

ENSINO DE ROBÓTICA ATRAVÉS DE ALGORITMOS GENÉTICOS

José Tarcísio Franco de Camargo – jffc@bol.com.br

Eliana Anunciato Franco de Camargo – eafcamargo@yahoo.com

Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL
Av. Hélio Vergueiro Leite, s/n – Jardim Universitário
CEP 13990-000 – Espírito Santo do Pinhal – SP

Gilmar Barreto – gbarreto@dsif.fee.unicamp.br

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação, UNICAMP, FEEC
Av. Albert Einstein, 400, Cidade Universitária Zeferino Vaz, Barão Geraldo
CEP 13083-852 – Campinas – SP

Jorge Cândido – jocandido@utfpr.edu.br

Universidade Tecnológica Federal do Paraná
Campus de Campo Mourão – UTFPR-CM
Departamento de Gestão e Economia – DAGEE
Vila Rosalina Maria dos Santos, 1.23
CEP – 87.301-899 – Campo Mourão – PR

Estéfano Vizconde Veraszto – estefanovv@cca.ufscar.br

Universidade Federal de São Carlos
Depto. de Ciências da Natureza, Matemática e Educação, UFSCar, CCA
Rodovia Anhanguera, Km 174
CEP 13604-900 – Araras - SP

Resumo: *O estudo de robôs articulados passa, necessariamente, pelo desenvolvimento dos seus modelos cinemáticos. Por sua vez, a cinemática de um robô pode ser descrita através de seus modelos direto e inverso. O modelo cinemático inverso, através do qual obtém-se o estado das juntas em função da posição desejada para a extremidade livre do robô, normalmente é descrito de forma algébrica. Contudo, esta representação é, frequentemente, de difícil obtenção. Assim, embora seja inquestionável a necessidade da determinação exata do modelo cinemático inverso, o uso de algoritmos genéticos na fase de projeto pode ser muito atraente, pois permite prever o comportamento do robô antes do desenvolvimento formal de seu modelo. Neste sentido, os resultados deste trabalho apresentam uma forma relativamente rápida para se simular o modelo cinemático inverso, a qual se mostrou útil no ensino de robótica em um curso de engenharia, permitindo que o estudante pudesse ter uma visão mais ampla do modelo, vindo a identificar pontos que poderiam ser otimizados na estrutura de um robô. Pode-se concluir que o uso dos algoritmos genéticos no ensino de robótica é viável, tendo como principais vantagens a sua fácil implementação computacional e sua precisão na representação de modelos cinemáticos.*

Palavras-chave: Algoritmos Evolutivos. Otimização de Processos. Simulação Computacional.

1 INTRODUÇÃO

O movimento descrito por um robô manipulador pode ser representado através de seus modelos cinemáticos direto e inverso, conforme descrito em Craig (2017). A obtenção do modelo cinemático direto é relativamente simples, sendo este definido através de um conjunto de transformações entre os sistemas de referência de suas juntas ou “graus de liberdade”. Através deste modelo podemos determinar a posição da “ferramenta” na extremidade livre do robô sendo conhecidas as posições de suas juntas.

O modelo cinemático inverso, por sua vez, permite-nos determinar o estado das juntas de um robô em função da posição desejada para a sua ferramenta. Dessa forma, tendo sido definida uma trajetória para a ferramenta, é possível determinar o conjunto de posições das juntas que permitirão ao robô descrever o deslocamento desejado (MILLER, 2017).

A obtenção do modelo cinemático inverso, contudo, tende a ser mais complexa do que a obtenção do modelo cinemático direto, uma vez que envolve a solução de um sistema de equações não lineares que pode admitir mais de uma solução. Mesmo em casos relativamente simples, como para o “robô planar” com dois graus de liberdade, descrito a seguir e utilizado como modelo didático no ensino de robótica, a definição da cinemática inversa não é trivial.

Dessa forma, ser capaz de prever o comportamento de um robô, de forma relativamente simples, antes do desenvolvimento formal de seu modelo cinemático inverso, pode-se tornar um fator relevante para o aprendizado de robótica em um curso de engenharia. Através do uso de algoritmos genéticos (AGs) é possível simular o comportamento de um robô, determinando com relativa precisão o estado de suas juntas em função da posição desejada para seu extremo livre, permitindo que falhas de projeto possam ser detectadas, bem como pontos passíveis de otimização possam ser identificados.

