função ou um algoritmo para os quais não se conhece os parâmetros eficientes [7].

O objetivo deste trabalho é apresentar a simulação do algoritmo de campos potenciais, na geração de trajetória livre de colisão de um manipulador SCARA (*Selective Compliance Assembly Robot Arm*), com Otimização por Enxame de Partículas (PSO), Algoritmo Genético (GA) e Evolução Diferencial (DE). Essas metaheurísticas serão utilizadas para otimizar as constantes de repulsão do campo potencial do ponto inicial e dos obstáculos e a constante de atração do ponto final que estão relacionadas diretamente com a intensidade da força resultante do campo potencial, ou seja elas definirão o quanto melhor ou pior é o campo potencial na geração de trajetórias livres de colisão. Como resultados, são apresentados: as trajetórias livres de colisão, o erro médio entre ponto final da trajetória gerada pelo CPA e ponto final de destino e custo computacional de cada algoritmo.

A. Contribuições

As motivações para a realização deste trabalho são: a demora para conseguir encontrar os parâmetros de forma empírica tornando o processo dispendioso [5]; algumas abordagens usam-se busca exaustiva para ponderar parâmetros, o seu ponto negativo é o custo computacional, causando estouro de pilha e o tempo limite excedido ao se comparar com atividades reais.

Este artigo traz uma melhoria no algoritmo de CPA clássico no sentido de determinar os parâmetros de forma otimizada, e trajetórias mais suaves pois as metaheurísticas vão encontrar os parâmetros de forma mais rápida a partir de uma função que é a soma das forças do algoritmo de CPA.

Josias G. Batista, José L. N. Silva, Jonatha R. Costa - Instituto Federal do Ceará-IFCE, Fortaleza, CE - Brazil: Josias Guimarães (e-mail: josi-asgb@dee.ufc.br/josiasbatista@ifce.edu.br).

Kaio M. Ramos, Darielson A. Souza and Laurinda L. N. Reis - Departamento de Engenharia Elétrica, Universidade Federal do Ceará - UFC, Campus Pici, Fortaleza-CE, Brasil.

Este artigo está organizado da seguinte maneira. A Seção II apresenta a fundamentação sobre o manipulador SCARA, CPA, GA, PSO e DE. Na Seção III é apresentada a revisão bibliográfica utilizada neste trabalho. A Seção IV apresenta a metodologia do trabalho, como foram utilizadas as meta-heurísticas com o CPA. As simulações e resultados obtidos são apresentados na Seção V. Finalmente, as conclusões e trabalhos futuros são mencionados na Seção VI.

II. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentadas os assuntos sobre: os detalhes sobre o manipulador SCARA, Campos Potenciais Artificiais, Otimização por Enxame de Partículas, Algoritmos Genéticos e Evolução Diferencial.

A. Manipulador SCARA

O manipulador SCARA utilizado neste trabalho é um robô de quatro graus de liberdade (GDL), sendo que aqui será trabalhado com dois GDL, a Fig. 1 mostra o robô. Por ser um robô tipo SCARA de 4 GDL, as duas primeiras juntas, a partir da base, são de rotação em torno de eixos verticais e trabalhando portanto num plano horizontal (plano xy), como se fosse um robô de 2 GDL planar.

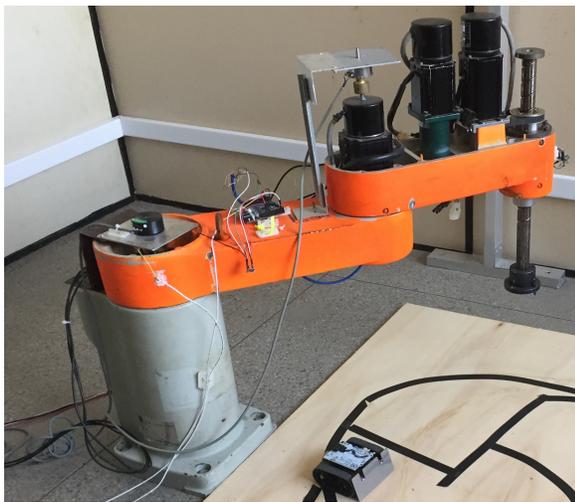


Fig. 1. Manipulador robótico tipo SCARA.

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

B. Campos Potenciais Artificiais

O método de CPA foi introduzido por Khatib em 1986 [3], onde sua pesquisa foi aplicada, inicialmente, em robôs manipuladores com um ponto alvo e com obstáculos estáticos e em movimento. Neste método, considera-se uma posição de um ponto que se move num campo de forças. O ponto final fornece uma força atrativa e os obstáculos, forças repulsivas. No método de campos potenciais artificiais o manipulador se move em um campo de forças [3].

Apesar deste método ter sido inicialmente utilizado em robôs manipuladores, atualmente, a maioria das aplicações são na robótica móvel [8]. O método de CPA deve o seu nome ao fato de que um campo (vetorial) de forças $F(x)$ ser derivado de um campo (escalar) potencial $U(x)$ [9].

1. Planejamento de trajetória utilizando campos potenciais artificiais

A utilização de campos potenciais artificiais consiste de um ponto inicial $q_{inicial}$, que gera uma força repulsiva, o alvo q_{final} , fornece uma força atrativa e os obstáculos, forças repulsivas.

