

ESTRATÉGIAS HEURÍSTICAS APLICADAS AO PLANEJAMENTO ÓTIMO DA TRAJETÓRIA DE ROBÔS

Rogério Rodrigues dos Santos

Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia
Av. João Naves de Ávila, 2160 - Campus Santa Mônica - Bloco 1M - Uberlândia/MG
rrsantos@mecanica.ufu.br

Valder Steffen, Jr.

Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia
vsteffen@mecanica.ufu.br

Resumo: Robôs manipuladores são dispositivos programáveis projetados para executar uma grande variedade de tarefas de forma repetitiva. Em ambientes industriais, pode-se conseguir o aumento de produtividade e diminuição dos custos associados à operação e manutenção do robô a partir da diminuição de exigências dinâmicas associadas à realização de uma mesma tarefa. Adicionalmente, em alguns casos é viabilizada a realização de tarefas que exijam a capacidade máxima do sistema a partir da redução das exigências dinâmicas do movimento. Neste trabalho, apresenta-se uma proposta de redução do torque total requerido pelo movimento, a partir de conceitos de otimização global e da caracterização da dinâmica do robô manipulador serial. Propõe-se uma metodologia para ajuste da trajetória pelo uso de nós intermediários de interpolação das coordenadas de junta. Resultados numéricos ilustram o desempenho das diferentes estratégias de otimização e confirmam a viabilidade da metodologia proposta.

Palavras-chave: planejamento de trajetórias, otimização global, robôs manipuladores.

1. INTRODUÇÃO

Dentre os objetivos mais importantes da automação industrial, está a redução do custo de produção e o aumento da produtividade. Conseqüentemente, para viabilizar o uso de sistemas robóticos, é de grande importância que se considere o planejamento e otimização da trajetória a ser executada.

Com o objetivo de estabelecer um controle mais preciso sobre o movimento, é justificada a realização de um planejamento prévio (*off-line programming*) que inclui a análise de características cinemáticas e dinâmicas do sistema durante o movimento.

O planejamento do movimento de robôs manipuladores tem sido uma área de pesquisa ativa nas últimas décadas (Hwang e Ahuja, 1992). A solução proposta para este problema, freqüentemente chamada de planejamento de percurso (*path planning*) ou desvio de obstáculos (*collision avoidance / obstacle avoidance*), pode ser categorizada de forma geral em duas classes: os métodos globais e os métodos locais (Seraji e Bon, 1999). Os métodos globais freqüentemente operam no espaço de coordenadas de junta, onde a configuração do manipulador é representada por um ponto. Os obstáculos são mapeados do espaço cartesiano e no espaço de coordenadas de junta, e o percurso é então determinado na porção desocupada do espaço de coordenadas de junta, a partir do ponto inicial até ao ponto final, de forma a desviar do obstáculo.

Os métodos globais são computacionalmente caros, e o custo computacional cresce rapidamente em função do aumento do número de juntas do manipulador.

Os métodos de planejamento de trajetória local (*local path planning*) (Khatib, 1986) requerem menor demanda computacional, e podem considerar a existência de obstáculos móveis em ambientes não estruturados. Desta forma, são apropriados também para a determinação de trajetórias a partir de informações geradas por sensores.

Em relação à consideração de ambiente irrestrito, o trabalho de Santos *et al.*, 2005c apresenta uma proposta de otimização do torque através de técnicas de controle ótimo. Uma proposta de otimização do torque e potência mecânica através de uma formulação multicritério, resolvida através de controle ótimo é apresentada em Santos *et al.*, 2006a.

Considerando a presença de obstáculos fixos no espaço de trabalho, Santos *et al.*, 2005b apresenta uma proposta de otimização de trajetória que leva em conta o tempo total de percurso, posicionamento do efetuador e potência mecânica, sendo a trajetória obtida a partir de interpolação tipo *spline*. Uma proposta de otimização da potência mecânica do manipulador através do uso de controle ótimo, onde é considerada a presença de obstáculos móveis em ambiente bidimensional é discutida em Santos *et al.*, 2006b. O trabalho de Santos *et al.*, 2006c traz uma proposta para a determinação e otimização de trajetória de robôs capaz de considerar a interceptação de objetos em movimento.