Assim, este trabalho visa apresentar o tema dos AGs no contexto da simulação em robótica, procurando apresentar uma solução genérica capaz de representar o comportamento de modelos cinemáticos inversos de robôs articulados de maneira prática, eficiente e relativamente simples, com vistas à exploração de oportunidades de otimização em projetos robóticos, no decorrer de um curso de engenharia.

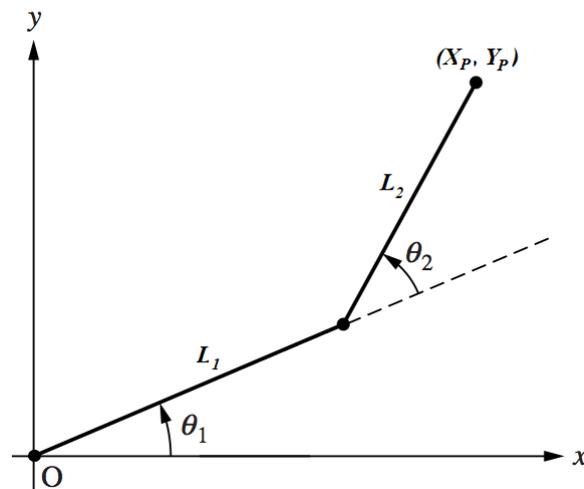
2 CINEMÁTICA DIRETA E INVERSA DO MOVIMENTO DE UM ROBÔ

O estudo dos modelos cinemáticos direto e inverso em robótica pode ser introduzido, de forma simples, através de um robô manipulador planar, conforme apresentado na Figura 1.

Na estrutura da Figura 1 temos um robô com dois graus de liberdade, estando sua extremidade inferior presa ao sistema de coordenadas referencial e seu extremo superior (ferramenta) livre para se deslocar sobre o plano cartesiano. Este robô é composto por dois braços de comprimento L_1 e L_2 , sendo que a cada braço temos associado uma junta rotacional, cujos ângulos de rotação são θ_1 e θ_2 , respectivamente. A extremidade livre deste robô encontra-se na posição (X_P, Y_P) do plano.

O modelo cinemático direto deste manipulador, que indica a posição da extremidade livre do robô em função dos ângulos das juntas rotacionais, pode ser representado, de forma bastante simples, através das equações (1) e (2).

Figura 1 – Estrutura de um robô planar com dois graus de liberdade.



Fonte: Elaborado pelos autores.

$$X_p = L_1 \cdot \cos \theta_1 + L_2 \cdot \cos(\theta_1 + \theta_2) \quad (1)$$

$$Y_p = L_1 \cdot \sin \theta_1 + L_2 \cdot \sin(\theta_1 + \theta_2) \quad (2)$$

Por sua vez, o modelo cinemático inverso, que indica os ângulos θ_1 e θ_2 em função da posição desejada para a ferramenta, pode ser descrito, segundo Craig (2017), através das equações (3) e (4).

$$\theta_2 = \pm \cos^{-1} \frac{X_p^2 + Y_p^2 - L_1^2 - L_2^2}{2 \cdot L_1 \cdot L_2} \quad (3)$$

$$\theta_1 = \tan^{-1} \frac{Y_p \cdot (L_1 + L_2 \cdot \cos \theta_2) - X_p \cdot L_2 \cdot \sin \theta_2}{X_p \cdot (L_1 + L_2 \cdot \cos \theta_2) + Y_p \cdot L_2 \cdot \sin \theta_2} \quad (4)$$

Neste modelo, o ângulo θ_2 pode assumir um valor positivo ou negativo, dependendo da posição do “cotovelo” do robô (“para cima” ou “para baixo”).