O método de campos potenciais artificiais é devido ao campo (vetorial) de forças $F(\theta)$ ser derivado de um campo (escalar) potencial $U(\theta)$.

$$F(x) = \nabla[U(\theta)] \quad (1)$$

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 [10]. 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 (2):

$$U(\theta) = U_{atr}(\theta) + U_{rep}(\theta) \quad (2)$$

onde $U(\theta)$ 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:

$$F(\theta) = F_{atr}(\theta) + F_{rep}(\theta) \quad (3)$$

com

$$F_{atr}(\theta) = -\nabla[U_{atr}(\theta)] \quad (4)$$

e

$$F_{rep}(\theta) = -\nabla[U_{rep}(\theta)] \quad (5)$$

onde $F(\theta)$ 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 $U_{rep}(\theta)$ [11]. 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 [12].

2. 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 [13]:

$$U_r(x, y) = \frac{K_r}{\sqrt{(x - x_0)^2 + (y - y_0)^2}} \quad (6)$$

onde 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. 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 [8]:

$$U_a(x, y) = \frac{1}{2}K_a[(x - x_f)^2 + (y - y_f)^2] \quad (7)$$

onde 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.

4. 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 [8]:

$$U_o(x, y) = \frac{K_o}{\sqrt{(x - x_{oi})^2 + (y - y_{oi})^2}} \quad (8)$$

onde 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 [8]. 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.

C. 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. O método foi proposto por Eberhart e Kennedy em 1995 [14], tem como objetivo buscar a solução ótima, em um espaço de busca, através da troca de informações entre indivíduos de uma população determinando qual trajetória cada um deles deverá tomar no espaço de busca. O PSO possui algumas vantagens tais como facilidade de implementação e rápida convergência, porém muitas vezes enfrenta um problema em que as suas partículas ficam “presas” em ótimos locais. Esse problema é muitas vezes chamado de convergência prematura [15].

Neste algoritmo, os indivíduos, chamados agora de partículas funcionam como um conjunto de pássaros que buscam por um formato de voo considerando a posição de cada partícula dentro do espaço, isso é em decorrência de uma curva de evolução dentro da sociedade ou conjunto de partículas. Cada partícula terá seu sucesso definido por tendência geral da

população [15]. A posição num instante t é atualizada por $x_i(t)$ e no tempo futuro $t + 1$ será dada por:

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (9)$$

onde $v_i(t)$ é a velocidade [7]. Cada partícula apresentará um componente cognitivo, que será uma relação da distância entre ela mesma e o melhor (*optimal solution*), além da componente social que é o entendimento do conjunto sobre a existência de uma dada partícula. Para esse problema utilizou-se o PSO global (*Global best PSO*), no qual a velocidade da partícula é atualizada por:

$$v_{ij}(t + 1) = v_{ij}(t) + c_1r_1(t)[y_{ij}(t) - x_{ij}(t)] + c_2r_2(t)[\hat{y}_{ij}(t) - \hat{x}_{ij}(t)] \quad (10)$$

para $v_{ij}(t)$ sendo a velocidade da partícula, numa dada dimensão no instante t . Novamente, 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 [7].

Ao contrário de outras técnicas de computação evolutiva, no algoritmo PSO, cada partícula está associada a uma velocidade. Partículas voam por meio do espaço de busca com velocidades que são ajustados dinamicamente de acordo com seus comportamentos históricos. Finalmente, as partículas têm uma tendência a percorrer a melhor área de pesquisa ao longo do processo de busca [7]. As trajetórias das juntas do manipulador podem ser otimizadas utilizando uma estratégia que divide o robô em várias funções de aptidão para realizar o ajuste ideal com PSO modificado com operador de crossover [16].

D. Algoritmos Genéticos

O algoritmo Genético é comumente usado para encontrar soluções de problemas com um grande espaço de busca, problemas com muitas restrições e com diversos parâmetros com grande possibilidade de combinações [17]. Isto se aplica ao problema em questão pois serão encontrados os parâmetros que irão melhorar o CPA, para gerar uma trajetória otimizada. A ideia ao se utilizar GA é que ele seja capaz de evoluir uma população de soluções candidatas de um determinado problema, de modo que um operador inspirado por variação genética natural e seleção natural encontrará melhor solução [17]. A Fig. 2 sintetiza o processo de busca e escolha da melhor solução.

Cada indivíduo gera uma solução de acordo com a busca para satisfazer os valores dos parâmetros. A partir disso, será avaliado como o melhor indivíduo evolui ao longo da busca; entende-se por melhor indivíduo aquele que tem menor custo (*fitness*). Os indivíduos com melhores avaliações serão selecionados para gerar novos indivíduos, formado, assim, uma nova geração de soluções [18].

Quando gera-se um conjunto de soluções (população) utilizando algoritmo genético, esses resultados são avaliados individualmente, recebendo uma nota de acordo com a necessidade e limitações impostas pelo projeto. Para fazer essa avaliação, o GA utiliza uma relação de *fitness* onde serão encontrados os melhores parâmetros.

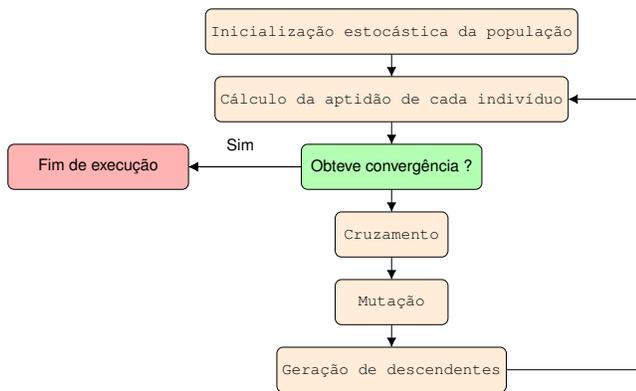


Fig. 2. Passos para implementação do GA.

