

PROJETO ÓTIMO DE MANIPULADORES 3R USANDO ALGORITMOS EVOLUTIVOS

Giovana Trindade da Silva Oliveira

Universidade Federal de Uberlândia – Campus Santa Mônica
Av. João Naves de Ávila, n. 2160, 38400-902, Uberlândia – MG – Brasil
gtrindade@mecanica.ufu.br

Sezimária Fátima Pereira Saramago

Universidade Federal de Uberlândia – Campus Santa Mônica
Av. João Naves de Ávila, n. 2160, 38400-902, Uberlândia – MG – Brasil
saramago@ufu.br

Paulo Roberto Bergamaschi

Universidade Federal de Goiás – Campus Catalão
Av. Dr. Lamartine Pinto de Avelar, nº 1120, 75701-220, Catalão – GO – Brasil
paulob@innet.psi.br

Resumo: Neste artigo, o projeto ótimo de manipuladores 3R é definido considerando o volume de trabalho máximo, mantendo as dimensões do robô dentro de uma faixa limitada e impondo restrições geométricas, o que significa redução dos custos de fabricação do equipamento e maior faixa de aplicação operacional. O espaço de trabalho é a região definida pelo conjunto de pontos do espaço possíveis de serem alcançados pelo efetuador. No caso de manipuladores 3R, é representado por um sólido de revolução. O estudo da envoltória é de fundamental importância para o desenvolvimento do projeto de um manipulador, pois representa a fronteira de seu espaço de trabalho. Sua forma depende da arquitetura e das dimensões do robô e tais parâmetros devem ser determinados para que o volume máximo seja obtido. Como métodos de otimização, propõe-se a utilização de alguns algoritmos evolutivos, tais como algoritmos genéticos, evolução diferencial e otimização por colônia de partículas. São apresentadas simulações numéricas para validação da metodologia proposta e comparação das técnicas utilizadas.

Palavras-chave: Robótica, Otimização, Espaço de trabalho, Algoritmos evolutivos

1. INTRODUÇÃO

A aplicação de otimização em vários campos da ciência tem crescido consideravelmente nos últimos tempos. Existem muitos métodos de otimização e cada um deles alcança melhor resultado dependendo do tipo de problema a que são aplicados. A escolha do método depende de uma série de características do problema a ser otimizado, principalmente do comportamento da função que o representa. De acordo com estas características, os métodos podem ser classificados em métodos de programação linear (Bregalda et al., 1988) e programação não-linear. Os métodos de programação não-linear podem ser divididos em métodos determinísticos e métodos naturais.

Os métodos determinísticos utilizam apenas uma solução atualizada em cada iteração. Existe uma ampla bibliografia disponível sobre esta classe de métodos (Reklaitis et al., 1983; Vanderplaats, 1999).

Os métodos naturais são procedimentos iterativos que tentam simular os processos usados na natureza para resolver problemas difíceis. Pode-se citar Recozimento Simulado, Algoritmos Genéticos, Estratégias de Evolução, Evolução Diferencial, algoritmos baseados em inteligência coletiva, entre outros (Goldberg, 1989; Schwefel, 1995; Kennedy et al., 2001). Estes métodos se

baseiam em população de indivíduos, onde cada indivíduo representa um ponto de busca no espaço das soluções candidatas de um dado problema.

O objetivo deste trabalho é maximizar o espaço de trabalho de manipuladores 3R usando os algoritmos evolutivos: Algoritmos Genéticos, Evolução Diferencial e Otimização por Colônia de Partículas e comparar os resultados obtidos.

2. FORMULAÇÃO DO PROBLEMA ÓTIMO

Seja o problema geral de otimização de uma função objetivo de n variáveis de projeto, sujeita a restrições, que definem o espaço de busca viável, escrito como:

$$\text{Minimizar } f(X), \quad X = (x_1, x_2, \dots, x_n)^T \quad (1)$$

$$\text{Sujeito a } \begin{cases} g_j(X) \leq 0, & j = 1, \dots, J \\ h_l(X) = 0, & l = 1, \dots, L \\ x_i^{\inf} \leq x_i \leq x_i^{\sup}, & i = 1, \dots, n \end{cases} \quad (2)$$

Muitos algoritmos evolutivos são desenvolvidos para resolver problemas irrestritos, assim, no caso de problemas com restrições, é necessário introduzir modificações nestes métodos. Neste artigo, é utilizado o conceito de Função de Penalidade (Vanderplaats, 1999), onde os problemas com restrições são transformados em problemas sem restrições adicionando uma função de penalidade $P(X)$ à função objetivo original para limitar as violações das restrições:

$$\Phi(X) = f(X) + r_p P(X), \quad \text{sendo } P(X) = \left[\sum_{j=1}^J \{ \max[0, g_j(X)] \}^2 + \sum_{l=1}^L [h_l(X)]^2 \right] \quad (3)$$

onde $f(X)$ é a função objetivo original dada na Equação 1, $P(X)$ é a função de penalidade imposta, dada pela Equação 3, $g_j(X)$ e $h_l(X)$ são funções de restrições de desigualdade e igualdade, respectivamente, conforme a Equação 2. O escalar r_p é um multiplicador que quantifica a magnitude da penalidade. Observa-se que para a eficiência dos métodos naturais, devem ser adotados elevados valores para o fator de penalidade, garantindo a obediência a todas as restrições.