Uma forma de simular a cinemática inversa de um robô sem a definição explícita deste modelo consiste no uso de AGs. Neste caso, um AG pode ser construído de tal forma que, dada uma estimativa inicial para os valores de θ_1 e θ_2 e uma função objetivo (chegar o mais próximo possível das coordenadas X_p e Y_p), esta estimativa possa ser refinada em direção a uma solução ótima. Por meio desta estratégia, o AG pode vir a ser inicialmente alimentado por uma estimativa aleatória para os valores de θ_1 e θ_2 , evoluindo esta solução até uma determinada condição de mínimo erro de posicionamento. Por exemplo, a distância relativa entre (X_p, Y_p) desejado para a extremidade do robô e (X_p', Y_p') determinado a partir da otimização das estimativas para θ_1 e θ_2 .

Dessa forma, através do uso de AGs, é possível estudar a cinemática inversa de um robô sem a especificação formal de seu modelo, permitindo, portanto, que determinados comportamentos possam vir a ser identificados pelo algoritmo genético. Assim, o estudo em robótica aqui proposto encontra-se direcionado ao uso de AGs, como alternativa aos métodos convencionais para a determinação do modelo cinemático inverso de um robô manipulador.

3 FUNDAMENTOS DOS ALGORITMOS GENÉTICOS

O uso de algoritmos genéticos em problemas de otimização foi proposto inicialmente em Holland (1975), sendo popularizado através de Goldberg (1989) e Haupt (2004). De forma sucinta, pode-se afirmar que AGs constituem uma analogia à Teoria da Evolução das Espécies de Charles Darwin (DARWIN, 2004), a qual, por sua vez teve início com a integração de conceitos entre seleção natural e genética realizados por Gregor Mendel (MILLER, 2009). Em resumo, em um ambiente computacional, busca-se a evolução de uma determinada solução para um problema, desde uma estimativa inicial, até uma solução ótima ou próxima desta. Para tanto, este processo de otimização exige um “espaço de busca”, formado por “indivíduos” de uma “população”, onde se procura a solução ótima para o caso estudado, além de uma “função objetivo”, a qual encaminha a busca em direção à melhor solução possível (BING, 2016; GUPTA, 2016; KRAMER, 2017).

Dessa forma, o uso de AGs implica em um processo estocástico, onde possíveis soluções são agrupadas formando uma população, sendo todos os indivíduos avaliados simultaneamente, com notas mais altas atribuídas aos mais “aptos”, ou seja, às melhores soluções.

Por sua vez, a evolução da população em direção a soluções otimizadas passa por eventos onde os indivíduos combinam-se entre si, em processos de “cruzamento” (*crossover*), ou sofrem “mutações”, de forma análoga ao que ocorre em populações biológicas. Tal evolução dá origem a novas gerações que devem representar melhores soluções para o problema abordado.

3.1 O algoritmo genético binário

A implementação computacional de um AG é relativamente simples, sendo importante codificar os indivíduos de uma população em uma representação binária, para a devida aplicação do algoritmo. Assim, a partir da população inicial de indivíduos que constitui um espaço de busca em direção à melhor solução do problema, enquanto um determinado critério evolutivo não for alcançado, devem ser repetidos os seguintes passos.

1. Avalia-se cada um dos indivíduos da população, atribuindo aos mesmos notas que representam suas respectivas “aptidões” para a solução do problema tratado. Tais notas são construídas a partir da função objetivo, que representa o caminho para a melhor solução. Quanto maior a nota de um indivíduo, mais próximo ele está da solução ótima.
2. Selecionam-se os melhores indivíduos da população para que estes possam se combinar aos pares, em um processo denominado *crossover*. Através destes cruzamentos, os indivíduos, trocam parte de seus *bits*, dando origem a uma nova geração para a população.
3. Submete-se cada um dos indivíduos da nova geração a uma eventual “mutação”. Neste processo, *bits* de um determinado indivíduo podem mudar seu valor, mediante a ocorrência de um evento de baixa probabilidade.
4. Aplica-se o “elitismo” na nova geração. Isto implica em trazer para a nova geração o(s) melhor(es) indivíduo(s) da geração atual, mantendo assim preservada a melhor solução obtida até o momento.
5. Repete-se este procedimento, a partir do passo inicial, até que sejam satisfeitos os critérios evolutivos esperados, ou seja, um de seus indivíduos aproxime-se da solução ótima.