E. Evolução Diferencial

O algoritmo Evolução Diferencial (do inglês *Differential Evolutionary* - DE é semelhante ao Algoritmo Genético - GA), diferenciando-se pelo fato de que o dado tratado, os indivíduos de uma suposta população representam um distanciamento de uma certa população inicial. A posição do indivíduo no conjunto tem preponderância sobre o que costuma-se chamar de *fitness function* (aptidão). A população inicial é definida de forma pseudoaleatória e uniforme. A seleção aqui usada é do tipo torneio.

A equação para o algoritmo DE pode ser vista a seguir:

$$u_i(t) = x_{i1}(t) + F(x_{i2}(t) + x_{i3}(t)) \quad (11)$$

onde x são os vetores do algoritmo; $i_1, i_2, i_3 \in 1, 2, \dots$ são os índices mutuamente distintos; F é uma constante que determina o tamanho do passo a ser dado na direção definida pelo vetor diferença.

III. REVISÃO DA LITERATURA

Os trabalhos apresentados por [19] e [31] destacam a utilização de CPA para geração de trajetórias e desvio de obstáculos em veículos aéreos não tripulados (VANT). Estes trabalhos utilizam o método de campo potencial, uma vez que tem uma estrutura simples, baixa complexidade computacional e fácil de implementar, além de ser utilizado na navegação autônoma em ambientes conhecidos ou não. [19] citam alguns métodos propostos para resolver problemas de desvio de obstáculos, tais como sistema de visão computacional, radar, Sistema de Posicionamento Global (GPS), digitalização de laser, entre outros, porém utilizou-se CPA, pois apresenta simplicidade para implementação e análise matemática. O CPA foi aplicado em um quadricóptero para evitar colisões, pois voa em baixa altitude.

As pesquisas realizadas por [20] e [21] propõem estratégias onde o método de CPA é utilizado com outros métodos para resolver o problema de mínimos locais em navegação de robôs móveis em ambientes de exploração. Um método usa o CPA melhorado com base em um campo simultâneo, onde as funções potenciais são utilizadas para calcular o caminho otimizado e encurtar a distância do percurso planejado. Uma outra abordagem é usar o campo vetorial calculado a partir

da solução numérica definida como Problema do Valor Limite (PVL) que gera um caminho suave e livre de mínimos locais.

No trabalho de [22], os autores aplicaram CPA para gerar um caminho livre de colisões para um manipulador móvel desviar de obstáculos estáticos em um ambiente conhecido. Essa aplicação apresenta novos desafios pois a dificuldade de geração de trajetória aumenta com a quantidade de GDL do manipulador. Além disso, a utilização de manipuladores com muitos GDL envolve outros problemas, tais como modelar seu ambiente de trabalho (geralmente desestruturado, desconhecido ou parcialmente conhecido). Outro trabalho, de simulação, apresentado por [23] é a aplicação de CPA com *Fuzzy* em um robô planar paralelo para evitar colisão entre suas partes mecânicas (*links* e a estrutura do robô) e obstáculos em seu espaço de trabalho.

As pesquisas realizadas por [24] e [25] apresentam trabalhos onde utiliza o método de CPA em um robô assistente de cirurgia de implante dentário. No primeiro trabalho o CPA é aplicado em um manipulador de cinco GDL onde manipulador é adaptado para encontrar a posição e ajuste do ângulo. No segundo trabalho, foi utilizado um sistema de navegação baseado em visão estéreo implementado em um robô de três GDL. Os trabalhos se propõem a três requisitos, que são: 1) o usuário pode liberar ou mover o manipulador quando estiver distante do destino; 2) o manipulador irá aplicar a força de ajuste para convergir para o destino; 3) o manipulador vai aplicar a força necessária para evitar colisões ao passar do ponto de destino.

O algoritmo de PSO pode ser utilizado como um método de planejamento de trajetória usando uma função de valor de estado e uma Otimização Iterativa por Enxames de Partículas (IPSO) [26], [27]. No trabalho de [28] foi utilizado Algoritmo Genético (AG) na geração de trajetória para um robô de 8 GDL que pode andar em terreno plano e subir escadas com deformação no solo. O método proposto usa *splines* para modelar os ângulos de cada junta com uma única camada do AG, o que a torna mais rápida e simples do que outros modelos. O método incorpora a dinâmica de um robô 8 GDL real para encontrar o máximo de energia ideal. Pode-se citar [29], onde utilizaram AG para filtragem de dados para fazer uma primeira seleção de genes de DNA. Um AG combinado com um recozimento simulado e um classificador SVM (Máquina de Vetor de Suporte), depois uma segunda seleção e classificação são realizadas a partir de um subconjunto de genes. Com a combinação dessas técnicas, foi possível melhorar taxa de classificação obtida com um subconjunto de genes pequenos.

Na pesquisa de [30], foi utilizada Evolução Diferencial (*Differential Evolution* - DE) com estratégia para encontrar a solução ótima de trajetórias para um manipulador de 7 GDL em um espaço qualquer. Os resultados da simulação são apresentados para planejamento de trajetória do manipulador de 7 GDL utilizando a cinemática redundante montado em um espaço flutuante.

Como apresentado nas referências, a utilização de CPA é bem ampla na geração de caminhos livre de colisão, bem como as metaheurísticas utilizadas neste trabalho, que são PSO, AG e DE.