Neste trabalho apresenta-se um estudo sobre o ajuste da trajetória intermediária de um robô manipulador de forma a diminuir o torque total requerido durante o percurso. A análise é realizada pela determinação de torques adequados à dinâmica do sistema, considerado através da formulação de um problema de otimização e resolvido por diferentes estratégias.

A contribuição atual consiste na avaliação de diferentes estratégias heurísticas para a determinação do mínimo global, respeitando-se os limites mínimos e máximos estabelecidos para os ângulos de junta. Consideram-se os valores ótimos obtidos pelo índice de desempenho, o número de chamadas à função objetivo e o tempo computacional requerido como informações relevantes de cada estratégia. Comparam-se estes quesitos com aqueles requeridos por uma estratégia de programação não linear clássica.

O trabalho tem a seguinte estrutura: o modelo dinâmico do robô manipulador e a função objetivo que calcula o torque total requerido são apresentados na seção 2. A seção 3 discute a geração da trajetória através da interpolação e o uso de nós intermediários como variáveis de projeto. Diferentes estratégias de otimização heurística são apresentadas na seção 4. A seção 5 apresenta os experimentos numéricos, enquanto a conclusão do trabalho é discutida na seção 6.

2. MODELO DINÂMICO

A dinâmica reúne as relações entre a cinemática e a estática. Para se mover um robô ao longo de uma trajetória, os motores devem exercer forças ou torques nas juntas do mesmo. A dinâmica trata do cálculo destas forças e torques (conhecidos indistintamente como forças generalizadas), no sentido de fazer com que uma trajetória planejada seja executada, de fato, pelo efetuador do robô.

Existem várias técnicas para se modelar dinamicamente um robô industrial. O conhecimento de seu modelo dinâmico é fundamental (Fu *et al.*, 1987) para a simulação computacional do movimento do robô e para o projeto de seu controle.

2.2. Modelo Dinâmico do Manipulador Planar

A seguir, considera-se as características do robô manipulador planar com 2 graus de liberdade, que tem os parâmetros de Denavit-Hartenberg apresentados na Tabela 1.

Tabela 1: Parâmetros de Denavit-Hartenberg, (*) variável de junta.

Link	a (m)	α (rad)	d (m)	θ (rad)
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

Desenvolvendo-se a equação de Euler-Lagrange e substituindo a expressão do Lagrangeano obtêm-se explicitamente as equações que representam a dinâmica do sistema, que podem ser representadas compactamente (Fu *et al.*, 1987) por

$$Q(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) + G(\mathbf{q}) = \mathbf{F} \quad (1)$$

onde $\mathbf{q} = (\theta_1, \theta_2, \dots, \theta_n)^T$, $\dot{\mathbf{q}} = (\dot{\theta}_1, \dot{\theta}_2, \dots, \dot{\theta}_n)^T$ e $\ddot{\mathbf{q}} = (\ddot{\theta}_1, \ddot{\theta}_2, \dots, \ddot{\theta}_n)^T$ são vetores $n \times 1$ de posição, velocidade e aceleração respectivamente, das variáveis de junta. $Q(\mathbf{q})$ é uma matriz $n \times n$ simétrica que representa a inércia, $C(\mathbf{q}, \dot{\mathbf{q}})$ é um vetor $n \times 1$ que representa o efeito de Coriolis e a força centrífuga, $G(\mathbf{q})$ é um vetor $n \times 1$ que representa os efeitos da aceleração da gravidade e $\mathbf{F} = (F_1, F_2, \dots, F_n)^T$ é um vetor $n \times 1$ das forças generalizadas nas juntas.

