

Algoritmos genéticos aplicados na cinemática inversa de manipulador robótico

Gabriella Ragazzi Costa (UTFPR - CP) gabriellacosta@alunos.utfpr.edu.br

Márcio Mendonça (UTFPR-CP) mendonca@utfpr.edu.br

Bruno Costalonga Leite (UTFPR-CP) brunoleite@alunos.utfpr.edu.br

Jônatas Favotto Dalmedico jdalmedico@alunos.utfpr.edu.br

Thalita Lopes Salvadori (UTFPR-CP) thalitasalvadori@alunos.utfpr.edu.br

Resumo:

O presente trabalho trata-se sobre robôs manipuladores, estes, por sua vez, têm como principal objetivo realizar o deslocamento, solda entre outras interações em uma linha de montagem ou produção. O tipo mais conhecido de um robô manipulador é um braço mecânico, que consiste em corpos rígidos interligados, com mera semelhança ao braço de um humano, porém nem sempre com a mesma capacidade de movimento. O problema tratado por este artigo refere-se à cinemática inversa de um braço robótico didático resolvida por algoritmos genéticos. O método obteve êxito no cumprimento da proposta, com erros relativamente baixos e, portanto, aceitáveis, visto que os resultados são ainda iniciais porém motivam futuras investigações.

Palavras chave: Algoritmo genético, cinemática inversa, manipulador robótico.

Genetic algorithms applied in the inverse kinematics of robotic manipulator

Abstract

The work deals with robot manipulators, these, in turn, as main objective, displacement, welding among other interactions in an assembly line or production. The best-known type of a robot manipulator is a mechanical arm, consisting of rigid bodies interconnected, with a mere resemblance to a human's arm, but not always with a movement capacity. The problem addressed by this article concerning the inverse kinematics of a didactic robotic arm solved by genetic algorithms. The method was successful in complying with the proposal, with relatively low and therefore acceptable errors, since the results are still initiatives but not motivate future investigations.

Key-words: Genetic algorithm, reverse kinematics, robotic manipulator.

1. Introdução

Robôs são considerados um desenvolvimento relativamente recente. Em 1921, Karl Capeck usou a palavra *checha robota* em um livro e peça chamado R.U.R A palavra foi traduzida para o inglês como *robot*. Uma série de coisas devem ser consideradas ao determinar a resposta à pergunta “O que é de fato um robô?” (MILLER, 2017). Uma definição de robô muito aceita no mundo é do Instituto de Robótica da América (RIA) (SPONG, 1989), “Um robô é um manipulador multifuncional reprogramável projetado para mover materiais, peças, ferramentas, ou dispositivos especiais através de variáveis de movimento programadas para a realização de uma variedade de tarefas”.

Os primeiros robôs manipuladores surgem no ano de 1954, o dispositivo de Transferência Programada de Artigos de George C. Devol e, o dispositivo robótico patenteado pelo britânico Cyrill Walter Fenward. A partir daí muitos projetos se desenvolveram com êxito, como exemplo, o braço manipulador da Universidade de Stanford e o robô PUMA. A Pesquisa Operacional surgiu a partir de tentativas de lidar com problemas de natureza logística, tática e estratégia. Os braços manipuladores na área de sistemas operacionais desempenham um importante papel, uma vez que são programáveis, sendo possível definir trajetórias e outros comandos que podem ser repetidos continuamente. O foco no estudo de braços robóticos se dá para a garantia da segurança do ser humano no quesito economia de tempo e esforço.

Os robôs não têm capacidade para responder em situações de emergência, a menos que a situação seja prevista e a resposta esteja incluída no sistema. Isto divide a robótica em dois campos, a robótica programada como, por exemplo, o robô PUMA citado anteriormente, e robótica autônoma.

A área de robótica autônoma vem atraindo a atenção de um grande número de pesquisadores, devido ao desafio que este novo domínio de pesquisas propõe: dotar sistemas de uma capacidade de raciocínio inteligente e de interação com o meio em que estão inseridos (MEDEIROS 1998, BEKEY 2005). Redes neurais, sistemas fuzzy e algoritmos evolutivos são técnicas computacionais inteligentes que vêm sendo utilizadas mais amplamente na área de robótica autônoma.

