

# AVALIAÇÃO DA PERFORMANCE DO ALGORITMO GENÉTICO PARA A SOLUÇÃO DE PROBLEMAS DE TRAJETÓRIAS EM REGIÕES SINGULARES – UMA AVALIAÇÃO SISTEMÁTICA SOBRE UM MANIPULADOR ROBÓTICO

VICTOR N. PRAXEDES<sup>1</sup>, VICTOR HUGO C. DE ALBUQUERQUE<sup>2</sup>

1. Laboratório de Sistemas de Controle, Centro de Ciências Tecnológicas, Universidade de Fortaleza, Av. Washington Soares, 1321, Bloco L, Sala 04, 60811-905, Edson Queiroz, Fortaleza, Ceará, Brasil.  
*E-mail: victor.praxedes@gmail.com*
2. Programa de Pós Graduação em Informática Aplicada, Centro de Ciências Tecnológicas, Universidade de Fortaleza, Av. Washington Soares, 1321, Bloco J, Sala 30, 60811-905, Edson Queiroz, Fortaleza, Ceará, Brasil.  
*E-mail: victor.albuquerque@unifor.br*

**Abstract**— This paper presents two algorithms capable of solving singularities while moving a robot arm through a trajectory. Those methods are: local and global genetic algorithm. Polynomial trajectories up to the 3rd degree are used to evaluate each algorithm performance by analyzing, mainly, the error, number of singularities and computation cost. The observed results showed that the global genetic algorithm had the best performance, once it followed the 3rd degree trajectory with the least error, number of singularities and computation cost. Therefore, there is enough evidence that show that the global genetic algorithm is the better technique, between the other presented in this paper, for tracking trajectories going through singular regions.

**Keywords**— Trajectory tracking, singularity, genetic algorithm, robotic arm, robotic manipulator, robot control and planning.

**Resumo**— Este trabalho apresenta dois algoritmos para solucionar o problema de singularidades durante uma trajetória para um robô manipulador de 6 graus de liberdade. Os métodos considerados são: algoritmo genético local e global. São utilizadas trajetórias polinomiais de até 3º para avaliar a performance de cada um dos métodos, analisando, principalmente, o erro, a quantidade de singularidades e o custo computacional. A partir dos resultados obtidos, pôde-se observar que o melhor desempenho foi o do algoritmo genético global, uma vez que apresenta o menor erro, menor quantidade de singularidades e menos custo computacional para percorrer uma trajetória do 3º grau. Portanto, pode-se concluir que o algoritmo genético global é a técnica mais adequada, entre as estudadas neste trabalho, para seguir uma trajetória que atravessa regiões singulares.

**Palavras-chave**— Trajetória, singularidades, algoritmo genético, braço robótico, manipulador robótico, controle e planejamento de robôs.

## 1 Introdução

O uso de manipuladores robóticos, juntamente com as tecnologias de projeto/desenho assistido por computador (*Computer-Aided Design - CAD*) e manufatura assistida por computador (*Computer-Aided Manufacturing - CAM*), estão em crescente expansão na área de automação do processo de manufatura como, por exemplo, aplicações industriais.

Processos automatizados tem por finalidade a redução do custo na fabricação de materiais devido à redução de mão de obra, menor perda de materiais, maior quantidade de produtos acabados, redução de estoque, bem como a redução do tempo de produção e setup do processo. Além do mais, por eliminar a subjetividade de um operador, os processos automatizados garantem uma maior precisão e rapidez durante a fabricação de um determinado produto. Por estas razões, busca-se uma constante melhoria da tecnologia usada nos processos automatizados, referindo-se, principalmente, aos manipuladores robóticos (CRAIG 2004; SPONG 2005).