Para o robô manipulador planar com dois graus de liberdade tem-se $n=2$ e os respectivos elementos das matrizes Q , C e G dados abaixo:

$$Q_{11} = \left(\frac{1}{3}\right) m_1 a_1^2 + \left(\frac{1}{3}\right) m_2 a_2^2 + m_2 a_1^2 + m_2 a_1 a_2 \cos(\theta_2) \quad (2)$$

$$Q_{12} = \left(\frac{1}{3}\right) m_2 a_2^2 + 0.5 m_2 a_1 a_2 \cos(\theta_2) \quad (3)$$

$$Q_{21} = \left(\frac{1}{3}\right) m_2 a_2^2 + 0.5 m_2 a_1 a_2 \cos(\theta_2) \quad (4)$$

$$Q_{22} = \left(\frac{1}{3}\right) m_2 a_2^2 \quad (5)$$

$$C_{11} = -m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2 - 0.5 m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_2^2 \quad (6)$$

$$C_{21} = 0.5 m_2 a_1 a_2 \sin(\theta_2) \dot{\theta}_2^2 \quad (7)$$

$$G_{11} = -0.5 m_1 g a_1 \cos(\theta_1) - 0.5 m_2 g a_2 \cos(\theta_1 + \theta_2) - m_2 g a_1 \cos(\theta_1) \quad (8)$$

$$G_{21} = -0.5 m_2 g a_2 \cos(\theta_1 + \theta_2) \quad (9)$$

onde $g = 9.81 \text{ m/s}^2$ representa a constante gravitacional. Desta forma, as Equações (2)-(9) definem completamente o comportamento dinâmico do sistema.

Devido à relação existente entre energia e força, pode-se estimar a energia necessária para a realização do movimento através da força generalizada $F_j(t)$ associada a cada junta j no instante t . Neste trabalho considera-se o uso da expressão

$$f = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^n |F_j(i)| \quad (10)$$

para o cálculo do torque total requerido pelo sistema. A variável $j=1, \dots, n$ representa o número de juntas do robô, enquanto i consiste na subdivisão do conjunto que representa o tempo total $[t_0, t_f]$ em N passos, de tal forma que $i_0=t_0$ e $i_N=t_f$.

3. TRAJETÓRIA

A técnica mais comumente utilizada para determinação da transição entre as posições iniciais e finais desejadas (ou seja, o cálculo efetivo do movimento) consiste em realizar uma interpolação entre estes dois pontos, a partir das respectivas coordenadas de junta.

Pode-se realizar uma interpolação polinomial simples, ou utilizar técnicas mais sofisticadas, como a interpolação através de funções do tipo *spline*. As diferentes técnicas de interpolação permitem a análise não só do posicionamento e orientação do robô, mas também oferecem informações sobre velocidade, aceleração, torque e energia requeridos para a realização da tarefa (Saramago and Steffen Jr, 1998).

A trajetória é definida de acordo com o histórico de posições, velocidades e acelerações de cada junta, como funções que dependem do tempo.

Dado um manipulador com n graus de liberdade, inicialmente devem-se estabelecer os respectivos ângulos de junta, θ_j , $j=1 \dots n$, relativos à configuração inicial e final. Então se define uma função de interpolação, $P_j(t)$, para cada coordenada de junta j , de tal forma que as especificações de posição sejam satisfeitas. Este procedimento requer a especificação do tempo inicial e final, t_0 e t_f , os vetores de posição inicial e final, $\theta_j(t_0)$ e $\theta_j(t_f)$, e as velocidades inicial e final, $\dot{\theta}_j(t_0)$ e $\dot{\theta}_j(t_f)$, respectivamente. As restrições sobre os valores das coordenadas de junta são expressas por

$$P_j(t_0) = \theta_j(t_0) \text{ e } P_j(t_f) = \theta_j(t_f), \quad (11)$$

e as especificações para a velocidade são dadas por

$$\frac{\partial P_j}{\partial \theta_j}(t_0) = \dot{\theta}_j(t_0) \text{ e } \frac{\partial P_j}{\partial \theta_j}(t_f) = \dot{\theta}_j(t_f). \quad (12)$$

A trajetória assim obtida, através de interpolação *spline* por exemplo, satisfaz os requisitos de suavidade. Entretanto, ela não representa necessariamente a configuração ótima do sistema, de acordo com outros critérios de desempenho. Por este motivo, são definidos $k=1, \dots, 5$ nós intermediários de interpolação p_{kj} , para cada junta $j=1, 2$. A ordenada destes nós intermediários são as variáveis de projeto do processo de otimização. O significado geométrico desta informação é representado por asterisco (*) na Figura 1.