IV. METODOLOGIA

O planejamento de trajetória pode ser tratado como um problema de otimização, isto é, encontrar o mínimo global em U , a partir da configuração inicial $\theta_{inicial}$. Um dos algoritmos mais fáceis para resolver este problema é através do gradiente. Neste caso, o gradiente negativo de U pode ser considerado como uma força que atua sobre o robô, ou seja está ligado diretamente ao movimento das juntas [9]. De maneira geral, o campo U é um campo resultante de uma adição onde uma parcela constitui a atração e a outra parcela constitui a repulsão, como mostrado na equação (2). A força resultante dos campos potenciais artificiais que deve ser aplicada no robô é dada conforme a equação (3).

A ação total do campo potencial, para o manipulador planar, é obtida pela soma de cada força em sua respectiva direção [8], ou seja

$$F(x, y) = F_{xya}(x, y) + F_{xyr}(x, y) + F_{xyo}(x, y) \quad (12)$$

A equação (12) foi utilizada como função de avaliação (*fitness*) para realizar a simulação do algoritmo do PSO, do GA e DE. A equação (12) 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 .

Para a geração de trajetória do manipulador SCARA (*Selective Compliance Assembly Robot Arm*), livre de colisão, foi considerado o espaço cartesiano (xy), pois para realizar a implementação do caminho resolve-se a cinemática inversa, a partir dos pontos obtidos pelo CPA, em seguida, pode ser aplicado em qualquer manipulador que tenha o espaço de trabalho compatível. O caminho gerado deve evitar a colisão de dois obstáculos circulares, com tamanhos iguais cujo raio é igual a 2 cm. O cenário para a simulação é formado pela posição inicial do robô, posição final do robô e o obstáculo (círculos). As posições são dadas no sistema de coordenadas cartesianas.

Foram utilizados para a implementação do algoritmo a posição inicial do robô, no espaço cartesiano, (0, 4) e a posição final (0, -4). Para os obstáculos utilizou-se as posições (1,2; 0) para o obstáculo 1 (que se encontra mais à esquerda), e (0; -1,5) para o obstáculo 2 (que se encontra mais à esquerda) 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 CPA, conforme indica Volpe e Khosla (1990) [32], 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.

A. Algoritmo PSO

Realizou-se várias simulações do algoritmo 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 apresentaram os melhores valores para o CPA e teve uma convergência entre a melhor partícula, a solução e a média de todas as partículas.

Para o algoritmo do PSO foram utilizados os seguintes valores dos elementos:

- 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 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:  $v_{ij} = wv_{ij} + c_1r_1(p_i - x_{ij}) + c_2r_2(g_j - x_{ij})$ ;
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.

B. Algoritmo Genético (GA)

Para várias simulações com o GA, utilizou-se os seguintes valores dos parâmetros do algoritmo, que foram os que apresentaram o melhor desempenho:

- Tamanho da população = 30 indivíduos;
- Probabilidade de cruzamento = 0.8;
- Probabilidade de mutação = 0,01;
- Número de gerações = 50;

O algoritmo GA, é apresentado a seguir:

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

C. Algoritmo DE

Para o algoritmo de DE utilizou-se os seguintes valores dos parâmetros do algoritmo:

- Tamanho da população (NP) = 30 indivíduos;
- Número de gerações = 50 gerações
- Probabilidade do cruzamento (CR) = 0,9;
- Tamanho do passo a ser dado na diferença (F) = 0,5.

O algoritmo DE, é apresentado a seguir:

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

Algoritmo GA

- 1: **inicializa** população P ;
- 2: **repita**
- 3: **selecione** uma subpopulação P' ;
- 4: **para** $i \leftarrow 1$ **até** cruzamento CR **faça**
- 5: **escolha** $S_1, S_2 \in P'$, aleatoriamente;
- 6: filho \leftarrow cruzamento (S_1, S_2) ;
- 7: **se** $f(S_1) \geq f(S_2)$ **então** $S_{aux} \leftarrow S_1$;
- 8: **senão** $S_{aux} \leftarrow S_2$;
- 9: **se** $f(S_{aux}) \leq f(\text{filho})$ **então**
- 10: filho substitui S_{aux} em P ;
- 11: **fim-se**;
- 12: **fim-para**;
- 13: **para** $i \leftarrow 1$ **até** n^0 mutações **faça**
- 14: **selecione** um cromossomo S_j em P ;
- 15: $S_j \leftarrow$ mutação(S_j)
- 16: **fim-para**
- 17: **até que** critério de parada seja satisfeito;

Algoritmo DE

- 1: função $x = DE(NP, CR, F, range, f)$;
- 2: $x \leftarrow rand(range, NP)$;
- 3: $fit_x \leftarrow f(x)$;
- 4: **enquanto** critério de parada não for satisfeito **faça**;
- 5: **para** $i = 1$ **até** NP **faça**;
- 6: $v_{i,G+1} \leftarrow$ mutação $(x_{i,G}, F)$;
- 7: $u_{i,G+1} \leftarrow$ cruzamento $(x_{i,G}, v_{i,G+1}, CR)$;
- 8: **fim para**
- 9: $fit_u \leftarrow f(u)$;
- 10: **para** $i = 1$ **até** NP **faça**
- 11: **se** $fit_u(i) > fit_x(i)$ **então**;
- 12: $x_{i,G+1} \leftarrow u_{i,G+1}$;
- 13 **se não**
- 14: $x_{i,G+1} \leftarrow x_{i,G}$;
- 15: **fim se**
- 16: **fim para**
- 17: **fim enquanto**

V. RESULTADOS

Nesta seção são apresentados os resultados da implementação do algoritmo de CPA+PSO, CPA+GA e CPA+DE. São apresentados as trajetórias livres de colisão bem como as superfícies potenciais para cada algoritmo.