Algoritmos Evolutivos têm sido utilizados em Robótica principalmente porque o projeto de robôs autônomos e de seus controladores para ambientes não-estruturados, flexíveis, e/ou parcialmente desconhecidos é uma tarefa muito difícil para um projetista humano (MEYER, 1998). Meyer fazia menção às tarefas que o robô iria executar, como também ao ambiente em que o mesmo iria atuar. Para um projetista humano é praticamente impossível prever todas as situações que o robô estaria exposto, e é aí que entra em ação os algoritmos evolutivos.

Baseados na Teoria da Evolução Natural das Espécies de Darwin, os algoritmos evolutivos, mais conhecidos como algoritmos genéticos, a partir de uma população inicial de soluções denominadas cromossomos que vão sendo evoluídas até uma solução que atenda um critério específico pré-estabelecido. As evoluções são caracterizadas por uma função medidora do nível de aptidão de cada cromossomo, chamada *fitness*. Assim, os cromossomos de melhor *fitness* sofrem operações de mutações ou cruzamentos com outros cromossomos e, são selecionados para compor a próxima geração. A partir disso, a cada geração o conjunto de soluções é melhorado até que se chegue a uma solução ótima.

Este artigo aborda a respeito da utilização de algoritmos genéticos para solucionar um problema muito comum dos manipuladores robóticos, o problema da cinemática inversa. Observa-se que este artigo é inspirado em trabalhos que empregam algoritmos evolutivos na área de robótica como, por exemplo, os trabalhos de Adriane B. S. Serapião e Raphael P. Azzolini, Algoritmos Genéticos para navegação autônoma com múltiplos objetivos em robótica coletiva e, Navegação autônoma de robôs em ambientes aleatórios através de estratégias evolutivas e, a tese de mestrado de Eduardo J. S. Pires – Algoritmos Genéticos: aplicação à robótica, pela similaridade que os mesmos possuem com o problema abordado.

1.1 Problemática

O robô manipulador estudado neste artigo também pode ser dito como robô braço articulado pois possui uma configuração similar ao braço humano. O manipulador consiste em componentes lineares correspondentes ao braço e antebraço humano, uma pinça que pode ser correlacionada a uma mão, montados num pedestal e conectados por juntas rotacionais

correspondentes ao ombro, cotovelo e pulso. Este é caracterizado como 4 graus de liberdade GDL com elos rotacionais: robô do tipo manipulador quatro graus-de-liberdade de articulações rotacionais.

Dependendo de sua configuração e do tamanho de suas ligações e articulações, os robôs podem chegar a uma coleção de pontos ao redor deles, que constituem um espaço de trabalho. O espaço de trabalho pode ser definido matematicamente incluindo as limitações do robô, como os alcances dos movimentos de cada articulação. Por simplicidade e estratégia para planejamento de trajetória somente as três juntas planares foram modeladas, para a posteriori, existirem dois tipos distintos de ações de controle, as ações de controle que irão posicionar o modelo explorado nessa pesquisa com 3GDL responsáveis pela movimentação das juntas destacadas em vermelho em um ponto específico no espaço dentro da área de alcance para finalmente uma ação de controle deslocar o plano até a posição desejada no espaço 3D, e uma rotação no elo da base detalhada na figura2. A Figura 4 mostra a nuvem de pontos planares do espaço de trabalho modelado inicialmente nesta pesquisa.

O cálculo da posição e da orientação da mão do robô é chamado de análise cinemática direta. Dada a representação de um manipulador de quatro articulações no plano xy (Figura 1), pelas equações da cinemática direta tem-se a equação (1), (2). Observa-se que o modelo empregado nesse trabalho foi geométrico, devido a modelagem planar, entretanto o modelo a ser utilizado na sequência da pesquisa (modelo completo do braço manipulador em 3D) deverá ser o modelo de Denavit-Hartenberg D-H (CRAIG, 2012).

$$\begin{cases} x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \sin(\theta_1 + \theta_2 + \theta_3) \end{cases} \quad (1),(2)$$