A aptidão de um indivíduo x_i da população pode ser representada através de uma função $f(x_i)$, a qual indica quão próximo este indivíduo se encontra da solução ótima para o problema estudado. Assim, em uma população composta por N indivíduos, cada um destes terá a sua própria aptidão definida através de $f(x)$.

A análise e comparação das aptidões de uma população com N indivíduos irá estabelecer a probabilidade $p(x_i)$ que um indivíduo i terá de gerar descendentes através do processo de *crossover*. Para o caso onde esta probabilidade é diretamente proporcional ao valor numérico de $f(x)$, então a mesma pode ser calculada através da Equação (5).

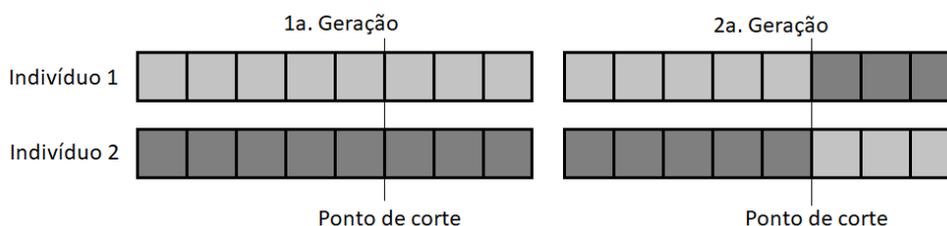
$$p(x_i) = \frac{f(x_i)}{\sum_{k=1}^N f(x_k)} \quad (5)$$

Caso a probabilidade aumente à medida em que o valor numérico da função objetivo tenda a zero, como é o caso do modelo discutido neste artigo, então a probabilidade de seleção de um indivíduo deverá ser calculada como apresentado na Equação (6).

$$p(x_i) = \frac{1 - \left(\frac{f(x_i)}{\sum_{k=1}^N f(x_k)}\right)}{(N-1)} \quad (6)$$

O procedimento de *crossover* a que serão submetidos os indivíduos com melhor aptidão pode ser visualizado através da Figura 2.

Figura 2 – Procedimento de crossover entre dois indivíduos de uma população.

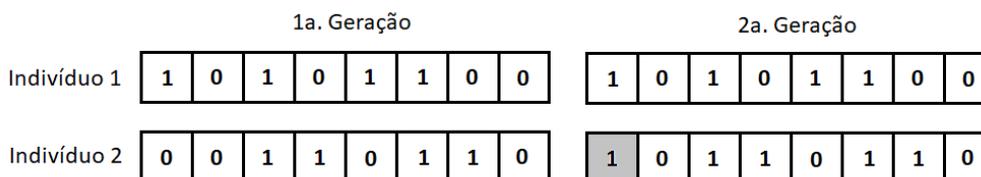


Fonte: Elaborado pelos autores.

Previamente à realização do *crossover*, deve ser definido, aleatoriamente, um “ponto de corte”, que indicará a região a ser trocada entre os dois indivíduos. Através da Figura 2 pode ser observado que, a partir do ponto de corte, há a troca de informação entre o par.

Por sua vez, o procedimento de mutação é representado através da Figura 3.

Figura 3 – Mutação ocorrida em um dos indivíduos da população.



Fonte: Elaborado pelos autores.

A mutação é um evento aleatório, de baixa probabilidade, onde pode ocorrer a inversão do valor de um ou mais *bits* dos indivíduos de uma determinada população. Ao se aplicar o procedimento de mutação em um algoritmo genético, deve-se ter cuidado para não tornar este processo um evento com frequência elevada, o que poderia causar a degeneração da solução representada pela população.

Em um AG, o “elitismo” tem por objetivo preservar as melhores características da geração atual, transportando-as para a geração seguinte. Assim, o indivíduo mais apto (ou os mais aptos) passa diretamente da geração atual para a nova geração, sem sofrer modificações.