A. Simulação do CPA+PSO

Na Fig. 3, a seguir, tem-se o gráfico das aptidões e gerações, onde é possível observar que para quantidade de partículas igual a 20 e 30 iterações, tem-se uma convergência entre a melhor partícula, a solução e a média de todas as partículas.

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

A Fig. 4 apresenta as linhas do campo potencial artificial e a trajetória livre de colisão, no espaço cartesiano para a simulação executada, utilizando os parâmetros da Tabela I, obtidos a partir das localizações dos obstáculos.

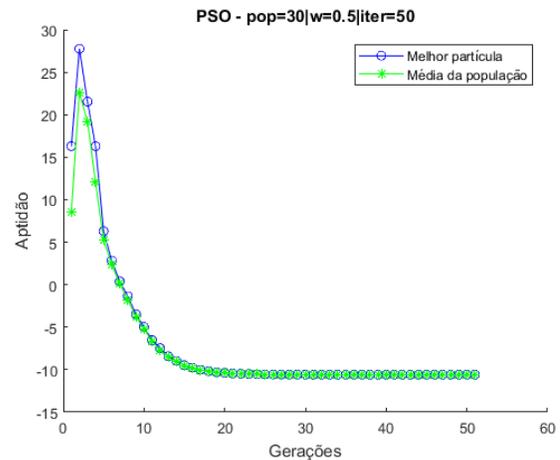


Fig. 3. Gráfico do PSO convergindo para 30 partículas e 50 iterações.

TABELA I
VALORES DOS PARÂMETROS E INTENSIDADES ENCONTRADOS PELO PSO PARA 30 PARTÍCULAS E 50 ITERAÇÕES

Parâmetro	K_r	K_a	K_o
Intensidade	3,92	7,87	1,44

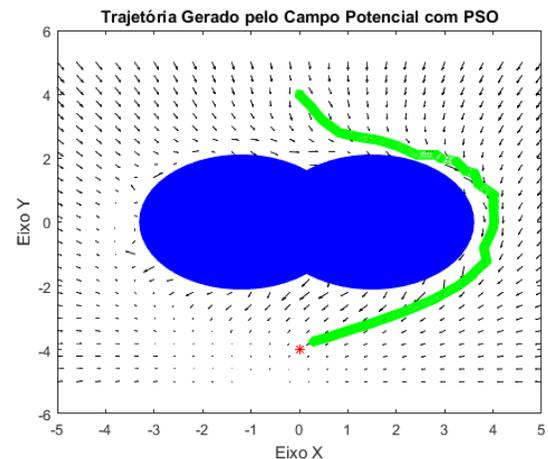


Fig. 4. Linhas do campo potencial e trajetória realizada pelo algoritmo de CPA para os parâmetros da Tabela I.

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

B. Simulação do CPA+GA

Também é apresentado na Fig. 6 o gráfico das aptidões e gerações (50 gerações), observa-se a convergência entre o melhor indivíduo e a média da população.

Na Tabela II, são apresentados os parâmetros do campo potencial encontrados na simulação do algoritmo do GA para 30 indivíduos e 50 gerações.

Também são apresentados na Fig. 7 as linhas do campo potencial artificial e a trajetória livre de colisão, no espaço cartesiano para a simulação executada, utilizando os parâmetros da Tabela II, obtidos a partir das localizações do obstáculo.

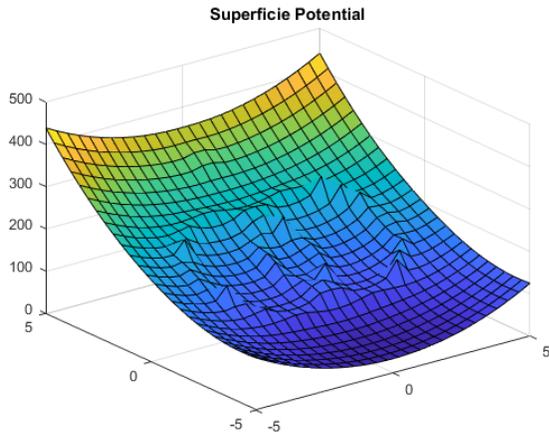


Fig. 5. Superfície resultante do algoritmo de CPA da cena da Fig. 4.

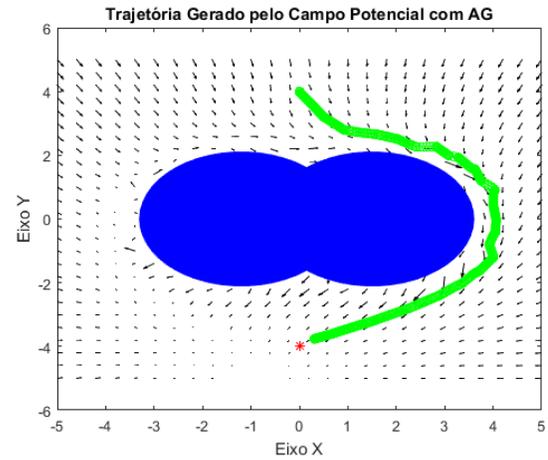


Fig. 7. Linhas do campo potencial e trajetória realizada pelo algoritmo de CPA para os parâmetros da Tabela II.