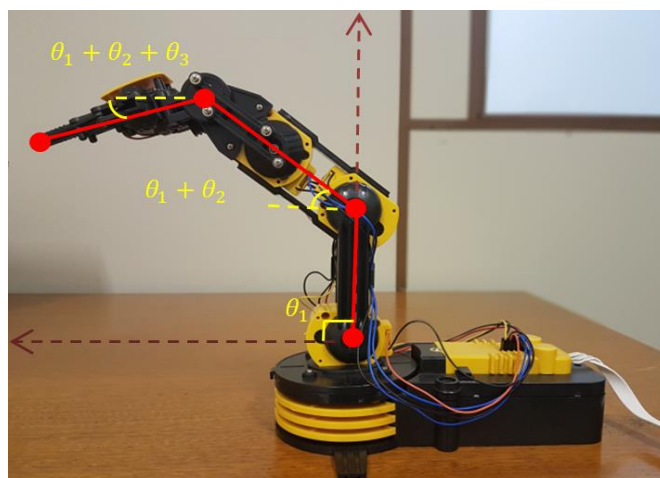


Figura 1 – Exemplo de manipulador 4GDL RR no plano xy.
Fonte: Autoria própria.

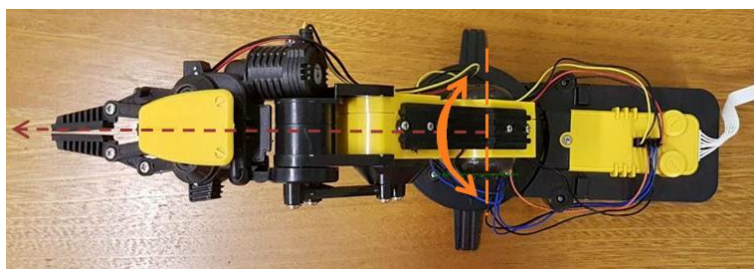


Figura 2 – Exemplo de manipulador 4GDL RR no plano xy (vista superior).
Fonte: Autoria própria.

No entanto, se quisermos colocar a mão do robô em um local e orientação desejados, precisamos saber quais devem ser os ângulos das articulações do robô. Isso é denominado análise cinemática inversa. De acordo com Guez e Ahmad (1988), uma das problemáticas dos braços robóticos, a qual é tratada neste artigo, é a chamada cinemática inversa. Trata-se da maneira de encontrar os parâmetros das juntas para uma determinada posição e orientação do efetuador, ou seja, como calcular os ângulos necessários para que o manipulador atinja a posição final desejada.

A cinemática inversa é a base para a movimentação de robôs manipuladores, pois todas as tarefas realizadas necessitam do cálculo da cinemática inversa para obtenção dos ângulos referentes à posição final pré-determinada. Portanto, o cálculo da cinemática inversa é importante dentro do estudo da cinemática de robôs manipuladores.

Diferente da cinemática direta, para cinemática inversa não existe um procedimento sistemático explícito.

É possível posicionar um ponto no espaço com base em coordenadas cartesianas, cilíndricas, esféricas ou articulares. *A priori*, é necessário o planejamento da trajetória. Esta refere-se à forma como o robô é movido de um ponto para outro de maneira controlada, seguindo a coleção de sequências de configurações de percurso. Na especificação da trajetória devem ser levadas em consideração três tipos de restrições:

- a) Espaciais, ou seja, o ponto inicial e o ponto final do caminho. A presença de obstáculos pode requerer a especificação de eventuais pontos intermediários;
- b) Temporais, ou duração de cada trecho da trajetória;
- c) E mecânicas, melhor dizendo, a trajetória gerada deve corresponder a pontos dentro do espaço de trabalho do manipulador.

A problemática da cinemática inversa pode ser solucionada por intermédio de, além de outras conhecidas, soluções técnicas computacionais inteligentes como, por exemplo, a proposta por este artigo, Algoritmos Genéticos. Inerente a cinemática inversa temos o problema de múltiplas soluções, ou seja, um conjunto de soluções de ângulos diferentes que levam a ferramenta final do braço manipulador as mesmas posições iniciais finais e intermediárias de uma determinada trajetória. Além disso, as possíveis soluções aumentam exponencialmente com o aumento dos graus de liberdade. Esse trabalho restringe-se a solução do problema da cinemática inversa, entretanto a proposta desse trabalho é de se utilizar algoritmos genéticos para definir a aptidão de uma solução com a proximidade da solução imediatamente anterior buscando uma possível solução para múltiplas soluções e de trajetórias.