3. ALGORITMOS EVOLUTIVOS

3.1. Algoritmos Genéticos (AGs)

Os algoritmos genéticos foram introduzidos por Holland, no início da década de 70. Trata-se de um procedimento de busca aleatória baseada na teoria de evolução de Charles Darwin. Esses métodos operam com conjuntos de candidatos, chamados de população, que são constantemente modificados utilizando dois princípios básicos da evolução natural: seleção e variação. Estes princípios tentam representar a competição onde os indivíduos mais aptos reproduzem e repassam seu material genético às gerações futuras, e os indivíduos menos aptos tendem a desaparecer da população (Deb, 2001).

A codificação dos parâmetros da função objetivo pode ser expressa por *bits* (binária) ou por *parâmetros contínuos* (*números reais* ou *pontos flutuantes*).

Os procedimentos básicos que transformam a população ao longo das gerações em busca da solução do problema de otimização caracterizam os operadores genéticos: seleção, cruzamento e mutação. O operador seleção determina quais indivíduos (*cromossomos*) serão escolhidos para o cruzamento. O grau de adaptação de cada cromossomo é caracterizado de acordo com o valor da função objetivo, conhecida como *fitness*. Em um problema de maximização, por exemplo, quanto maior este valor, maior a probabilidade de contribuir à geração seguinte. Existem vários

mecanismos para executar o operador seleção (Goldberg, 1989). Nos sistemas biológicos o cruzamento pode ocorrer durante a reprodução sexuada permitindo a troca de material genético entre dois indivíduos. O algoritmo genético utiliza os cromossomos determinados no processo de seleção para gerarem novos descendentes. O cruzamento é uma forma do algoritmo explorar o espaço de busca e evitar uma convergência prematura.

A mutação é uma modificação aleatória do material genético dos indivíduos, sendo uma outra forma do algoritmo explorar a região de busca. Este operador pode introduzir características que não dependem da população inicial aumentando a diversidade da população, podendo evitar convergência prematura do algoritmo.

A finalização dos procedimentos acima é realizada através de algum critério de parada, sendo que um número máximo de gerações deve sempre ser estabelecido. O melhor indivíduo é adotado como solução do problema (Haupt and Haupt, 1998; Deb, 2001).

3.2. Evolução Diferencial (ED)

Evolução Diferencial é um método de otimização natural, similar aos Algoritmos Genéticos. Foi desenvolvido por Storn e Price em meados da década de 90 (Storn and Price, 1995). O algoritmo é iniciado criando-se uma população escolhida aleatoriamente, geralmente utiliza-se uma distribuição de probabilidade uniforme.

A idéia principal da evolução diferencial é gerar novos indivíduos, denotados vetores modificados ou doadores, pela adição da diferença ponderada entre dois indivíduos aleatórios da população a um terceiro indivíduo. Esta operação é chamada *mutação*.

Os operadores da evolução diferencial são utilizados para manter a diversidade da população e evitar convergências prematuras, assim como nos AGs. São eles:

Mutação: Para a obtenção do vetor doador $V^{(q+1)}$, considere os vetores $X_\alpha^{(q)}$, $X_\beta^{(q)}$ e $X_\gamma^{(q)}$ distintos entre si e escolhidos aleatoriamente em uma população com Np indivíduos. Np deve ser maior ou igual a 4 para garantir uma quantidade suficiente de indivíduos para a execução do método.

Na geração q um par de vetores (X_β, X_γ) define uma diferença $X_\beta - X_\gamma$. Esta diferença é multiplicada por $F > 0$, sendo denotada diferença ponderada e utilizada para perturbar o terceiro vetor X_α ou o melhor vetor X_{best} da população. Este processo pode ser escrito matematicamente como:

$$V^{(q+1)} = X_\alpha^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)}) \quad \text{ou} \quad V^{(q+1)} = X_{best}^{(q)} + F(X_\beta^{(q)} - X_\gamma^{(q)}) \quad (4)$$

onde os índices aleatórios $\alpha, \beta, \gamma \in \{1, \dots, Np\}$ são inteiros distintos entre si, F é um número real e constante pertencente ao intervalo $[0,4, 0,8]$ (Oliveira, 2006) que controla a amplitude da diferença ponderada. A Figura 1 mostra um exemplo bidimensional que ilustra os diferentes vetores que participam da geração do vetor doador $V^{(q+1)}$.