À medida que a customização da tecnologia aplicada nos manipuladores vai se tornando mais eficiente e eficaz, os braços robóticos começaram a ser utilizados em outras áreas do conhecimento que necessitam de maior precisão como, por exemplo, a

Biomédica (MORASSO et al. 1977; VLUGT et al. 2003), e outras que não é possível a presença humana, uma vez que apresentam um ambiente insalubre. Como exemplos deste ambiente, pode-se citar: exploração submarina, processos em uma indústria petroquímica, químicas, gás e óleo, bem como lugares de difícil acesso (SILVA et al. 2009; FREITAS et al. 2011; NORMANDO et al. 2010; ALBUQUERQUE et al. 2010). Devido a sua complexidade e diferentes aplicações, os manipuladores robóticos envolvem, principalmente, conhecimentos de Engenharia Elétrica, Eletrônica, Mecânica e Ciência da Computação.

Independente da aplicação do manipulador robótico, todas elas possuem algo em comum: uma trajetória a ser percorrida. Toda e qualquer aplicação em um braço robótico requer que ele se desloque a partir de um ponto inicial e consiga realizar precisamente todo um percurso até seu destino final. Para tal, se faz necessário que ele siga um caminho específico (pré-determinado), que pode ser representado por uma linha reta ou por um trajeto sinuoso para desviar de obstáculos, por exemplo. Além da precisão a ser seguida, dependendo da aplicação, as trajetórias possuem restrições de velocidade, espaço útil e tempo de execução, aumentando ainda mais a complexidade e necessidade de controladores robustos capazes de atender tais requisitos.

Neste contexto, durante o percurso da trajetória, podem aparecer situações que são compostas por pontos, nas quais a configuração desejada para os ângulos das juntas do robô (COLOMÉ 2011) não os alcançam, sendo denominada comumente por singularidade. Quando o manipulador passa por uma região de singularidade, cuja configuração/implementação é mal condicionada, ele se comporta de maneira indesejada, apresentando picos de velocidades e saindo da trajetória, sendo necessário utilizar métodos robustos para solucionar esse problema de maneira estável como, por exemplo, a reparametrização da trajetória e do perfil de velocidade das juntas (LLOYD 1998), a utilização do método de Levenberg-Marquardt (BUSS 2004), a remoção das componentes degenerativas da matriz do Jacobiano (OETOMO e JR 2009) e através da matriz do Jacobiano estendida (ŠOCH e LÓRENCZ 2005).

Os métodos propostos neste trabalho solucionam singularidades sem a necessidade de reformular as configurações singulares, caso o robô seja modificado fisicamente. Desta forma, é possível solucionar problemas de singularidades em tempo real, sendo esta uma das principais contribuições deste trabalho.

Entre os objetivos específicos para o algoritmo, pode-se considerar:

- Percorrer trajetórias de 2° e 3° graus com o menor erro e tempo possível com precisão;
- Comparar a eficiência entre o algoritmo genético local e o algoritmo genético global;
- Solucionar, com robustez, problemas de singularidades sem desvios de sua trajetória nem atingindo valores impraticáveis para as velocidades das juntas.

## 2 Metodologia

### 2.1 Singularidades e a matriz Jacobiana

A matriz Jacobiana é, para o manipulador de 6 graus de liberdade, uma matriz quadrada de 6x6 com seu posto, que corresponde ao número de linhas ou colunas linearmente independentes  $\leq 6$ . A relação entre as velocidades das ferramentas e das juntas são definidas pela matriz Jacobiana inversa (WHITNEY 1969). Portanto, quando o posto for máximo, todas as velocidades são linearmente independentes e a matriz jacobiana é caracterizada como bem condicionada ou invertível. Quando o posto do Jacobiano é menor que o seu posto máximo, o manipulador não consegue exercer movimentos em qualquer direção, efetivamente, perdendo um ou mais graus de liberdade, zerando seu determinante. Desta forma, sua matriz inversa não é computável, sendo conceituada de matriz mal condicionada (SPONG 2005; OETOMO e JR 2009).