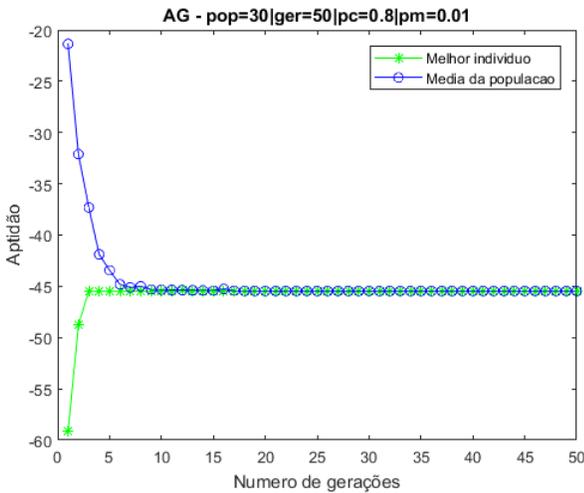


Fig. 6. Gráfico do GA convergindo para 30 indivíduos e 50 gerações.

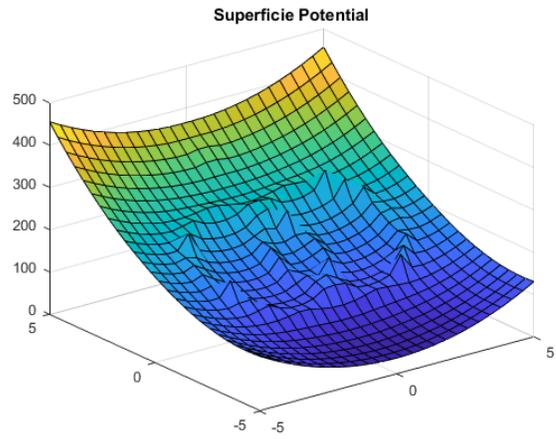


Fig. 8. Superfície resultante do algoritmo de CPA da cena da Fig. 7.

TABELA II
VALORES DOS PARÂMETROS E INTENSIDADES ENCONTRADOS PELO GA

Parâmetro	K_r	K_a	K_o
Intensidade	2.99	8.13	1.62

TABELA III
VALORES DOS PARÂMETROS E INTENSIDADES ENCONTRADOS PELO DE

Parâmetro	K_r	K_a	K_o
Intensidade	2,83	5,18	1,41

Na Fig. 8, a seguir também é apresentada a superfície resultante do campo potencial para a cena mostrada na Fig. 7.

C. Simulação do CPA+DE

Também é apresentado na Fig. 9 o gráfico das aptidões e gerações para quantidade de indivíduos igual a 30 e 50 gerações, também é apresentada o melhor indivíduo e a média da população.

Na Tabela III, são apresentados os parâmetros do campo potencial encontrados na simulação do algoritmo do DE para ara 1000 indivíduos e 30 gerações.

Também são apresentados na Fig. 10 as linhas do campo potencial artificial e a trajetória livre de colisão, no espaço cartesiano para a simulação executada, utilizando os parâ-

metros da Tabela III, obtidos a partir das localizações do obstáculo.

Na Fig. 11, a seguir também é apresentada a superfície resultante do campo potencial para a cena mostrada na Fig. 10.

D. Custo Computacional dos Algoritmos

Foi realizada uma comparação do custo computacional de cada algoritmo utilizado para encontrar os parâmetros do CPA. A Tabela IV a seguir, apresenta o tempo gasto de cada algoritmo para encontrar os parâmetros. Cada algoritmo utilizou o mesmo tamanho da população e mesmo número de gerações, por isso foi possível realizar essa comparação.

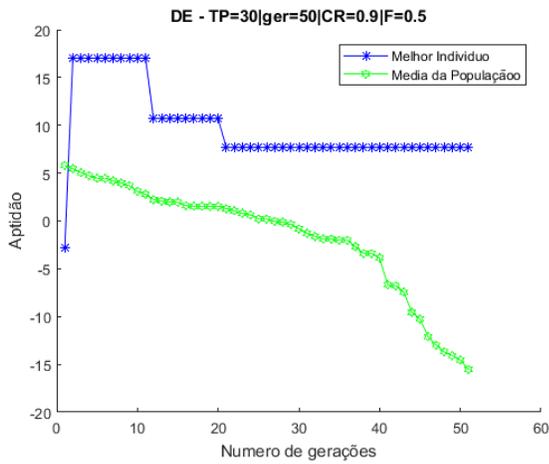


Fig. 9. Gráfico do DE para 30 indivíduos e 50 gerações.

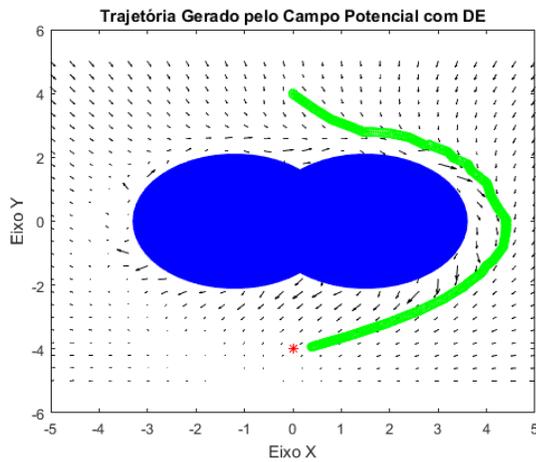


Fig. 10. Linhas do campo potencial e trajetória realizada pelo algoritmo de CPA para os parâmetros da Tabela III.