2. Metodologia

Inicialmente o braço mecânico com três graus de liberdade é modelado, no qual o tamanho das articulações 9cm da base, com de 45° até a primeira articulação, 11 cm, com 45° na articulação do meio e 11 cm, com 60° na parte da pinça.

O manipulador da Figura 3 pode ser representado algebricamente por meio de equações da cinemática direta, correspondente aos 3 graus de liberdade a fim de determinar possíveis posições no plano (x,y) para a pinça. Dessa forma, dado os ângulos de cada abertura das articulações, e seus tamanhos, pode-se evidenciar por meio de uma nuvem de pontos, quais posições o manipulador pode adquirir. Abaixo a representação da nuvem de pontos é observada na Figura 4 restringindo-se os ângulos θ_1 de 0 a 90 graus, θ_2 0 a 90 e θ_3 0 a 90. O anexo 1 endereça maiores informações para a geração da nuvem de pontos por meio do script do MatLab.

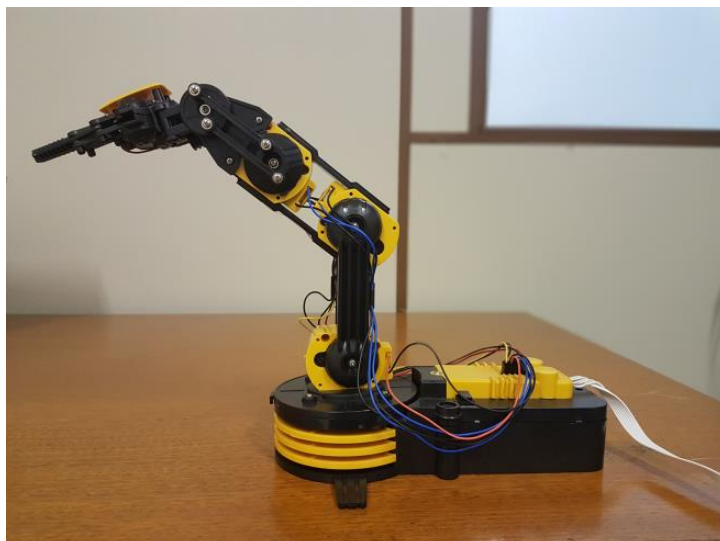


Figura 3 – Braço mecânico

Fonte: Autoria própria

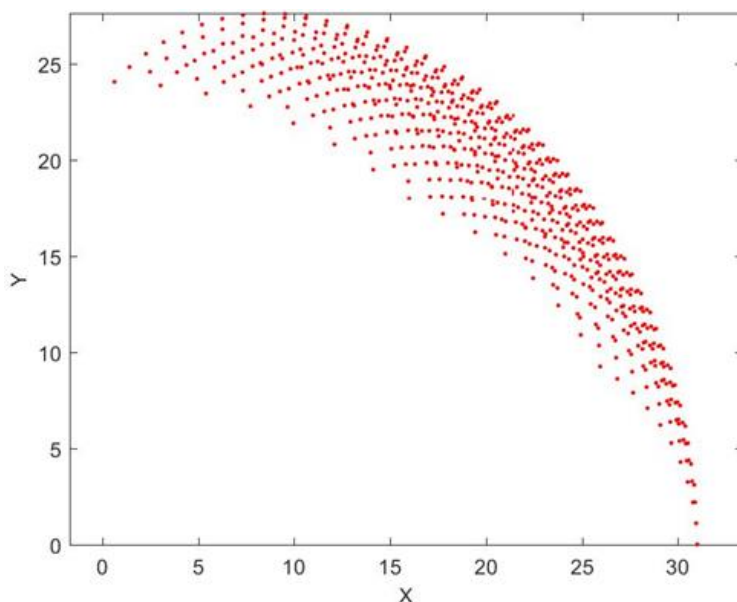


Figura 4 – Nuvem de Pontos

Fonte: Autoria Própria

3. Algoritmos Genéticos

O desenvolvimento de simulações computacionais de sistemas genéticos teve início nos anos 50 e 60 através de muitos biólogos, mas foi John Holland que começou a desenvolver as primeiras pesquisas no tema. Em 1975, Holland publicou "Adaptation in Natural and Artificial Systems", ponto inicial dos Algoritmos Genéticos (AGs). Segundo Kramer (2017), Algoritmos genéticos são abordagens de pesquisa heurística aplicáveis a uma ampla gama de problemas de otimização, o que os torna atraente para diversos problemas. Nesse contexto, de um modo geral os algoritmos genéticos possuem como objetivo solucionar problemas do mundo real de forma otimizada, uma vez que trabalham com um conjunto de parâmetros e não com os parâmetros individualmente, utilizando regras de transição probabilística e não determinística, que podem ser aplicadas em diversas áreas. Trabalham com uma população de soluções iniciais, denominados cromossomos que vão sendo evoluídos até que se chegue

em uma solução ótima. Porém, para que isto ocorra é preciso que cada geração de cromossomos seja analisada, para se ter conhecimento do nível de aptidão, chamado de função *fitness*.