Entretanto, o comportamento degenerativo do ponto singular não ocorre somente quando a ferramenta está exatamente posicionada em um ponto específico. Nas proximidades de um ponto singular, o determinante do jacobiano é muito pequeno, cau-

sando altas velocidades de juntas a partir de baixas velocidades da ferramenta (WILDAN et al. 2008), sendo necessário, portanto, definir regiões singulares ao invés de pontos singulares. Essas são definidas por regiões em que o determinante do Jacobiano é menor que uma constante (OETOMO e JR 2009), que, para os métodos propostos neste trabalho, foi determinado empiricamente o valor de 0,001. Assim, qualquer ponto que possua o determinante menor que 0,001, será considerado como uma singularidade, que será solucionada ou não através dos algoritmos propostos.

### 2.2 Algoritmo Genético

O algoritmo genético (AG) é uma abordagem metaheurística utilizado para solucionar problemas de busca/minimização, bem como seleção de features para problema de classificação. O algoritmo é designado para encontrar os melhores indivíduos dentro de uma população, os quais minimizam alguma função matemática, chamada de função *fitness*, que corresponde à aptidão de um “indivíduo de sobreviver”. No caso do algoritmo implementado, o indivíduo é representado por uma configuração das 6 juntas do manipulador, a população é uma porção de configurações do manipulador geradas a partir de ângulos randômicos para orientação da ferramenta e a função de *fitness*, a qual se deseja minimizar, é o inverso do valor do determinante do Jacobiano. Quanto menor o determinante, maior será a função *fitness*, fazendo com o que o algoritmo efetivamente maximize o valor do determinante do Jacobiano. O AG realiza operações como cruzamento e mutação, de forma que, iteração após iteração, os indivíduos com a menor função *fitness* são escolhidos para a próxima iteração. Cada iteração do AG é chamada de “geração” (GOLDBERG 1989; MICHALEWICZ 1996).

#### 2.2.1 Algoritmo Genético Local

Utilizando a toolbox do MATLAB R2009b, foi implementado um AG no qual a função *fitness*, descrita na sessão anterior, corresponde ao inverso do valor do determinante do Jacobiano, visando maximizar o determinante para sair região singular. Duas funções para restringir os resultados possíveis do AG foram criadas: (i) a variação no ângulo de orientação no eixo X entre a configuração singular e configuração calculada não pode ser maior que 60°, evitando movimentos muito amplos em torno do eixo X e (ii) pode haver um pequeno erro de 0,000001 entre o ponto da singularidade e o ponto calculado. Em muitos casos, não é possível encontrar uma solução para um determinado ponto singular, mas existe solução para um de seus pontos adjacentes. Esta última restrição torna possível resolver esse tipo de singularidade em específico.

As populações iniciais geradas pelo toolbox de AG do MATLAB são randômicas e, muitas vezes, podem fazer com que o algoritmo fique “preso” em

uma mínima local, considerando que aquela configuração já minimizou a função *fitness* para a menor possível. A solução para o problema da mínima local é discutida na sessão seguinte.

### 2.2.2 Algoritmo Genético Global

Utilizando as funções de otimização da toolbox de AG do MATLAB R2009b, pode-se definir valores mínimos e máximos para a população inicial de forma que ela abranja o valor mínimo global da solução. Dessa forma, foi considerado que suas populações iniciais são os ângulos X e Y atuais da orientação da ferramenta  $\pm 100^\circ$ . Assim, o AG converge para uma mínima global, em que o Jacobiano será o maior possível para aquele ponto. As funções de *fitness* e as restrições são as mesmas apresentadas na sessão 2.2.1.

## 3 Resultados

Para avaliar o desempenho dos métodos de controle de trajetória e singularidade, nomeados por algoritmo genético local (AGL) e global (AGG), medidas estatísticas como precisão e custo computacional foram consideradas, bem como parâmetros importantes como, por exemplo, velocidade de deslocamento da ferramenta e quantidades de singularidades. Para todos os experimentos realizados neste trabalho, foi utilizado um Macbook Pro, Intel Core i5 2.4GHz com 4 GB 1067 Mhz DDR3 usando o sistema operacional Windows.