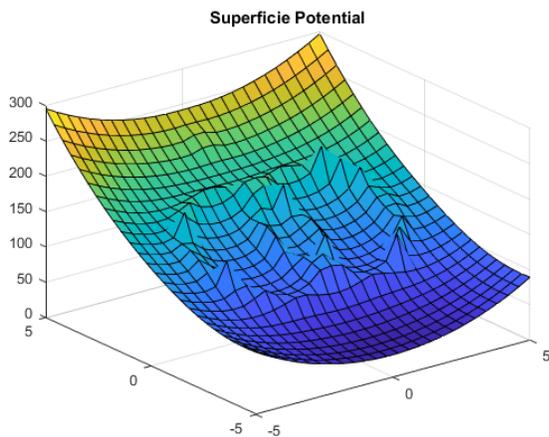


Fig. 11. Superfície resultante do algoritmo de CPA da cena da Fig. 10.

E. Cálculo do Erro de Posição

Calculamos aqui o erro médio entre a posição final desejada e a última posição da trajetória gerada pelo algoritmo de

TABELA IV
CUSTO COMPUTACIONAL DOS ALGORITMOS PSO, GA E DE

Algoritmo	Tempo (s)
PSO	2,289623
GA	1.069343
DE	0.519601

CPA melhorado com PSO, GA e DE. Os valores de erro são apresentados na Tabela V, para os resultados utilizando os parâmetros do CPA com PSO, GA e DE apresentados nas tabelas I, II e III.

TABELA V
VALORES DO ERRO MÉDIO ENTRE A POSIÇÃO FINAL DESEJADA E A ÚLTIMA POSIÇÃO DA TRAJETÓRIA GERADA PELOS ALGORITMOS DE CPA MELHORADO COM PSO, GA E DE

Algoritmo	x (cm)	y (cm)
PSO	0,2970	0,1485
GA	0,3091	0,1546
DE	0,3891	0,1946

F. Discussões

Com a aplicação dos algoritmos de PSO, GA e DE, consegue-se uma melhoria do algoritmo de CPA, pois os parâmetros do CPA são muitas vezes adotados de forma empírica, tornando o processo dispendioso. Com a utilização dos algoritmos de PSO, GA e DE conseguiu-se aqui uma melhoria do algoritmo de campo potencial para gerar trajetórias livres de colisão.

Foram apresentadas as simulações do PSO, GA e DE para uma população de 30 indivíduos (partículas) com 50 iterações (gerações), todas as três situações apresentaram resultados satisfatórios, no entanto a trajetória do CPA com DE apresentou a trajetória mais suave. Também foi possível verificar o erro médio de posição entre a posição final desejada e a última posição da trajetória gerada pelo CPA melhorado com PSO, GA e DE. Na comparação do custo computacional, o algoritmo de DE apresentou um menor tempo de execução para encontrar os parâmetros do algoritmo de CPA.

Como vantagens deste trabalho pode-se citar: a melhoria do algoritmo de CPA, pois os parâmetros são encontrados de forma otimizada; baixo custo computacional dos algoritmos utilizados; precisão dos pontos das trajetórias alcançadas. Como desvantagem pode-se citar que o algoritmo não pode ser utilizado *on-line*, pois os algoritmos são rodados separadamente, isso pode ser contornado criando um único algoritmo híbrido.

VI. CONCLUSÕES

Pode-se concluir que as simulações do método de CPA melhorado com PSO, GA e DE apresentam baixo custo computacional dos algoritmos, permitindo aplicações em ambientes dinâmicos para trajetórias livres de colisão. O método de campo potencial artificial 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 distância mínima. Conclui-se aqui então que em todas as avaliações realizadas, o algoritmo de DE apresentou melhor resultado na geração de uma trajetória suave e obteve um melhor custo computacional, no entanto no erro de posição não foi o melhor. Os algoritmos de PSO e GA também mostraram-se satisfatórios pois as trajetórias do CPA são livres de colisão.

Este trabalho representa uma contribuição pelo fato de aplicar uma técnica consagrada na literatura, Campo Potencial Artificial melhorado com PSO, GA, e DE para geração de trajetória de manipuladores industriais, pois através de pesquisa bibliográfica verifica-se sua maior aplicação na robótica móvel [8]. Os autores estão pesquisando outras formas de implementação melhoradas do algoritmo de CPA, com sistemas mais robustos não lineares como Redes Neurais Artificiais, Mínimos Quadrados, Filtro de Kalman.