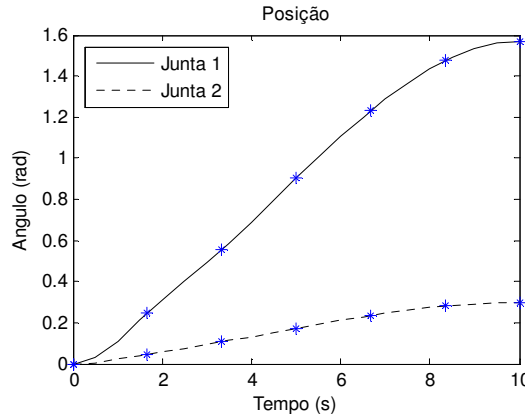


Figura 1: Nós intermediários p_{kj} (*) utilizados como variáveis de projeto.

Observa-se que os nós dos instantes iniciais ($t_0=0$) e finais ($t_f=10$ s) são fixos, pois representam a posição inicial e final do robô.

4. OTIMIZAÇÃO

Foram consideradas diferentes estratégias de busca global heurística para a minimização da Equação (10).

4.1 Algoritmos genéticos

A primeira estratégia considerada é referente ao Algoritmo Genético (Vose, 1999) (Schmitt, 2001) (Schmitt, 2004). Esta estratégia é inspirada por elementos de biologia evolucionária, tais como herança, mutação, seleção e cruzamento.

A classe de algoritmos genéticos consistem em uma simulação computacional onde a população em questão (também denominada cromossomo) de soluções candidatas (chamada de

indivíduos) para o problema de otimização evoluem para uma solução melhor. A solução usualmente tem representação binária, entretanto outras codificações também são possíveis.

A evolução é comumente iniciada a partir de uma população de indivíduos gerados aleatoriamente. A cada iteração o valor dos indivíduos é avaliado na função objetivo, estabelecendo-se assim seu respectivo *fitness*. Alguns indivíduos da população são selecionados e modificados (pelos operadores de cruzamento e mutação) a fim de formar uma nova população.

Esta nova população é então utilizada na a próxima iteração do algoritmo, que repete este processo um número finito de vezes. Podem-se estabelecer critérios de parada alternativos, tais como tempo máximo de processamento, alcance de um valor objetivo específico, ou a ausência de melhoria por um dado número de gerações.

Uma codificação típica desta estratégia requer a definição da representação genética do domínio das variáveis de projeto e da função objetivo para avaliar a qualidade de cada solução (*fitness*).

A função objetivo é definida a respeito da representação genética, mede a qualidade da solução provida pelo indivíduo, e é específica para cada problema analisado.

No estudo atual esta função representa a soma do valor do torque requerido em cada junta do robô para realizar o movimento especificado, conforme descrito pela Equação (10).

Cada indivíduo da população representa um conjunto de combinações diferentes para o posicionamento dos nós intermediários da interpolação, como apresentado na Figura 1. Assim, diferentes indivíduos correspondem a diferentes polinômios de interpolação.

4.2 Busca de padrões

Considera-se o uso do método *Generalized Pattern Search* (Lewis *et al.*, 2002) (Conn *et al.*, 1991) (Conn *et al.*, 1997) como alternativa para a resolução do problema de otimização proposto.

Esta estratégia calcula uma seqüência de pontos que se aproximam da configuração ótima. A cada passo, o algoritmo procura um conjunto de pontos, chamados de malha (*mesh*), próximos ao ponto atual. A malha é formada pelo acréscimo do ponto atual multiplicado por um conjunto de vetores denominado padrão (*pattern*).