Os cromossomos com o melhor *fitness* são selecionados para darem origem a próxima geração, através de cruzamento e mutação. O que se espera é que a cada geração o conjunto de soluções seja melhorado, até que se chegue a uma solução que satisfaça as características desejadas.

A *priori* é selecionada uma população inicial constituída por indivíduos aleatórios (se for grande perde eficiência e se for pequena perde a diversidade), *posteriori* é definida a função *fitness* e assim são cruzados e sofrem mutação. Ocorre também o *crossing over* (troca de material genético). Essas etapas ocorrem até que a melhor solução seja encontrada.

Por conseguinte, o objetivo deste trabalho institui-se pela utilização de um Algoritmo Genético – GA para a obtenção da cinemática inversa, dado um braço manipulador de 4 GDL RR ilustrado na Figura 2, com modelo cinemático apenas 3 GDL no plano, como previamente abordado. Determinando, portanto, os números de iterações utilizados 200, 50, 20 com a finalidade de se comparar os resultados obtidos e o erro ser analisado. As tabelas 1, 2, e 3 abaixo contém soluções encontradas pelo GA juntamente com os dados desejados para vários pontos pertencentes a nuvem de pontos, ou seja, as coordenadas (x,y) e os ângulos correspondentes para a sua trajetória. O erro apresentado foi a distância euclidiana do ponto desejado e da solução encontrada pelo A.G.

A distância euclidiana entre os pontos em um espaço n-dimensional pode ser definida através da seguinte equação (3):

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sum_{i=1}^n (p_i - q_i) \quad (3)$$

Theta ₁	Theta ₂	Theta ₃	X	Y	X _{final}	Y _{final}	Erro
39.642	36.307	57.573	2	24	2.025	x24.388	0.389
26.731	43.330	57.679	5	23	4.488	23.204	0.550
16.258	21.311	62.000	15	20	15.529	20.073	0.534
7.543	9.725	67.012	20	15	20.522	15.392	0.653
5.606	3.886	49.848	25	12	25.415	12.155	0.443
0.873	9.905	2.129	30	5	30.154	4.545	0.479

Fonte: Autoria própria

Tabela 1 - Dados correspondentes a utilização do GA para coordenadas iniciais (número de iterações: 200)

Theta ₁	Theta ₂	Theta ₃	X	Y	X _{final}	Y _{final}	Erro
39.929	35.371	58.443	2	24	2.087	24.363	0.374
28.027	41.765	57.474	5	23	5.083	23.306	0.317
13.559	28.461	57.018	15	20	15.193	20.337	0.388
5.008	16.698	59.173	20	15	20.929	15.715	1.172
6.690	0.735	54.869	25	12	24.960	12.209	0.212
2.541	2.442	14.926	30	5	30.292	5.100	0.308

Fonte: Autoria própria

Tabela 2: Dados correspondentes a utilização do GA para coordenadas iniciais (número de iterações: 50)