Cruzamento: Considere que para o vetor alvo $X_d^{(q)}$, escolhido aleatoriamente na população, d diferente dos índices α, β e γ , foi gerado um vetor doador. O cruzamento é introduzido para aumentar a diversidade dos indivíduos que sofreram a mutação. Assim, as componentes do vetor experimental $U^{(q+1)}$ são escolhidas pela seguinte comparação:

$$u(i)^{(q+1)} = \begin{cases} v_i^{(q+1)}, & \text{se } r_i \leq Pc. \\ x_{i,d}^{(q)}, & \text{se } r_i > Pc, \quad i = 1, \dots, n \end{cases} \quad (5)$$

onde r_i é um número gerado aleatoriamente entre 0 e 1, Pc é a probabilidade do cruzamento ocorrer, representa a probabilidade do vetor experimental herdar os valores das variáveis do vetor doador, e está compreendida entre 0 e 1, sendo fornecida pelo usuário.

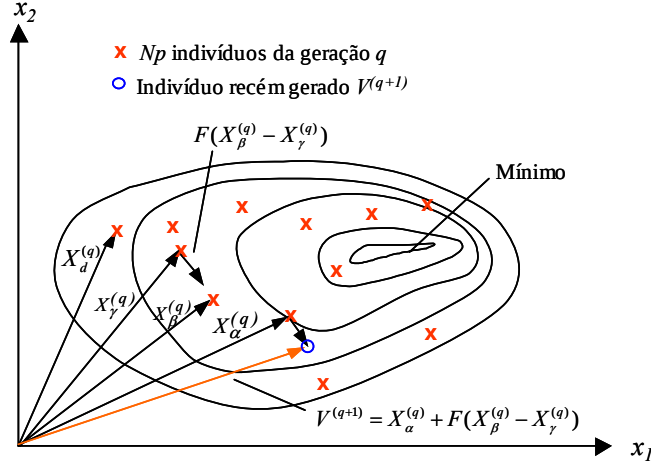


Figura 1: Processo de gerar o vetor doador $V^{(q+1)}$ para uma função bidimensional.

Se após o cruzamento uma ou mais componentes do vetor experimental estiver fora da região de busca, fazem-se as correções:

$$\text{Se } u_i < x_i^{\text{inf}}, \text{ então faz-se } u_i = x_i^{\text{inf}}; \quad \text{Se } u_i > x_i^{\text{sup}}, \text{ então faz-se } u_i = x_i^{\text{sup}} \quad (6)$$

Seleção: A seleção é o processo de produzir melhores descendentes. Diferentemente de outros algoritmos evolutivos, a evolução diferencial não usa hierarquia (elitismo) nem seleção proporcional. Em vez disso, o custo do vetor experimental $U^{(q+1)}$ é calculado e comparado com o custo do vetor alvo $X_d^{(q)}$:

$$\text{Se } f(U^{(q+1)}) \leq f(X_d^{(q)}), \text{ então } X_d^{(q+1)} = U^{(q+1)}; \text{ caso contrário, } X_d^{(q+1)} = X_d^{(q)} \quad (7)$$

O procedimento acima é finalizado através de algum critério de parada, sendo que um número máximo de gerações deve ser estabelecido.

3.3. Otimização por Colônia de Partículas

Otimização por colônia de partículas ou Particle Swarm Optimization (PSO) é uma técnica de otimização desenvolvida na década de 90 por James Kennedy e Russel Eberhart, neste modelo são analisados algoritmos que modelam o “comportamento social” visto em várias espécies de pássaros (Dorigo et al., 1991; Eberhart and Kennedy, 1995).

Dentre vários modelos destaca-se a técnica que se baseia no seguinte comportamento: pássaros estão dispostos aleatoriamente e estes procuram alimento e um local para construir o seu ninho, eles não sabem onde está esse lugar que é único. A indagação é qual o melhor comportamento que os pássaros terão que realizar para conseguir efetuar seu objetivo, parece mais evidente que eles sigam o pássaro que estiver mais próximo do alimento ou do ninho. Inicialmente os pássaros voam sem nenhuma orientação prévia, eles se aglomeram em bandos, até que um consegue encontrar o ninho e atrai os que estiverem mais próximos.

Pelo fato de um pássaro encontrar o ninho, a chance dos outros pássaros também encontrarem aumenta consideravelmente, isto se deve ao fato de a inteligência ser social, ou seja, o indivíduo aprende com o acerto do outro.

3.3.1 O algoritmo Particle Swarm Optimization

Neste algoritmo, os pássaros são chamados de partículas e durante a busca por alimento ou ninho usam de suas experiências e da experiência do bando, possui um vetor de velocidades e outro

de posição. A posição de cada partícula é atualizada de acordo com a velocidade atual, o saber adquirido pela partícula e o saber adquirido pelo bando, sendo dada conforme a Equação 8.

$$x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t \quad (8)$$

onde x_{k+1}^i é a posição da partícula i na iteração $k+1$, v_{k+1}^i é o vetor de velocidade desta partícula, dado na Equação 9, e Δt equivale ao espaço de tempo considerado.

$$v_{k+1}^i = wv_k^i + c_1 r_1 \frac{(p^i - x_k^i)}{\Delta t} + c_2 r_2 \frac{(p^s - x_k^i)}{\Delta t} \quad (9)$$

sendo r_1 e r_2 números gerados aleatoriamente entre 0 e 1. O cálculo da velocidade necessita, ainda, de alguns parâmetros dependentes do problema, que são: a inércia da partícula, w , que controla a capacidade de exploração do algoritmo, ou seja, um valor alto facilita um comportamento mais global, enquanto um valor baixo facilita um comportamento mais local, e os parâmetros de confiança c_1 e c_2 , que indicam o quanto uma partícula confia em si (c_1), e no bando (c_2). A Figura 2 mostra a aplicação da Equação 9, considerando duas partículas se deslocando em um espaço de projeto bidimensional.