Quando o padrão de pesquisa do algoritmo encontra um ponto na malha que melhora o valor da função objetivo (minimiza ou maximiza o valor de acordo com o objetivo considerado), este ponto se torna a referência atual para a próxima iteração do algoritmo. Podem-se utilizar vetores de direção fixos ou variáveis, sendo que esta escolha diferencia o algoritmo *Generalized Pattern Search* (GPS) do algoritmo *Mesh Adaptive Search* (MADS).

4.3 Recozimento simulado

Adicionalmente, considera-se a estratégia de recozimento simulado (*Simulated Annealing*) (Kirkpatrick *et al.*, 1983) (Ingber, 1993) para otimização da trajetória do robô.

Este método é inspirado no fenômeno físico de aquecimento de um material, e então lentamente diminui a temperatura do sistema para encontrar uma configuração de equilíbrio, através da minimização da energia do sistema. A cada iteração, um novo ponto é gerado randomicamente. A distância entre o ponto atual e o novo ponto é baseada em uma distribuição de probabilidade na qual a escala proporcional a temperatura.

Novos pontos que minimizem o valor do objetivo são aceitos. Entretanto, certos pontos que aumentem o valor da função objetivo também são aceitos com certa probabilidade. Pela aceitação destes pontos, o algoritmo evita a análise exclusivamente em torno de um mínimo local, estando, portanto, apto a uma exploração global em busca da solução ótima.

O método conta ainda com uma estratégia de recozimento para obter um decréscimo sistemático da temperatura durante a execução do algoritmo.

Considerou-se também uma variação esta estratégia, chamada *Threshold Acceptance* (Dueck and Scheuer, 1990). Ela é semelhante à estratégia anterior. Entretanto, neste caso são aceitos todos

os candidatos a configuração ótima que estão abaixo de um limiar fixado. Este limiar é diminuído conforme a temperatura decresce. Devido ao fato desta estratégia não requerer a aceitação probabilística característica do *Simulated Annealing*, usualmente localiza uma configuração ótima mais rapidamente.

4.4 Programação quadrática seqüencial

Finalmente, considera-se também o uso de um método clássico de programação não linear para a determinação da configuração ótima. Assim, faz-se uso da programação quadrática seqüencial (*Sequential Quadratic Programming*).

Este método difere dos anteriores por requerer o cálculo do gradiente da função objetivo a cada iteração. Por este motivo, ele deve ser aplicado a problemas onde haja continuidade da função objetivo e de suas respectivas derivadas.

A cada iteração, realiza-se uma aproximação da matriz Hessiana da função Lagrangeana (que inclui as restrições) através de uma estratégia de aproximação quase-Newton. Esta informação gera um subproblema de programação quadrática, cuja solução é utilizada para determinar a direção de busca do mínimo. Detalhes sobre este algoritmo são apresentados por Fletcher (1987), Gill *et al.* (1981) e Powell (1983), dentre outros.

5. EXPERIMENTOS NUMÉRICOS

Considera-se nesta seção o uso do manipulador planar, cuja dinâmica é dada pelas Equações (1)-(9). Os links têm massa $(m_1, m_2) = (1.7, 1.3) \text{ kg}$ e comprimento $(a_1, a_2) = (1.2, 0.9) \text{ m}$, respectivamente.

Para avaliar o desempenho de cada estratégia na obtenção da trajetória ótima, é considerado o movimento de um robô que se movimenta da posição inicial $(\theta_1, \theta_2) = (0,0) \text{ rad}$ até a posição final $(\theta_1, \theta_2) = (1.57, 0.3) \text{ rad}$. O movimento deve ser realizado no tempo $t_f = 10 \text{ s}$.

Os experimentos numéricos realizados incluem a configuração gerada a partir da interpolação *spline* cúbica entre a coordenada inicial e final de cada junta como um indivíduo da população inicial. Desta forma, pretende-se utilizar a otimização para a melhoria do resultado oferecido pela interpolação cúbica.