Theta ₁	Theta ₂	Theta ₃	X	Y	X _{final}	Y _{final}	Erro
36.659	42.936	50.800	2	24	2.077	24.570	0.575
41.348	24.564	55.298	5	23	5.545	25.395	2.457
14.028	30.249	51.199	15	20	15.556	20.811	0.983
14.541	2.609	59.328	20	15	21.794	16.198	2.157
2.075	4.548	58.300	25	12	24.582	11.557	0.607
4.444	0.084	13.246	30	5	30.413	4.924	0.420

Fonte: Autoria própria

Tabela 3: Dados correspondentes a utilização do GA para coordenadas iniciais (número de iterações: 20)

As simulações realizadas nos fornecem os ângulos necessários para se atingir os pontos estipulados inicialmente (X e Y) dessa forma o GA trabalhando com a cinemática inversa pode determinar as informações angulares dentro da faixa pré-estabelecida pelo manipulador e fornece a solução ótima encontrada pelo algoritmo genético, que são X_{final} e Y_{final} . Com o valor estipulado inicialmente e o valor encontrado, pode-se calcular o erro. Com isso nota-se que quanto maior o número de iterações mais precisa será a solução encontrada, uma vez que o erro é diminuído e a função *fitness* é melhorada. O intervalo de tempo para as simulações também deve ser considerado, uma vez que quanto maior o número de iterações o tempo para obter os resultados é maior.

4. Resultados Experimentais Simulados

Os resultados ainda são iniciais e, a partir das Tabelas apresentadas no tópico anterior, gerou-se os gráficos apresentados nas Figuras 5, 6, e 7.