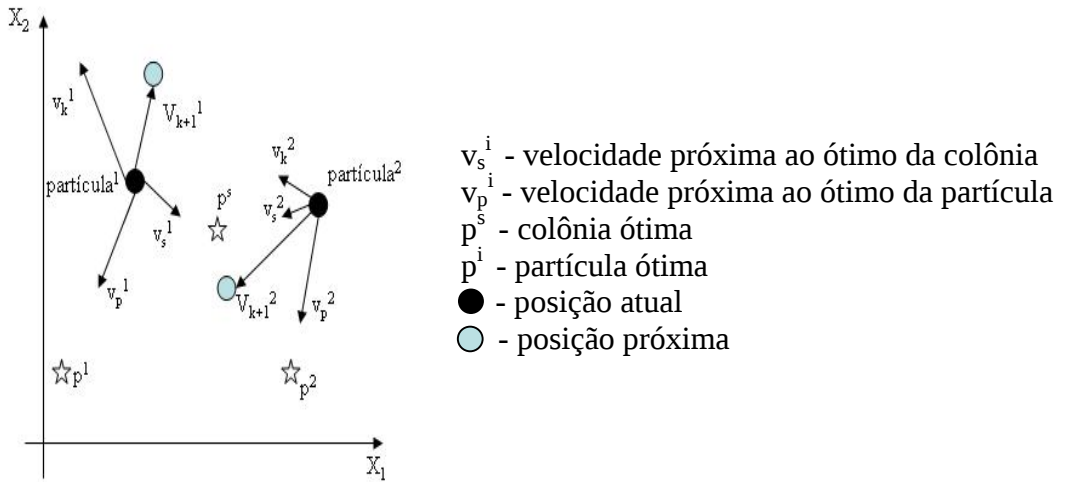


Figura 2: Vetor de velocidades em ação.

Os parâmetros de confiança e de inércia devem ser ajustados de acordo com o problema, pois são utilizados para a atualização do vetor velocidade. Alguns autores propõem que sejam adotados $c_1 = c_2 = 2$ e $0,7 < w < 1,4$. Sugere-se, também, a adoção de valores diferentes para c_1 e c_2 desde que satisfaçam $c_1 + c_2 = 4$ (Eberhart and Kennedy, 1995). A inércia pode ser atualizada de forma iterativa pela expressão:

$$w_{new} = f_w w_{old} \quad (10)$$

considerando o fator de redução, f_w uma constante entre 0 e 1.

4. APLICAÇÃO NUMÉRICA

Uma característica fundamental a ser considerada no projeto dimensional de robôs manipuladores é o seu volume do espaço de trabalho.

Neste artigo, deseja-se estudar os robôs manipuladores com três juntas rotacionais (3R), cujo esquema cinemático está representado na Figura 3. O efetuator é montado no final da terceira junta.

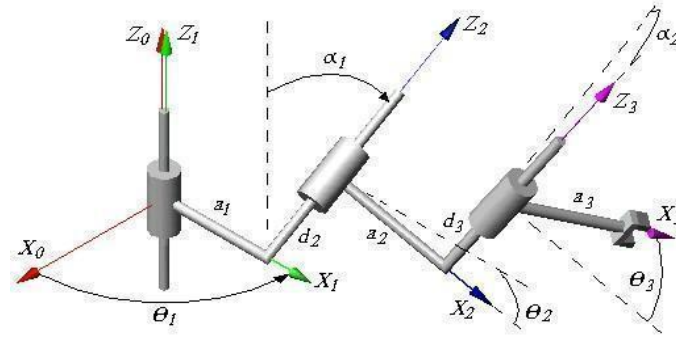


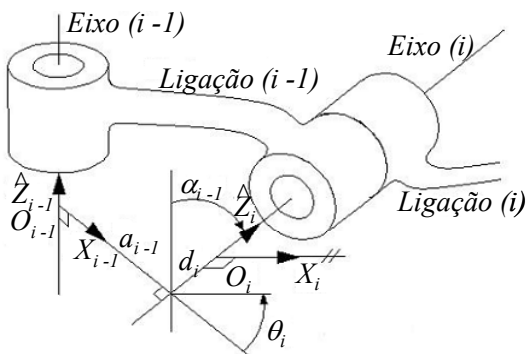
Figura 3: Esquema cinemático de um robô manipulador 3R e seus parâmetros cinemáticos.

A capacidade de um robô desenvolver uma determinada tarefa depende da sua arquitetura e da dimensão de suas ligações (*links*), assim como da posição por ele assumida no ambiente de trabalho. Estas características devem ser consideradas no projeto dos manipuladores robóticos, ou melhor, na definição de sua geometria. No caso de manipuladores com juntas puramente rotacionais, os parâmetros de projeto são $a_1, a_2, a_3, d_2, d_3, \alpha_1$ e α_2 ; os termos θ_1, θ_2 e θ_3 representam as variáveis cinemáticas (representados na Figura 3).