A trajetória inicial tem como nós intermediários de interpolação as ordenadas $\mathbf{p}_1 = [0.2454 \ 0.5529 \ 0.9028 \ 1.2315 \ 1.4754]$ e $\mathbf{p}_2 = [0.0469 \ 0.1056 \ 0.1724 \ 0.2352 \ 0.2818]$ para a primeira e segunda junta, respectivamente. A abscissa, comum a todas as juntas, é dada por $\mathbf{i} = [1.67 \ 3.33 \ 5.00 \ 6.67 \ 8.33]$. Adicionalmente, consideram-se restrições sobre o ângulo mínimo e máximo admissíveis para cada junta durante a minimização da Equação (10). A primeira junta tem o ângulo limitado por $-0.4 \leq \theta_1 \leq 2$ enquanto a segunda junta é limitada por $-1 \leq \theta_2 \leq 1$.

A posição dos nós intermediários de interpolação e o perfil de torque associado ao movimento são apresentados na Figura 2.

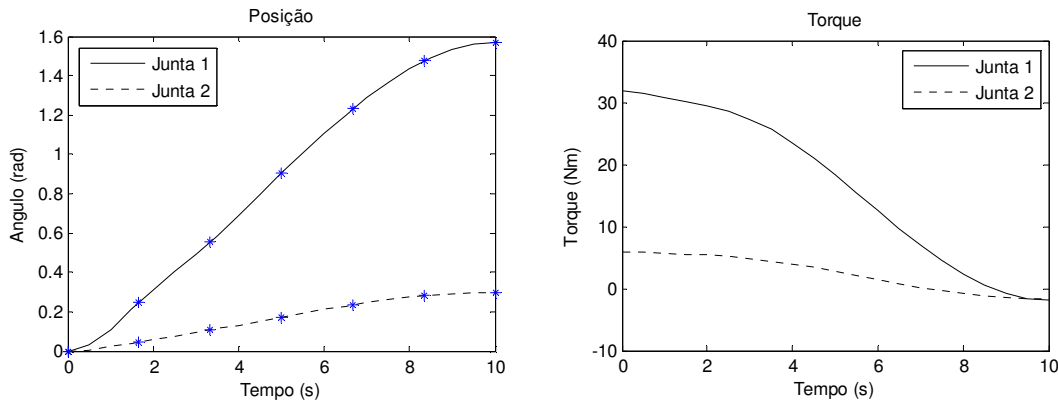


Figura 2: Nós intermediários de interpolação e correspondente perfil de torque.

As estratégias de otimização utilizam o conjunto de parâmetros padrão do Matlab, a exceção da população inicial, que contém a configuração da interpolação apresentada acima. Ao utilizar a informação da interpolação para o estabelecimento da trajetória obtém-se o valor inicial $f=19.9513$ para a função objetivo.

Diante a aleatoriedade característica aos processos evolutivos considerados, são realizadas 7 execuções com cada estratégia. Os resultados obtidos pelo algoritmo genético (GA), *pattern search* (PS), recozimento simulado (SA) e *threshold acceptance* (TA) são apresentados na Tabela 2. A coluna “Mínimo” apresenta o valor mínimo da função objetivo com o projeto ótimo, a coluna “Obj.” contém o número de chamadas para a função objetivo e a coluna “t (s)” apresenta o tempo computacional requerido. Estes dados se referem a um computador AMD Turion64 X2, 1.6 GHz.

Tabela 2: Resultado ótimo obtido pelas diferentes estratégias em cada experimento.

Exec.	GA			PS			SA			TA		
	Mínimo	Obj.	t (s)	Mínimo	Obj.	t(s)	Mínimo	Obj.	t(s)	Mínimo	Obj.	t(s)
1	8,6721	1400	5,2	6,1173	3874	13,9	6,3235	6534	27,7	8,0410	14358	69,8
2	7,5988	1180	4,0	6,1173	3874	13,4	6,6030	9846	41,8	6,8661	8270	43,5
3	8,1075	1160	3,9	6,1173	3874	13,4	6,4828	8421	35,5	7,0873	11022	54,9
4	8,1496	1060	3,61	6,1173	3874	13,4	6,6285	7802	32,7	7,8252	14920	71,6
5	7,1151	1280	4,3	6,1173	3874	13,4	6,5900	8125	34,2	7,8780	9583	48,4
6	7,6848	1640	5,5	6,1173	3874	13,4	6,8164	6906	28,8	7,8678	6775	36,1
7	7,3913	1140	3,9	6,1173	3874	13,4	6,3584	8613	36,4	7,1533	15925	76,3
Média	8	1266	4	6	3874	13	7	8035	34	8	11550	57
D.Padrão	0,52	197	0,72	0	0	0,18	0,17	1107	4,7	0,47	3561	15,5