Durante a comparação dos métodos anteriormente citados, foram geradas trajetórias parabólicas (segundo grau) e elípticas (terceiro grau) considerando dois pontos no espaço cartesiano. Desta forma, é possível analisar o comportamento dos métodos com relação ao comportamento da velocidade linear da ferramenta, velocidade das juntas e o erro em relação à trajetória desejada para regiões sem e com, sendo esta última representada por linhas tracejadas seguidas por "\*" na cor vermelha em todos os gráficos das figuras aqui ilustradas, e finalmente, a frequência em que ocorrem as singularidades.

Para avaliar a influência do custo computacional sobre o erro e quantidades de singularidades geradas, foram definidos, empiricamente, tempos na ordem de 1 e 5 segundos (s), uma vez que, para tempos superiores (maior custo computacional), as velocidades e os erros são insignificantes em relação ao tempo de 5 s. Adicionalmente, todas as trajetórias foram traçadas no plano XY com a ferramenta, sendo esta iniciada manualmente na posição de  $180^\circ$  do plano XY com o eixo Z fixo, representando com fidedignidade várias aplicações reais.

Nas Figuras 1a e 1b são apresentados os gráficos de uma simulação com uma trajetória parabólica partindo do ponto inicial ( $X= 0,65$ ,  $Y= 0,1$  e  $Z= 0,5$ ) até o ponto final ( $X= 0,65$ ,  $Y= -0,3$  e  $Z= 0,5$ ) para ser executada em 1s, utilizando os algoritmos AGL e AGG, respectivamente.

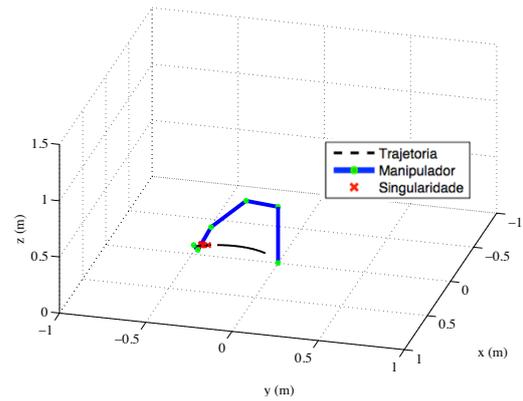


Figura 1a. Trajetória parabólica utilizando o AGL em 1s

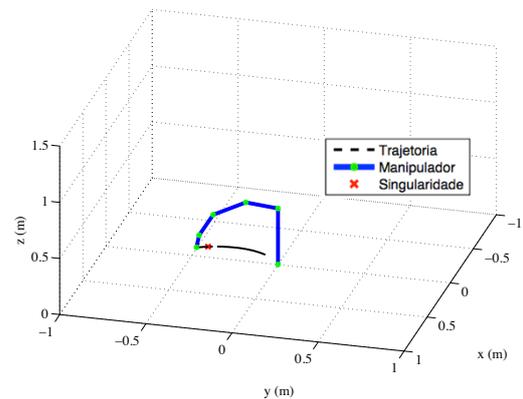


Figura 1b. Trajetória parabólica utilizando o AGG em 1s

Na Figura 1a observa-se uma maior quantidade de regiões singulares em comparação com a Figura 1b, que exibe apenas uma singularidade para o a trajetória. As velocidades das juntas estão demonstradas nas Figuras 2 e 3.

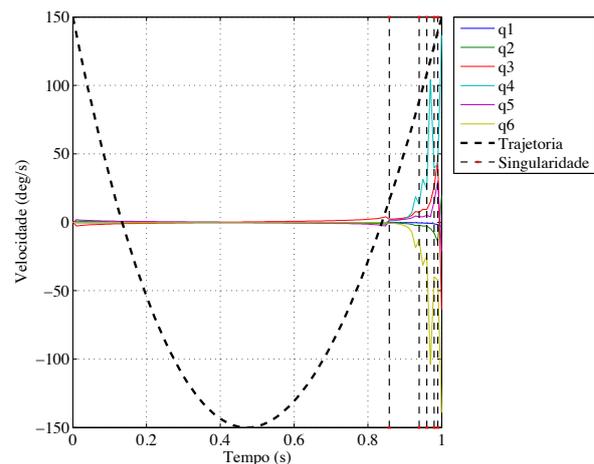


Figura 2. Velocidade das juntas na trajetória parabólica utilizando o AGL durante 1s