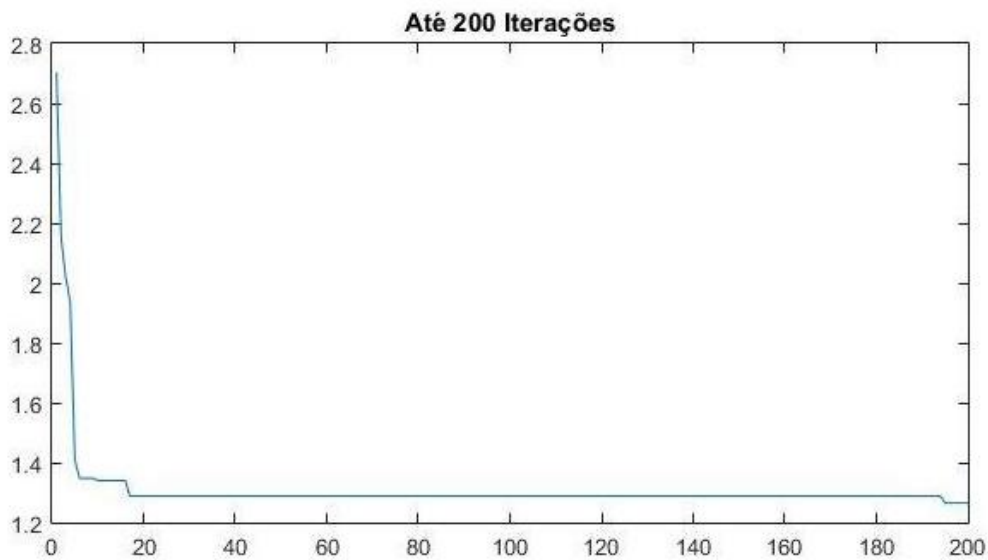


Figura 5 – Gráfico referente à Tabela 1

Fonte: Autoria Própria

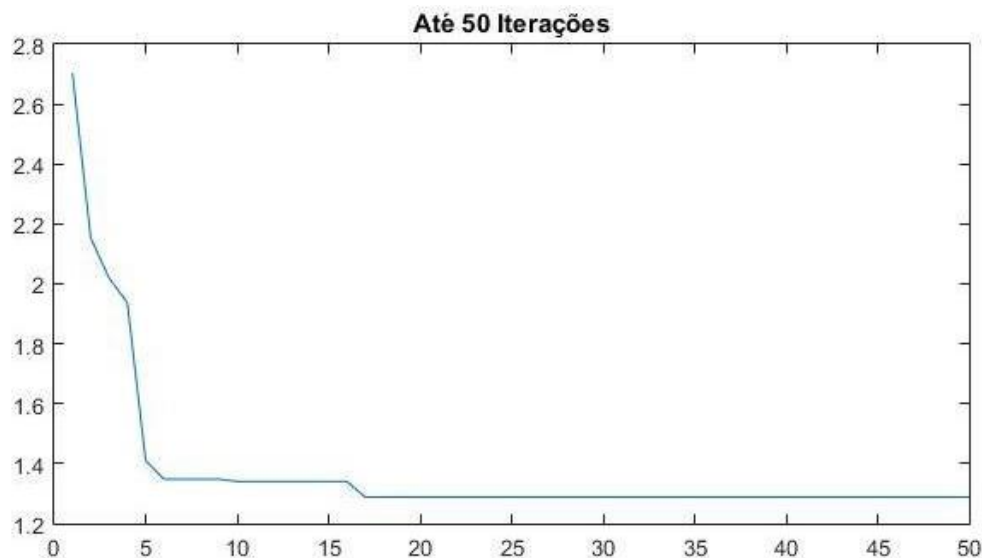


Figura 6 – Gráfico referente à Tabela 2

Fonte: Autoria Própria

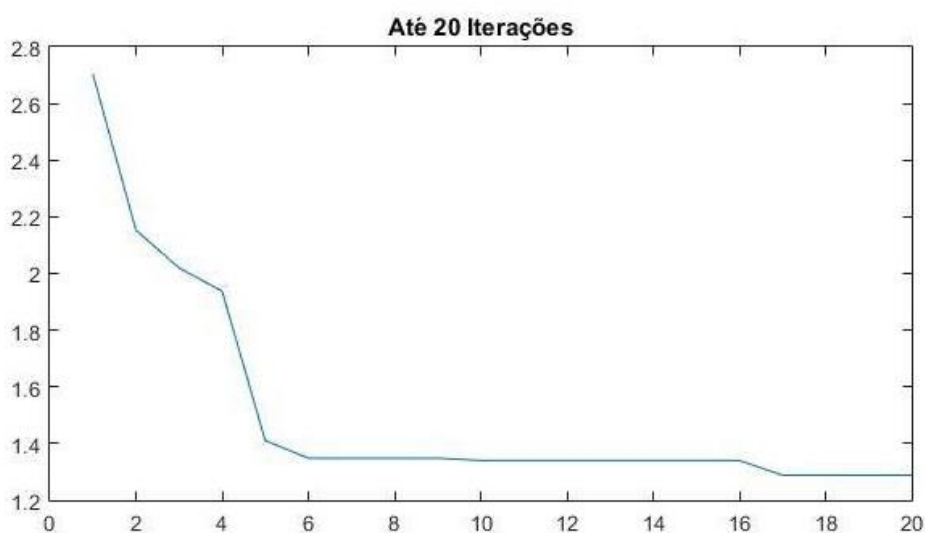


Figura 7 – Gráfico referente à Tabela 3

Fonte: Autoria Própria

Para X=15 e Y=25 Desejado							Erro para o número de iterações de		
Teta ₁	Teta ₂	Teta ₃	X	Y	X _{final}	Y _{final}	200	50	20
25.449	41.301	10.436	15	25	14.908	24.700	0.281	0.283	0.286

Fonte: Autoria própria

Tabela 5: Dados sobre ponto fixo (x,y)

Por meio das análises dos gráficos e tabelas acima, pode-se evidenciar que o uso do Algoritmos Genéticos sugere ser satisfatório, entretanto deve-se buscar uma melhor

parametrização para diminuir o erro encontrado. A maior vantagem poderá ser na investigação da geração de trajetórias e simultaneamente na atenuação do problema de múltiplas soluções. E, finalmente, observou-se que em vários experimentos simulados 20 gerações foi suficiente para o A.G. encontrar uma solução plausível para o problema proposto.