O cálculo exato do espaço de trabalho é muito importante, pois influi no projeto dimensional, no posicionamento do manipulador no ambiente de trabalho e na sua destreza para executar tarefas. Neste estudo, a determinação do volume do espaço de trabalho necessita do conhecimento da área de sua seção transversal em um plano radial, a qual é obtida por aproximação através de sua discretização em uma malha retangular. O problema de otimização é definido por uma função objetivo que maximiza o volume do espaço de trabalho. Foram incluídas restrições laterais para obter as dimensões do manipulador dentro de valores práticos, considerando restrições geométricas.

4.1. Cinemática de robôs manipuladores utilizando o método de Denavit-Hartenberg

Um dos métodos mais usados para descrever geometricamente um robô é utilizar a notação de Denavit e Hartenberg, cujo esquema é exibido na Figura 4. Esta notação basicamente consiste em construir a matriz de transformação homogênea genérica, T_{i-1}^i , que representa o sistema $X_i Y_i Z_i$, associado a i -ésima ligação do robô, em relação ao sistema $X_{i-1} Y_{i-1} Z_{i-1}$, associado a $(i-1)$ -ésima ligação, conforme Equação 11, para i variando de 1 a m , onde m é o grau de liberdade do robô



$$T_{i-1}^i = \text{Rot}(\alpha_i, X_{i-1}) \text{Trans}(a_{i-1}, 0, 0) \text{Trans}(0, 0, d_i) \text{Rot}(\theta_i, Z_i),$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & S\alpha_{i-1} & d_i S\alpha_{i-1} \\ -S\theta_i S\alpha_{i-1} & -C\theta_i S\alpha_{i-1} & C\alpha_{i-1} & d_i C\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(11)

Figura 4: Notação de Denavit e Hartenberg para manipuladores.

A matriz de transformação homogênea que representa as coordenadas do efetuador em relação à base, T_0^m , pode ser obtida através da seguinte expressão (Saramago and Steffen Jr, 1999):

$$T_0^m = T_0^1 \cdot T_1^2 \cdot T_2^3 \cdot \dots \cdot T_{m-1}^m \quad (12)$$

Seja um ponto H escolhido como o centro do efetuador, este ponto será denotado por H_3 e representado por:

$$H_3 = (a_3 \quad 0 \quad 0 \quad 1)^T \quad (13)$$

Utilizando a Equação 12, a representação vetorial de H_3 em relação à base, denotada por H_0 , é obtida da seguinte forma:

$$H_0 = T_0^{-1} T_1^{-2} T_2^{-3} H_3 = T_0^{-1} H_1 \quad (14)$$

4.2. Espaço de Trabalho de Manipuladores 3R

O espaço de trabalho $W(H)$ de um ponto H situado na extremidade do robô manipulador é o conjunto de todos os pontos que H ocupa quando as variáveis de junta são variadas em todos os seus intervalos de definição (Gupta and Roth, 1982).

O procedimento mais imediato para investigar o espaço de trabalho é variar os ângulos θ_1 , θ_2 e θ_3 sobre seus intervalos de definição e estimar as coordenadas do ponto H com relação à base do manipulador. Desta maneira, obtém-se a posição do órgão terminal, tendo como resultado a representação vetorial dada por H_0 .

A Figura 5 exibe o espaço de trabalho seccionado de um determinado manipulador 3R. Como pode ser visto, o espaço de trabalho deste tipo de robô é um sólido de revolução, tendo Z como o seu eixo de revolução. Desta forma, é natural imaginar que o espaço de trabalho é o resultado da rotação, em torno do eixo Z , de uma seção radial plana que funcione como uma seção geratriz.

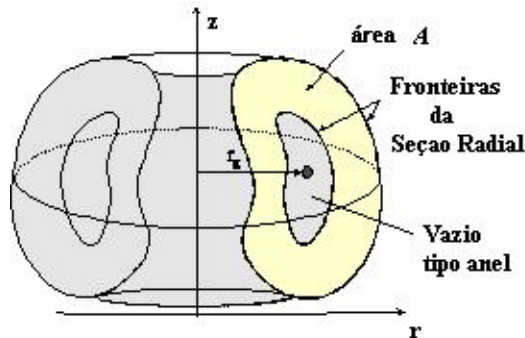


Figura 5: Espaço de trabalho de um manipulador 3R.