4 ALGORITMOS GENÉTICOS E A CINEMÁTICA DO ROBÔ PLANAR

Esta seção descreve o procedimento adotado em relação ao uso de um AG para a solução do problema cinemático inverso de um robô planar. A solução para o modelo descrito no início deste texto exige o cálculo dos ângulos das juntas, θ_1 e θ_2 , que satisfazem o posicionamento do extremo livre do robô em um determinado ponto do plano cartesiano.

A função objetivo a ser utilizada no processo de otimização será fundamentada na distância entre a posição (X_P', Y_P') , obtida com os valores θ_1 e θ_2 resultantes do AG e aplicados às equações (1) e (2), e a posição (X_P, Y_P) arbitrariamente definida para a extremidade do robô. Esta função é representada através da Equação (7).

$$d = \sqrt{(X_P - X_P')^2 + (Y_P - Y_P')^2} \quad (7)$$

Para a construção do AG, conforme descrito previamente, é necessário codificar as variáveis θ_1 e θ_2 em um formato binário. Considerando que os valores destes ângulos estarão restritos ao intervalo entre 0 e 2π , optou-se por uma codificação em dez *bits*, sendo os três *bits* mais significativos reservados à parte inteira e os demais sete *bits* reservados à parte fracionária do número. A implementação computacional do AG também exige que um indivíduo seja representado pela seguinte estrutura de dados:

Estrutura Indivíduo

```
{  
Real  $\theta_1$ ;  
Real  $\theta_2$ ;  
Real aptidão;  
Real probabilidade de seleção;  
};
```

Nesta estrutura, θ_1 e θ_2 definem a solução representada pelo indivíduo; **aptidão** sintetiza a nota atribuída a esta solução; e **probabilidade de seleção** representa a probabilidade do indivíduo ser selecionado para o procedimento de *crossover*.

A aptidão será calculada através da Equação (7), devendo ser lembrado que, por se tratar de uma distância, tanto menor seu valor numérico, tanto maior será a aptidão do indivíduo. Isto, por sua vez, implica no uso da Equação (6) para o cálculo da probabilidade de seleção do indivíduo.

Por sua vez, uma população com N indivíduos irá descrever o espaço de busca e evolução em direção à solução ótima. Esta população é descrita como um vetor de N indivíduos na implementação computacional deste algoritmo.

Indivíduo **Geração_atual**[N];

Dessa forma, a partir de uma geração inicial com N indivíduos, o algoritmo do Quadro 1 pode ser utilizado para a determinação das gerações futuras, até que uma solução ótima seja alcançada.

Quadro 1 – AG para solução da cinemática inversa de um robô planar

1. Selecione, aleatoriamente, N indivíduos para a 1ª. geração da população. Se existirem indivíduos cujos valores de θ_1 e θ_2 estejam fora dos limites permitidos ($0 \leq \theta < 2\pi$), substitua estes indivíduos por outros.
2. Calcule a aptidão de cada indivíduo, através das equações (1), (2) e (7).
3. Calcule a probabilidade de seleção de cada indivíduo, através da Equação (6).
4. Encontre o indivíduo com maior aptidão nesta geração.
5. Enquanto a maior aptidão não satisfizer o critério de parada:
 - a. Para iniciar uma nova geração, repita $N/2$ vezes:
 - i. Selecione, por sorteio, com base em sua probabilidade de seleção, dois indivíduos da geração atual.
 - ii. Realize o *crossover* dos dois indivíduos selecionados.
 - iii. Armazene os dois indivíduos gerados por *crossover* na nova geração.
 - iv. Fim.
 - b. Submeta todos os indivíduos da nova geração a um eventual processo de mutação.
 - c. Se na nova geração existirem indivíduos cujos valores de θ_1 e θ_2 estejam fora dos limites permitidos ($0 \leq \theta < 2\pi$), substitua estes indivíduos pelos mais aptos da geração atual.
 - d. Selecione o indivíduo menos apto da nova geração e substitua-o pelo indivíduo mais apto da geração atual.
 - e. Calcule a aptidão de cada indivíduo da nova geração.
 - f. Calcule a probabilidade de seleção de cada indivíduo da nova geração.
 - g. Selecione o indivíduo mais apto da nova geração.
 - h. Faça a nova geração ser a geração atual.
 - i. Fim.
6. Fim.