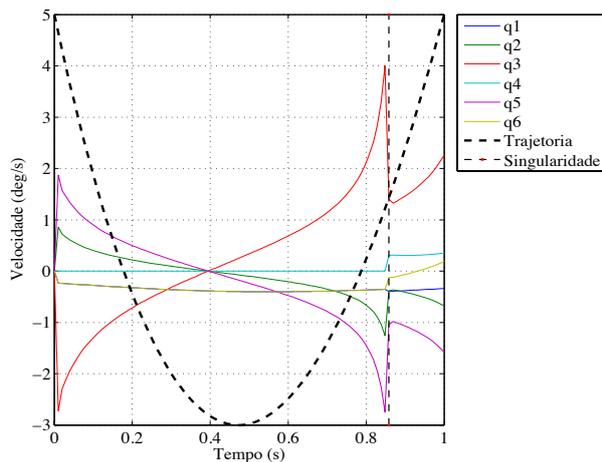


Figura 3. Velocidade das juntas na trajetória parabólica utilizando o AGG durante 1s

Claramente, o AGG (Figura 3), para percorrer a região singular, necessita de velocidades de juntas menores quando comparado ao AGL (Figura 2), em que o valor máximo da velocidade das juntas é, aproximadamente, 26 vezes maior. As Figuras 4 e 5 ilustram, respectivamente, a velocidade da ferramenta para o AGL e para o AGG.

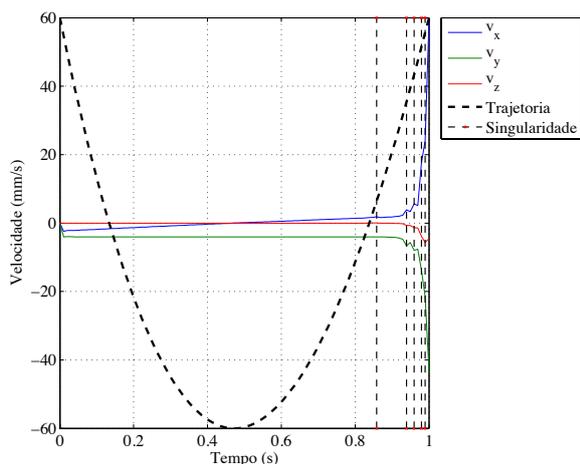


Figura 4. Velocidade da ferramenta na trajetória parabólica utilizando o AGL durante 1s

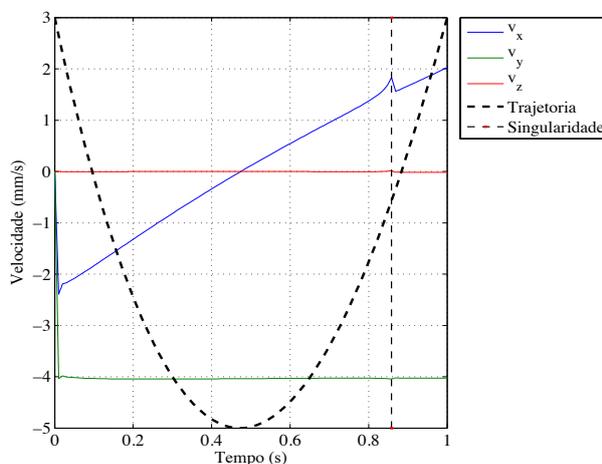


Figura 5. Velocidade da ferramenta na trajetória parabólica utilizando o AGG durante 1s

Semelhantemente as velocidades das juntas, a velocidade da ferramenta para o AGG (Figura 5) é, aproximadamente, 20 vezes maior. Nas Figuras 6 e 7 são mostrados os erros em relação à trajetória, respectivamente, para o AGL e AGG.