O espaço de trabalho de robôs com estrutura 3R pode ser obtido por intermédio da extensão radial r e da extensão axial z com relação à base (Ceccarelli, 1996). Assim, usando a Equação 14, as equações paramétricas do lugar geométrico descrito pelo ponto H sobre um plano radial são:

$$\begin{cases} r^2 = (H_1^x)^2 + (H_1^y)^2 \\ z = H_1^z \end{cases} \quad (15)$$

As expressões H_1^x , H_1^y e H_1^z são encontrados em Oliveira (2006). O volume do espaço de trabalho, V , é o volume do sólido de revolução obtido pela rotação da seção radial em torno do eixo Z_1 (Figura 3). Assim, usando o Teorema de Pappus-Guldin, o volume é dado através da equação:

$$V = 2 \pi r_g A, \quad (16)$$

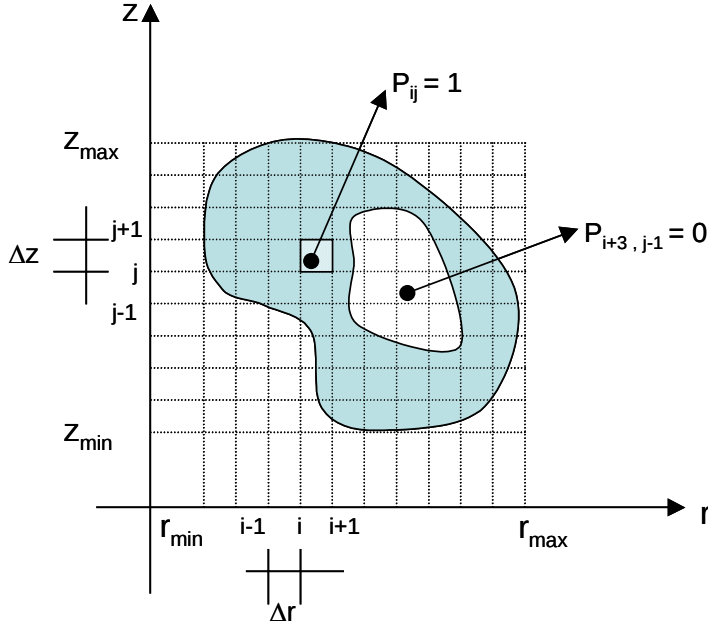
onde A é a área da seção radial plana que é coberta pela família de curvas.

Esta pesquisa propõe uma formulação numérica para aproximar o cálculo da área da seção radial, através de sua discretização em uma malha retangular. Inicialmente, deve-se obter os valores

extremos dos vetores r e z . Adotando-se o número de subintervalos desejados para a discretização ao longo de r e z (n_r e n_z), pode-se calcular as dimensões das áreas elementares Δr e Δz da malha.

A Equação 15 permite calcular todos os pontos da família de curvas que compõem a seção radial do espaço de trabalho. Dado um determinado ponto (r, z) , determina-se sua posição dentro da malha de discretização, conforme Figura 6, através do controle de índices dados nas Equações 17 e 18.

Conforme mostrado na Figura 6, o ponto da malha que pertence ao espaço de trabalho será identificado como $P_{ij} = 1$, do contrário terá valor nulo, conforme Equação 19.



$$r_{\min} = \min \{r\} \text{ e } r_{\max} = \max \{r\}$$

$$z_{\min} = \min \{z\} \text{ e } z_{\max} = \max \{z\}$$

$$\Delta r = \frac{r_{\max} - r_{\min}}{n_r} \text{ e } \Delta z = \frac{z_{\max} - z_{\min}}{n_z}$$

$$i = \text{int} \left(\frac{r - r_{\min}}{\Delta r} \right) + 1 \quad (17)$$

$$j = \text{int} \left(\frac{z - z_{\min}}{\Delta z} \right) + 1 \quad (18)$$

$$P_{ij} = \begin{cases} 0 & \text{se } P_{ij} \notin W(H) \\ 1 & \text{se } P_{ij} \in W(H) \end{cases} \quad (19)$$

Figura 6 - Discretização da seção radial usando malha retangular.

Desta forma a área total será obtida pela soma de todas as áreas elementares da malha que estão contidas, totalmente ou parcialmente, na seção radial. Na Equação 20, observa-se que apenas os pontos pertencentes ao espaço de trabalho contribuem para o cálculo da área. A coordenada do baricentro é calculada considerando a soma dos baricentros de cada área elemental, dividida pela área total:

$$A = \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta r \Delta z); \quad r_g = \frac{\sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (P_{ij} \Delta r \Delta z) \left((i-1) \Delta r + \frac{\Delta r}{2} + r_{\min} \right)}{A} \quad (20)$$

Finalmente, conhecendo-se os valores da área e do baricentro da seção radial, pode-se calcular o volume do espaço de trabalho do manipulador usando a Equação 16.