Fonte: Elaborado pelos autores.

Alguns pontos deste algoritmo são destacados a seguir. Uma vantagem considerável do mesmo consiste no fato de que ele não exige uma solução predefinida para que possa evoluir em direção a uma solução ótima. Dessa forma, o Passo (1.) do algoritmo permite criar uma geração inicial aleatória como ponto de partida. Como única restrição, considere que os valores de θ_1 e θ_2 desta primeira geração devem estar dentro da faixa permitida ($0 \leq \theta < 2\pi$).

O Passo (2.) define a aptidão do indivíduo a partir de seus valores de θ_1 e θ_2 . Através das equações (1) e (2), os valores destes ângulos irão determinar a posição (X_P', Y_P') que a solução representa. Por sua vez, através da Equação (7) determina-se a distância entre esta posição e aquela desejada para o extremo do robô (X_P, Y_P) . Esta distância representa a "aptidão" do indivíduo, sendo melhor tanto menor for o seu valor numérico.

A probabilidade de seleção de um determinado indivíduo para *crossover*, apontada no Passo (3.), deriva de sua aptidão, ou seja, da distância que sua respectiva solução representa. Esta probabilidade, calculada através da Equação (6), leva em conta a aptidão dos demais indivíduos, possuindo maior probabilidade de seleção aquele que se encontra a uma menor distância do objetivo (X_P, Y_P) .

Encontrar o indivíduo com maior aptidão, conforme previsto no Passo (4.), permite verificar quão distante a simulação está de seu critério de parada. Este critério, mencionado no Passo (5.), pode ser ajustado de diversas formas. Na simulação desenvolvida neste trabalho, optou-se por

estabelecer como critério de parada para a evolução da população a maior aptidão individual com valor numérico inferior a um limite preestabelecido. Outros critérios de parada a serem considerados podem ser um número determinado de gerações, a perda de diversidade de uma população ou a convergência para uma determinada solução após um certo número de gerações (HOLLAND, 1975).

A seleção de indivíduos para *crossover* por sorteio, indicada no Passo (5.a.i), pode ser interpretada através do formato de uma “loteria”, baseada na aptidão e na probabilidade de um indivíduo ser selecionado. Neste modelo de sorteio, atribui-se a cada indivíduo da população uma quantidade de “bilhetes de loteria”, a qual é proporcional à sua probabilidade de seleção. Assim, através do sorteio do “bilhete premiado”, define-se o indivíduo (“dono do bilhete”) que será selecionado para o *crossover*.

O *crossover*, indicado no Passo (5.a.ii), é o evento que desencadeará o nascimento de uma nova geração. Além disso, esta nova geração poderá passar por um processo de mutação, conforme previsto no Passo (5.b), aumentando a diversidade da população. Conforme indicado previamente, a mutação é um evento que deve ser utilizado com cautela em AGs, uma vez que altas taxas de mutação podem levar à degeneração da população e, por consequência, à perda da solução que ela representa.

Crossover e/ou mutações são procedimentos que, eventualmente, podem dar origem a indivíduos “degenerados”, ou seja, cujos valores de θ_1 e θ_2 se encontram fora dos limites permitidos. Este tipo de ocorrência pode ser contornado através da substituição dos indivíduos degenerados pelos mais aptos da geração atual (Passo (5.c)). Neste mesmo sentido, havendo ou não indivíduos degenerados, é conveniente preservar o indivíduo mais apto da geração atual (elitismo). Para tanto, o Passo (5.d.) propõe a substituição do indivíduo menos apto da nova geração pelo mais apto da geração atual.

Uma vez definida a nova geração, os parâmetros de aptidão e probabilidade de seleção de cada indivíduo devem ser recalculados. Finalmente, faz-se a nova geração tornar-se a geração atual e, caso o critério de parada não tenha sido atingido, repetem-se os procedimentos para a criação de mais uma geração.