REFERÊNCIAS

- [1] IFR. Internacional Federation of Robotics, Global industrial robot sales doubled over the past five years. World Robotics - Industrial Robot Report 2018 published. 2019. Disponível em: <https://ifr.org/ifr-press-releases/news/global-industrial-robot-sales-doubled-over-the-past-five-years>. Acesso em: 24 de abril de 2019.
- [2] PINTO, Milena F. et al. Modified approach using variable charges to solve inherent limitations of potential fields method. In: Industry Applications (INDUSCON), 2014 11th IEEE/IAS International Conference on. IEEE, 2014. p. 1-6.
- [3] KHATIB, Oussama. Real-time obstacle avoidance for manipulators and mobile robots. In: Autonomous robot vehicles. Springer, New York, NY, 1986. p. 396-404.
- [4] L. E. Kavraki, P. Svestka, J. -. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, Aug. 1996
- [5] Batista, J., Silva, J. and Thé, G. Thé. Can Artificial Potentials Suit for Collision Avoidance in Factory Floor? A Case Study of Harmonic Machine-machine Coexistence. In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2018) - Volume 2, pages 547-556.
- [6] GALCERAN, E.; EUSTICE, R. M.; OLSON, E. Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving. In: IEEE. 2015 IEEE Intelligent Vehicles Symposium (IV). [S.l.], 2015. p. 304-309.
- [7] Engelbrecht, A. P. (2007). Computational intelligence: an introduction, John Wiley and Sons.
- [8] Silva, L., Batista, J., The, G. and Reges, J. (2014). Proposal for planning and traffic control for industrial handlers, CBA 2014, CBA, pp. 3954-3961
- [9] SPONG, Mark W. et al. Robot modeling and control. New York: Wiley, 2006.
- [10] Latombe, J.-C. (1990). Robot motion planning (the kluwer international series in engineering and computer science).
- [11] Sciavicco, L. and Siciliano, B. (1996). Modeling and control of robot manipulators, Vol. 8, McGraw-Hill New York.
- [12] CHEN, Fei et al. Evolutionary artificial potential field method based manipulator path planning for safe robotic assembly. In: Micro-NanoMechatronics and Human Science, 2009. MHS 2009. International Symposium on. IEEE, 2009. p. 92-97.
- [13] LaValle, S. M. (2006). Planning algorithms, Cambridge university press.
- [14] Eberhart, R. and Kennedy, J. (1995). A new optimizer using particle swarm theory, Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on, IEEE, pp. 39-43.
- [15] PAIVA, Fabio Augusto Procopio; COSTA, Jose Alfredo Ferreira; SILVA, Claudio Rodrigues Muniz. A Serendipity-Based Approach to Enhance Particle Swarm Optimization Using Scout Particles. IEEE Latin America Transactions, v. 15, n. 6, p. 1101-1112, 2017.
- [16] Meng, Z., Feng, P., Chao, P., Weixing, L. and Qi, G. (2017). Trajectory optimization using time-separating strategy with improved pso on mechanical arms, Control Conference (CCC), 2017 36th Chinese, IEEE, pp. 2669-2674.
- [17] LINDEN, Ricardo. Algoritmos genéticos (2a edição). Brasport, 2008.
- [18] Andries, P. Engelbrecht. Computational Intelligence: An Introduction, 2ª Ed. San Francisco, USA. WILEY. 2007.
- [19] BUDIYANTO, Almira et al. UAV obstacle avoidance using potential field under dynamic environment. In: Control, Electronics, Renewable Energy and Communications (ICCEREC), 2015 International Conference on. IEEE, 2015. p. 187-192.
- [20] LI, Guanghui et al. An improved artificial potential field-based simultaneous forward search (Improved APF-based SIFORS) method for robot path planning. In: Ubiquitous Robots and Ambient Intelligence (URAI), 2015 12th International Conference on. IEEE, 2015. p. 330-335.
- [21] JORGE, Vitor AM et al. Ouroboros: Using potential field in unexplored regions to close loops. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015. p. 2125-2131.
- [22] HARGAS, Youcef et al. Mobile manipulator path planning based on artificial potential field: Application on RobuTER/ULM. In: Electrical Engineering (ICEE), 2015 4th International Conference on. IEEE, 2015. p. 1-6.
- [23] KAZEMI, Hossein et al. Path planning of 3-RRR parallel robot by avoiding mechanical interferences via Artificial Potential Field. In: Robotics and Mechatronics (ICROM), 2015 3rd RSI International Conference on. IEEE, 2015. p. 240-245.
- [24] YU, Koyo et al. Modulated potential field using 5 DoF implant assist robot for position and angle adjustment. In: Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE. IEEE, 2015. p. 002166-002171.
- [25] YU, Koyo et al. Stereo vision based robot navigation system using modulated potential field for implant surgery. In: Industrial Technology (ICIT), 2015 IEEE International Conference on. IEEE, 2015. p. 493-498.
- [26] Sulistijono, I. A., Risnumawan, A., Rachmanita, C. and Kubota, N. (2009). Humanfriendly arm robot using interactive particle swarm optimization for trajectory generation, ICCAS-SICE, 2009, IEEE, pp. 229-234.
- [27] ABDALLA, Turki Y.; ABED, Ali A.; AHMED, Alaa A. Mobile robot navigation using PSO-optimized fuzzy artificial potential field with fuzzy control. Journal of Intelligent and Fuzzy Systems, v. 32, n. 6, p. 3893-3908, 2017.
- [28] SHRIVASTAVA, Maitray; DUTTA, Ashish; SAXENA, Anupam. Trajectory generation using GA for an 8 DOF biped robot with deformation at the sole of the foot. Journal of Intelligent and Robotic Systems, v. 49, n. 1, p. 67-84, 2007.
- [29] MONTIEL, Luis Alberto Hernandez. Hybrid Algorithm Applied on Gene Selection and Classification from Different Diseases. IEEE Latin America Transactions, v. 14, n. 2, p. 930-935, 2016.
- [30] WANG, Mingming et al. Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm. Advances in Space Research, v. 61, n. 6, p. 1525-1536, 2018.
- [31] MAC, Thi Thoa et al. Improved potential field method for unknown obstacle avoidance using UAV in indoor environment. In: Applied Machine Intelligence and Informatics (SAMII), 2016 IEEE 14th International Symposium on. IEEE, 2016. p. 345-350.
- [32] Volpe, Richard, and Pradeep Khosla. "Manipulator control with superquadric artificial potential functions: Theory and experiments." IEEE Transactions on Systems, Man, and Cybernetics 20.6 (1990): 1423-1436.