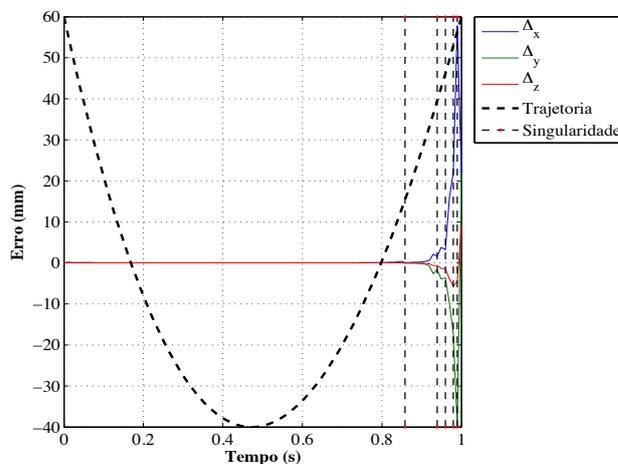


Figura 6. Erro em relação a trajetória parabólica utilizando o AGL durante 1s

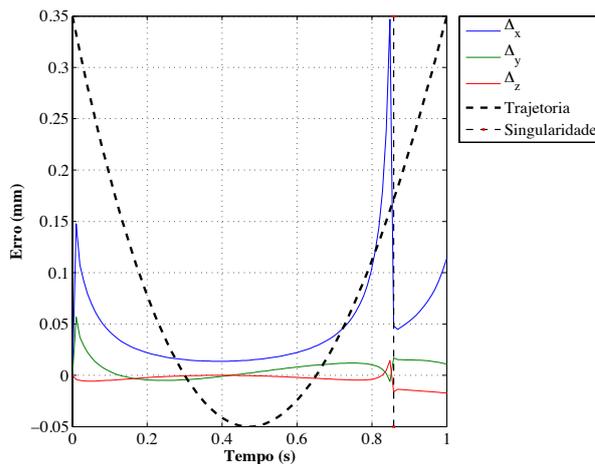


Figura 7. Erro em relação a trajetória parabólica utilizando o AGG durante 1s

O AGG prova ser mais eficiente em relação ao AGL, não apenas por obter erros muito menores (Figura 7), mas também pelo fato de que, após passar por uma singularidade, o valor do erro reduz ao invés de continuar aumentando (Figura 6).

Nas Figuras 8a e 8b são apresentados os gráficos de uma simulação com uma trajetória elíptica partindo do ponto inicial ( $X=0,65$ ,  $Y=0,1$  e  $Z=0,5$ ) até o ponto final ( $X=0,65$ ,  $Y=-0,3$  e  $Z=0,5$ ) para ser executada em 5s, utilizando os algoritmos AGL e AGG, respectivamente. Serão comparados os erros em relação a trajetória para os dois algoritmos.

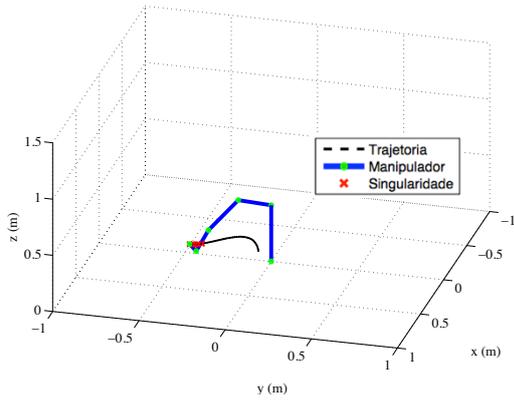


Figura 8a. Trajetória elíptica utilizando o AGL em 5s

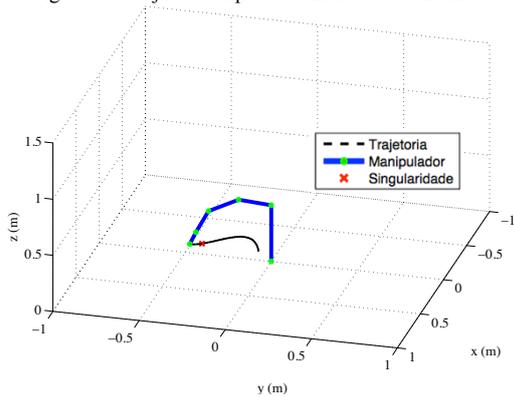


Figura 8b. Trajetória elíptica utilizando o AGG em 5s

Na Figura 8a observa-se, novamente, uma maior quantidade de regiões singulares em comparação com a Figura 8b, que exibe apenas uma singularidade para a trajetória. O erro em relação a trajetória são exibidos nas figuras 9 e 10.