5 RESULTADOS OBTIDOS E SUA DISCUSSÃO

O modelo apresentado ao longo do texto foi simulado para um robô planar, tendo seu antebraço L_1 comprimento de cinco unidades e seu braço L_2 quatro unidades. Os ângulos θ_1 e θ_2 , que definem a solução representada por um indivíduo, foram codificados em dez *bits*, sendo três *bits* dedicados à representação da parte inteira do ângulo e sete *bits* destinados à representação da parte fracionária. Os ângulos, por sua vez, estão restritos ao intervalo $0 \leq \theta < 2\pi$.

Na implementação desenvolvida foi utilizada uma população composta por dez indivíduos. O ponto de corte para o *crossover* é definido aleatoriamente, cada vez que esta operação é executada, sendo limitado a no máximo 60% dos *bits*, contados a partir do *bit* menos significativo. Por sua vez, a probabilidade de mutação foi limitada a 1%, de forma a ser evitada a degeneração da população.

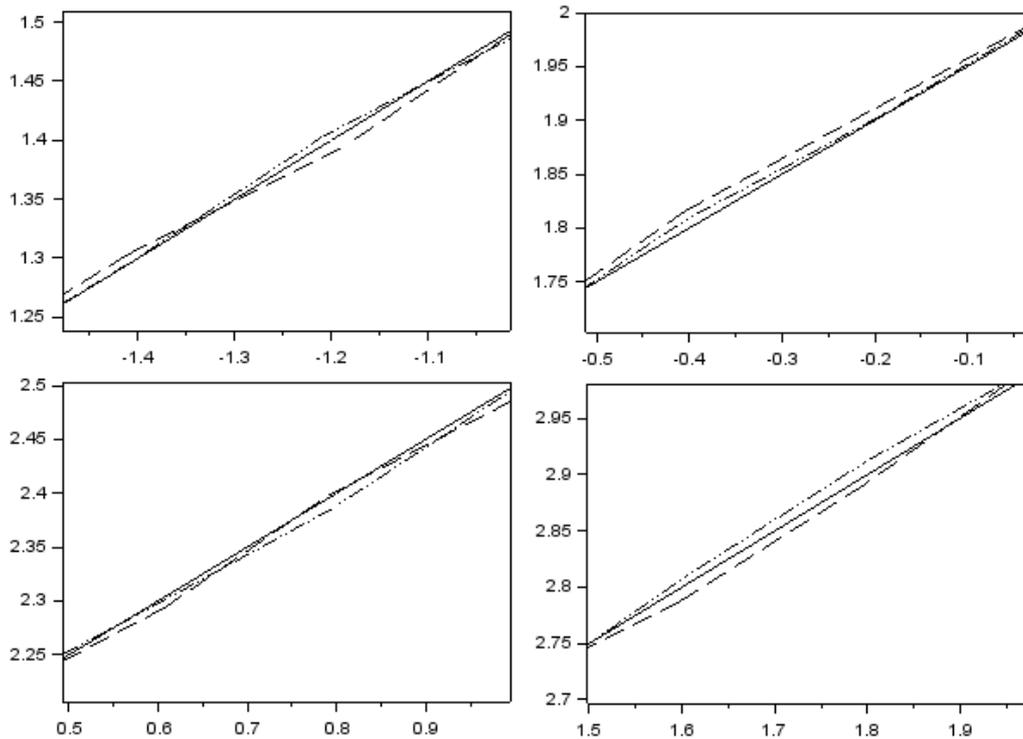
A simulação em questão consistiu no deslocamento do extremo livre do robô, segundo uma trajetória linear descrita pela Equação (8).

$$Y_P = 0,5.X_P + 2 \quad (8)$$

Esta trajetória foi limitada ao intervalo $-2 \leq X_P \leq 2$. O critério de parada utilizado para o encerramento do cálculo dos ângulos θ_1 e θ_2 associados a uma posição (X_P, Y_P) consiste na obtenção de uma distância inferior a 0,02 unidades em relação a (X_P, Y_P) calculado a partir da

Equação (8). A Figura 5 apresenta a sobreposição das trajetórias desejada, a qual foi determinada através da Equação (8), e obtidas a partir da simulação pelo AG.

Figura 5 – Sobreposição das trajetórias desejada e simulada.



Fonte: Elaborado pelos autores.