4.3. Formulação do Problema Ótimo

O problema é formulado visando maximizar o volume do espaço de trabalho, onde o vetor das variáveis de projeto é formado pelas dimensões do manipulador, conforme representado na Figura 3, sendo dado por $X = (a_1, a_2, a_3, d_2, d_3, \alpha_1, \alpha_2)^T$, $x_i^{\inf}$ e $x_i^{\sup}$ representam os limites inferiores e superiores de cada variável, respectivamente e $L_{\max}$ é uma restrição geométrica imposta pelo alcance máximo. O volume V é obtido usando as Equações 16 a 20, definido como:

$$\begin{aligned} &\text{Maximizar } f(X) = V \\ &\text{sujeito a } x_i^{\inf} \leq x_i \leq x_i^{\sup}, i = 1, \dots, n; \quad z > 0 \quad \text{e} \end{aligned} \quad (21)$$

$$r < L_{\max} = \sqrt{(a_1^{\sup})^2 + (d_2^{\sup})^2} + \sqrt{(a_2^{\sup})^2 + (d_3^{\sup})^2} + (a_3^{\sup})$$

5. RESULTADOS

A maximização do volume do espaço de trabalho foi desenvolvida segundo o problema de otimização definido na Equação 21, a Tabela 1 mostra os resultados obtidos. O espaço de busca foi limitado em: $0 \leq a_i \leq 1$ [u.m.], $i=1, 2,3$; $0 \leq d_j \leq 1$ [u.m.], $j=2,3$ e $0^\circ \leq \alpha_k \leq 90^\circ$, $k=1,2$. A variação dos ângulos θ_2 e θ_3 para a discretização da malha retangular foi $\Delta\theta_2 = \Delta\theta_3 = 0,025$ rd e $n_r = n_z = 200$ subintervalos. O número máximo de iterações adotado foi de 100 iterações em todas as técnicas.

Os parâmetros utilizados na ED foram: $Np=15$, $F=0,8$ e $Pc=0,6$. Os AGs foram aplicados usando o código computacional GAOT (Houck, 1995), adotando-se uma população com 30 indivíduos. Os parâmetros utilizados no PSO foram: número de partículas igual a 30, $c_1 = c_2 = 2$ e $w = 0,729$.

Foram efetuadas 10 execuções independentes para cada técnica apresentada. Nas Tabelas 1 e 2 observa-se que o volume ótimo foi semelhante para as três técnicas estudadas. É importante observar que a Evolução Diferencial apresentou uma significativa redução do tempo computacional, sendo menor que a metade do tempo exigido pelos Algoritmos Genéticos. Este resultado era esperado uma vez que ED necessita de um pequeno número de indivíduos.

Tabela 1: Volume máximo do espaço de trabalho de um robô 3R obtido em 10 experimentos.

Variáveis de Projeto	AG	PSO	ED
a_1 [u.m.]	0,8176	0,4389	0,4606
a_2 [u.m.]	0,5968	0,9104	0,9041
a_3 [u.m.]	0,8387	0,9873	0,9881
d_2 [u.m.]	0,9988	1,0000	1,0000
d_3 [u.m.]	1,0000	0,9970	0,9980
α_1 [graus]	26,59	17,70	16,45
α_2 [graus]	61,12	63,69	66,11
Volume Máximo [u.v.]	37,23	39,29	39,27
Tempo comput. [min]	170,82	76,30	66,72

Tabela 2: Comparativo entre os algoritmos – volumes obtidos em 10 experimentos.

	Mínimo	Máximo	média	Desvio Padrão
AG	30,48	37,23	33,737	2,145
PSO	30,13	39,29	36,557	3,419
ED	35,90	39,27	37,924	1,383

A seção transversal do espaço de trabalho ótimo é apresentada na Figura 7, sendo sua forma dependente essencialmente dos ângulos α_1 e α_2 , uma vez que as variáveis dimensionais a_i e d_j convergem para os limites máximos permitidos.

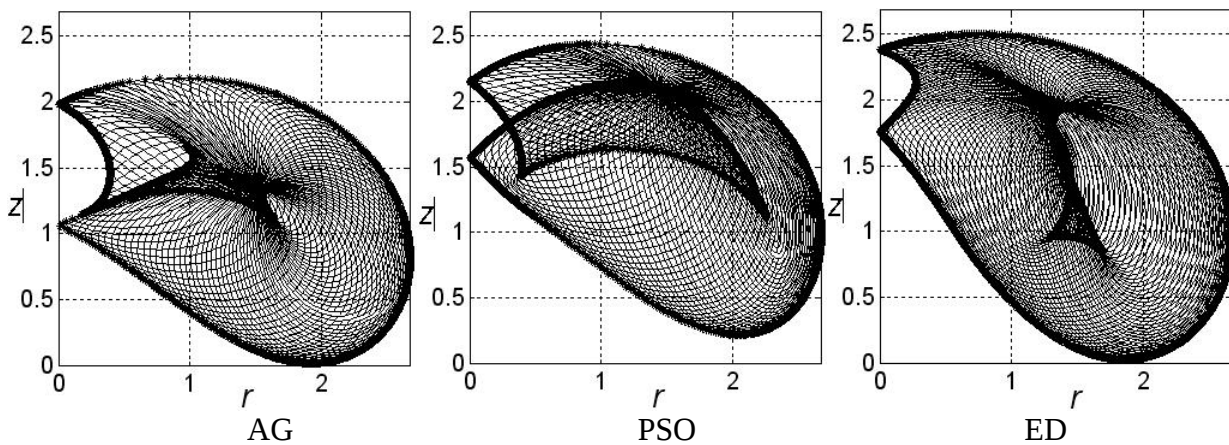


Figura 7: Espaço de trabalho ótimo de um manipulador 3R de acordo com as técnicas utilizadas.