5. Considerações finais

Os resultados simulados ainda são iniciais, com necessidade de diminuição dos erros encontrados, porém a solução por meio de Algoritmos Genéticos pode ser uma possível solução do problema abordado simultaneamente com a geração de trajetórias com pequenos incrementos nos ângulos inicialmente encontrados pela ferramenta. E, conseqüentemente atenuação do clássico problema de múltiplas soluções. Futuros trabalhos endereçam evoluir o modelo apresentado para o D-H com os 4 graus de liberdade. Após essa etapa, objetiva-se a conexão e comunicação do ambiente de desenvolvimento matlab e o protótipo didático modelado para investigação e tratamento do problema em uma situação real.

6. Agradecimentos

Os autores a Universidade Tecnológica Federal do Paraná Campus Cornélio Procopio pelo apoio e incentivo a produção desse trabalho.

Referências

- ALSINA, P. J.** *Cap. 5 Controle de Manipuladores*. Apostila, Cap. 5, p. 83-124, Centro de Tecnologia, Universidade do Rio Grande do Norte.
- BEKEY, G. A.** *Autonomous Robots: From Biological Inspiration to Implementation and Control*. MIT Press, Cambridge, MA, 2005.
- CRAIG, J. J.** *Robótica*. 3. ed. São Paulo: Pearson, 2012.
- GROOVER, M. P.** *Robótica: Tecnologia e Programação*. McGraw-Hill, 1998.
- HOLLAND, J. H.** *Adaptation in natural and artificial systems*. The University of Michigan Press, Ann Arbor, MI, 1975.
- KRAMER, O.** *Genetic Algorithm Essentials*. USA: Springer, 2017.
- MEDEIROS, A.** *Introdução à robótica*. In Anais do XVII Encontro Nacional de Automática, Vol. 1, p. 56-65, Natal, RN, Brasil, 1998.
- MEYER, J. A.** *Evolutionary approaches to neural control in mobile robots*. Proc. of the IEEE Conf. on Syst., Man, and Cyb, 1998.
- MILLER, R. M.; MILLER, R.** *Robots and Robotics: Principles, Systems, and Industrial Applications*. USA: Mc Graw Hill Education, 2017.
- NIKU, S. B.** *Introdução à robótica: análise, controle, aplicações*. 2. ed. Rio de Janeiro, RJ: LTC, 2013.
- PIRES, E. J. S.** *Algoritmos Genéticos: Aplicação à Robótica*. 1998. 236 p. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) – Faculdade de Ciências e Tecnologia, Universidade de Coimbra, 1998.
- SPONG, M. W; VIDYASAGAR, M.** *Robot Dynamics and Control*. New York: John Wiley & Sons, 1989.
- SERAPIÃO, A. B. S.; AZZOLINI, R. P.** *Algoritmos genéticos para navegação autônoma com múltiplos objetivos em robótica coletiva*. Anais do XIX Congresso Brasileiro de Automática, CBA, artigo aceito em 2012 para publicação.
- SERAPIÃO, A. B. S.; AZZOLINI, R. P.** *Navegação autônoma de robôs em ambientes aleatórios através de estratégias evolutivas*. X SBAI – Simpósio Brasileiro de Automação Inteligente, artigo aceito em 2011 para publicação.

ANEXO

Código (Nuvem de pontos) referente à Figura 3

l1 = 9; % length of first arm

l2 = 11; % length of second arm

l3 = 11; % length of thirt arm

%area determinate for arm manipulator

%within the defined range

theta1 = 0:0.1:pi/4; % all possible theta1 values

theta2 = 0:0.1:pi/4; % all possible theta2 values

theta3 = 0:0.1:pi/3; % all possible theta3 values

[THETA1,THETA2,THETA3] = meshgrid(theta1,theta2,theta3); % generate a grid of theta1, theta2 and theta3 values

X = l1 * cos(THETA1) + l2 * cos(THETA1 + THETA2) + l3 * cos(THETA1 + THETA2 + THETA3); % compute x coordinates

Y = l1 * sin(THETA1) + l2 * sin(THETA1 + THETA2) + l3 * sin(THETA1 + THETA2 + THETA3);
% compute y coordinates

data1 = [X(:) Y(:) THETA1(:)]; % create x-y-theta1 dataset

data2 = [X(:) Y(:) THETA2(:)]; % create x-y-theta2 dataset

data3 = [X(:) Y(:) THETA3(:)]; % create x-y-theta3 dataset