Na Figura 5, a linha sólida representa a trajetória desejada para o robô, a qual foi calculada através da Equação (8). Por sua vez, as linhas tracejadas representam as trajetórias simuladas com AG (soluções com “cotovelo para cima” e “cotovelo para baixo”). O gráfico permitir observar que as trajetórias obtidas a partir da simulação são significativamente próximas da trajetória desejada.

6 CONSIDERAÇÕES FINAIS

O modelo apresentado ao longo deste texto foi aplicado em uma disciplina de robótica. Os resultados deste estudo e sua aplicação no ensino de engenharia demonstraram que o uso de algoritmos genéticos para a solução do problema cinemático inverso da robótica é viável, contribuindo para um maior envolvimento dos alunos, propiciando melhores resultados no decorrer da aprendizagem. Uma vantagem considerável para o uso de AGs neste tipo problema consiste em sua implementação computacional relativamente simples. O algoritmo apresentado também é paralelizável, podendo ser fragmentado em um cluster de computadores, diminuindo o tempo de cálculo para a obtenção de soluções mais precisas. A precisão do algoritmo, por sua vez, depende fortemente da quantidade de dígitos utilizadas para a codificação binária de um indivíduo. Também foi possível notar, durante as aulas, que o uso de algoritmos genéticos tornou as simulações de modelos robóticos mais eficientes, otimizando o tempo de desenvolvimento de projetos e assim contribuindo positivamente para o estudo desta área no decorrer da disciplina.

REFERÊNCIAS

- BING, X.; MENGJIE, Z.; BROWNE, W. N. A Survey on Evolutionary Computation Approaches to Feature Selection. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 4, p. 606-626, 2016.
- CRAIG, John J. **Introduction to Robotics: Mechanics and Control**. 4th.edition, New York: Pearson, 2017.
- DARWIN, Charles. **A origem das Espécies e a Seleção Natural**. São Paulo: Madras, 2004.
- GOLDBERG, David E. **Genetic Algorithms in Search, Optimization and Machine Learning**. Boston: Addison-Wesley Longman Publishing Co., 1989.
- GUPTA, A.; YEW-SOON, O.; LIANG, F. Multifactorial Evolution: Toward Evolutionary Multitasking. **IEEE Transactions on Evolutionary Computation**, v. 20, n. 3, p. 343-357, 2016.
- HAUPT, Randy L.; HAUPT, Sue E. **Practical Genetic Algorithms**. 2nd. Edition, New Jersey: John Wiley & Sons, 2004.
- HOLLAND, John H. **Adaptation in natural and artificial systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence**. Ann Arbor: The University of Michigan Press, 1975.
- KRAMER, Oliver. **Genetic Algorithm Essentials**. New York: Springer, 2017.
- MILLER, Frederic P.; VANDOME, Agnes. F.; McBREWSTER, John. **Gregor Mendel**. London: Alphascript Publishing, 2009.
- MILLER, Mark R.; MILLER, Rex. **Robots and Robotics: Principles, Systems, and Industrial Applications**. USA: Mc Graw Hill Education, 2017.

TEACHING ROBOTICS THROUGH GENETIC ALGORITHMS

Abstract: *The study of articulated robots necessarily goes through the development of their kinematic models. In turn, the kinematics of a robot can be described through its direct and inverse models. The inverse kinematic model, through which the state of the joints is obtained as a function of the desired position for the free end of the robot, is usually described algebraically. However, this representation is often difficult to obtain. Thus, while the exact determination of the inverse kinematic model is unquestionable, the use of genetic algorithms in the design stage can be very attractive because it allows predicting the behavior of the robot before the formal development of its model. In this sense, the results of this work present a relatively fast way to simulate the inverse kinematic model, which can be useful in teaching robotics in engineering, allowing the student to have a broader view of the model, coming to identify points that must be corrected or that can be optimized in the structure of a robot. It can be concluded that the use of genetic algorithms in robotics is feasible, having as main advantages its easy computational implementation and its precision in the representation of kinematic models.*

Key-words: *Evolutionary Algorithms. Process Optimization. Computational Simulation.*