6. CONCLUSÕES

Neste trabalho, um problema de otimização foi definido com o propósito de maximizar o volume do espaço de trabalho de um robô com três juntas rotacionais, mantendo suas dimensões dentro de uma faixa limitada, considerando uma restrição geométrica imposta pelo alcance máximo. Foram utilizadas três técnicas evolutivas e todas apresentaram boa performance na obtenção da solução. Para o caso em estudo, os Algoritmos Genéticos apresentaram esforço computacional bem superior às demais técnicas. A Evolução Diferencial reduziu de forma significativa o tempo de execução, quando comparada com os outros métodos, mostrando ser uma técnica de grande potencial para ser aplicada a problemas complexos.

6. AGRADECIMENTOS

A primeira autora agradece o suporte do CNPq e a terceira autora agradece a Fapemig (processo TEC 635/05).

7. REFERÊNCIAS

- Bregalda, P. et al., 1988, "Introdução à Programação Linear", Ed. Campus.
- Ceccarelli, M., 1996, "A formulation for the workspace boundary of general N-revolute manipulators", IFToMM Jnl Mechanism and Machine Theory, Vol. 31, n. 5, pp 637-646.
- Deb, K., 2001, "Multi-objective optimization using Evolutionary Algorithms", John Wiley&Sons.
- Dorigo, M., Maniezzo, V., and Colorni, A., 1991, "Positive Feedback as a Search Strategy", Technical Report 91-016, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.
- Eberhart, R. C. and Kennedy, J., 1995, "A New Optimizer Using Particle Swarm Theory", Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, Piscataway, NJ: IEEE Service Center, pp 39-43.
- Goldberg, D. E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning", Reading, MA: Addison-Wesley.
- Gupta, K. C. and Roth, B., 1982, "Design Considerations for Manipulator Workspace", ASME, Journal of Mechanical Design. Vol. 104, pp 704-711,.
- Haupt, R. L. and Haupt, S. E., 1998, "Practical Genetic Algorithms", Wiley-Interscience Publication, New York.
- Houck, C. R., Joines, J. A., and Kay, M. G., 1995, "A Genetic Algorithm for Function Optimization: a Matlab Implementation", NCSU-IE Technical Report, University of North Carolina, USA.

- Kennedy, J., Eberhart, R. C., and Shi, Y., 2001, "Swarm Intelligence", San Francisco: Morgan Kaufmann Publishers.
- Oliveira, G. T. S., 2006, "Estudo e aplicações da Evolução Diferencial". Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia, MG, Brasil, 126p.
- Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M., 1983, "Engineering Optimization: Methods and Applications", John Wiley & Sons, USA.
- Saramago, S.F.P., and Steffen Jr, V., 1999, "Dynamic Optimization for the Trajectory Planning of Robot Manipulators in the Presence of Obstacles", Journal of the Brazilian Society of Mechanical Sciences, Brazil, Vol. XXI, n. 3, pp 371-383.
- Schwefel, H.P., 1995, "Evolution and Optimum Seeking", John Wiley & Sons.
- Storn, R., and Price, K., 1995, "Differential Evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley.
- Vanderplaats, G. N., 1999, "Numerical Optimization Techniques for Engineering Design", Vanderplaats Research & Development, Inc., Colorado Springs, CO, 3rd ed.

AN OPTIMAL DESIGN OF 3R MANIPULATOR BY USING EVOLUTIONARY ALGORITHMS

Giovana Trindade da Silva Oliveira

Federal University of Uberlândia, School of Mechanical Engineering
2160 João Naves de Ávila Av., Campus Santa Mônica, CEP 38400-902, Uberlândia, Brazil
gtrindade@mecanica.ufu.br

Sezimária Fátima Pereira Saramago

Federal University of Uberlândia, School of Mathematics
2160 João Naves de Ávila Av., Campus Santa Mônica, CEP 38400-902, Uberlândia, Brazil
saramago@ufu.br

Paulo Roberto Bergamaschi

Federal University of Goiás, School of Mathematics
1120 Dr. Lamartine Pinto de Avelar Av., Campus Catalão, CEP 75701-220, Catalão, Brazil
paulob@innet.psi.br

Abstract: *In this paper, the optimal project of manipulators 3R is defined considering the volume of maximum work, keeping the dimensions of the robot inside of a limited band and imposing geometric constraints, what it means reduction of the costs of manufacture of the equipment and greater band of operational application. The workspace is the region defined for the set of possible points of the space to be reached for the end effector. In the case of manipulators 3R, is represented by a revolution solid. The study of workspace boundary is of basic importance for the development project of a manipulator, therefore it represents the border of its workspace. Its form depends on the architecture and of the dimensions of the robot and such parameters must be determined so that the maximum volume is gotten. As optimization methods, it is considered use of some evolution algorithms, such as genetic algorithms, differential evolution and particle swarm optimization. Are presented numerical simulations for validation of the proposal methodology and comparison of the used techniques.*

Keywords: *Robotic, Optimization, workspace, constraints evolution algorithms*