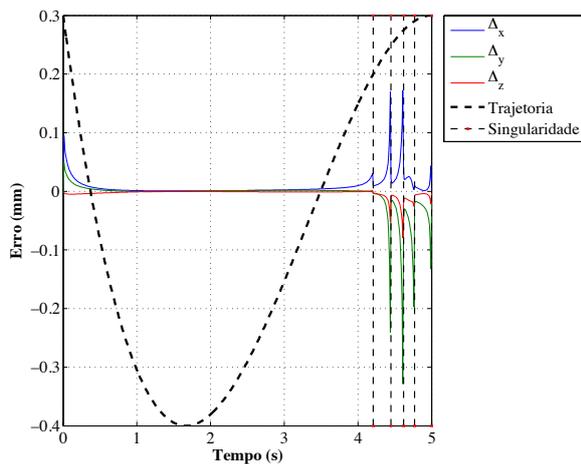


Figura 9. Erro em relação a trajetória elíptica utilizando o AGL em 5s

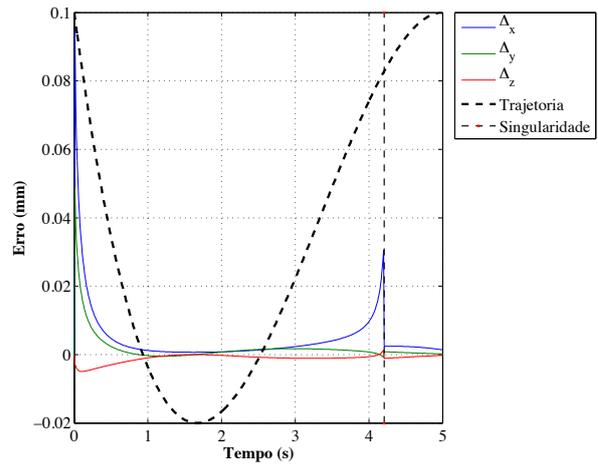


Figura 10. Erro em relação a trajetória elíptica utilizando o AGG em 5s

A partir da Figura 10, nota-se a diferença no valor do erro máximo entre o AGL e o AGG, sendo o AGG mais preciso em até 33,33%. As Tabelas 1 e 2, a seguir, consolidam os resultados obtidos na simulação quanto ao erro médio total, número de singularidades e tempo de computação.

Tabela 1. Comparação para a trajetória de 1s.

1s			
Método	Erro médio total	Nº Sing.	Tempo (s)
AGL	3,9803	5	11,49
AGG	0,1399	1	3,09

Tabela 2. Comparação para a trajetória de 5s.

5s			
Método	Erro médio total	Nº Sing.	Tempo (s)
AGL	0,0244	5	15,04
AGG	0,0138	1	8,76

A partir das Tabelas 1 e 2, pode-se concluir que o tempo e o erro médio total são inversamente proporcionais, quanto menor for o tempo para traçar a trajetória, maior é o erro. Isso está relacionado à alta velocidade que a ferramenta deve atingir para cumprir a trajetória em um menor tempo, gerando, principalmente em regiões singulares, nas quais o determinante do jacobiano é próximo de zero, maiores desvio, uma vez que o aumento da taxa de variação da velocidade da ferramenta dividido por esse determinante resultará em um aumento mais abrupto na taxa de variação da velocidade das juntas.

## 4 Conclusões

Nesse trabalho, foi avaliada a performance do algoritmo genético local e global para a solucionar

problemas de trajetórias que apresentam regiões com singularidades durante o percurso de um manipulador robótico. Para tal, gerou-se duas trajetórias para que fossem testadas as iterações dos algoritmos. A partir das simulações, pode-se concluir que o algoritmo genético global apresentou o menor erro e um custo computacional 271,84% menor do que o algoritmo genético local, bem como 5 vezes menos singularidades.

O método apresentado nesse trabalho demonstrou ser eficiente para solucionar singularidades durante um trajeto de até 3° graus, evitando picos muito altos nas velocidades das juntas e com erro em relação a trajetória da ordem de 0.1 mm.

### Referências Bibliográficas

- Albuquerque V. H. C., Silva E. M., Pereira L., Moura E. P., Araujo V. L. F., Tavares J. M. R. S. Spinodal decomposition mechanism study on the duplex stainless steel UNS S31803 using ultrasonic speed measurements. *Materials and Design*, v. 31, p. 2147-2150, 2010.
- Buss, S.R. Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Squares methods. *IEEE Journal of Robotics and Automation*, Department of Mathematics, p. 681-88, 2004.
- Colomé, A.F. Smooth Inverse Kinematics Algorithms for Serial Redundant Robots. Dissertação apresentada no Institut de Robòtica i Informàtica industrial, Universitat Politècnica de Catalunya, Barcelona, Espanha, 2011.
- Craig, J. J. Introduction to Robotics: Mechanics and Control. 3ª Edição. Upper Saddle River, NJ, EUA, Pearson Prentice Hall, 2004.
- Freitas, V. L. A., Normando, P. G., Albuquerque, V. H. C., Silva, E. M., Silva, A. A. Nondestructive Characterization and Evaluation of Embrittlement Kinetics and Elastic Constants of Duplex Stainless Steel SAF 2205 for Different Aging Times at 425°C and 475°C. *Journal of Nondestructive Evaluation*, 30, 130-160, 2011.
- Goldberg, D. Genetic Algorithms in Search, Optimization and Machine Learning. 1ª Edição. Boton, MA, USA: Addison-Wesley Longman Publishing Co, 1989.
- Lloyd, J. E. Desingularization of nonredundant serial manipulator trajectories using Puiseux series. *IEEE Transactions on Robotics and Automation*, v.14, p. 590-600, 1998.
- Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. 3ª Edição. London, UK: Springer, 1996.
- Morasso P., Sandini G., Tagliasco V., Zaccaria R, Interactive control of a biomedical manipulator. *Annual Review in Automatic Programming*, v. 8, p. 99-205, 1977.
- Normando P. G., Moura E. P., Souza J. A., Tavares S. S. M., Padovese L. R. Ultrasound, eddy current and magnetic Barkhausen noise as tools for sigma phase detection on a UNS S31803 duplex stainless steel. *Materials Science & Engineering: A*, v. 527, p. 2886-2891, 2010.
- Oetomo D., Jr M. H. A. Singularity Robust Algorithm in Serial Manipulators. *Robotics and Computer-Integrated Manufacturing*, v. 25, p. 122-134, 2009.
- Silva, E. M., Albuquerque, V. H. C., Leite, J. P., Varela, A., Moura, E. P., Tavares, J. M. R. S. Phase Transformations Evaluation on an UNS S31803 Duplex Stainless Steel based on Nondestructive Testing. *Materials Science & Engineering: A, Structural Materials: properties, microstructure and processing*, 516, 126-130, 2009.
- Šoch, M., Lórencz R. Solving Inverse Kinematics: A New Approach to the Extended Jacobian Technique. *Acta Polytechnica*, Czech Technical University in Prague, v. 45, 2005.
- Spong, M. W., Hutchinson, S., Vidyasagar, M. Robot Modeling and Control. 1ª Edição. New York, USA, John Wiley and Sons, 2005.
- Vlugt E., Schouten A. C., Van Der Helm F. C., Teerhuis P. C., Brouwn G. G., A force-controlled planar haptic device for movement control analysis of the human arm. *Journal of Neuroscience Methods*, v. 129, p. 151-168, 2003.
- Whitney, D. E. Resolved Motion Rate Control of Manipulators and Human Prostheses. *Man-Machine Systems*, *IEEE Transactions*, v. 10, p. 47-53, 1969.