A análise dos resultados mostra que o algoritmo genético é aquele que requer o menor tempo para a obtenção de uma solução. Entretanto, o projeto obtido pela estratégia *pattern search* é aquele que requer o menor torque, devido ao menor valor da função objetivo. Destaca-se também que esta estratégia foi a única que apresentou a mesma solução para as 7 execuções do método.

A melhor solução, obtida pela estratégia *pattern search*, tem como variáveis de projeto ótimas os vetores $\mathbf{p}_1=[1.3079 \ 1.5608 \ 1.5669 \ 1.5733 \ 1.5770]$ e $\mathbf{p}_2=[0.0525 \ 0.0380 \ 0.0028 \ 0.0100 \ 0.0273]$. O valor da função objetivo foi reduzido de $f=19.9513$ para $f=6.1173$.

A trajetória ótima e o torque correspondente são apresentados na Figura 3.

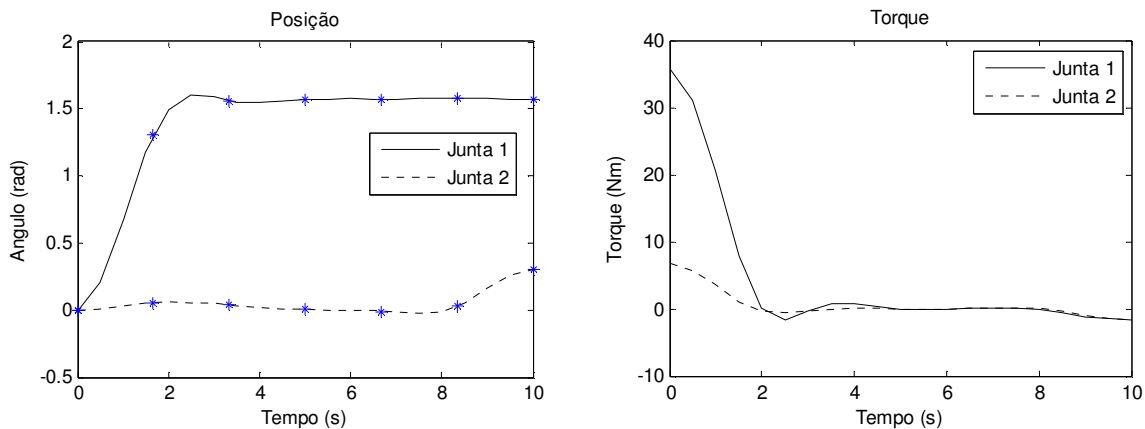


Figura 3: Trajetória e torque da configuração ótima obtida pela estratégia PS.

Adicionalmente, realizou-se a otimização com uma estratégia de programação não linear clássica. Diferentemente das estratégias consideradas anteriormente, ela é eficiente para a obtenção de um minimizador local a um custo computacional reduzido.

Nesta implementação considera-se o algoritmo de programação quadrática sequencial (QSP) e o conjunto de parâmetros padrão do Matlab. Neste caso é desnecessária a realização do

experimento repetidas vezes, uma vez que uma mesma configuração inicial levará sempre ao mesmo projeto ótimo (a técnica é determinística).

Ao se utilizar a configuração estabelecida pela interpolação como aproximação inicial para o processo de otimização, após 1006 avaliações da função objetivo (que requereu o tempo computacional de 5 segundos) obtém-se o projeto ótimo $\mathbf{p}_1 = [1.3263 \ 1.5502 \ 1.5668 \ 1.5716 \ 1.5748]$ e $\mathbf{p}_2 = [-0.0340 \ 0.0436 \ -0.0007 \ -0.0009 \ 0.0389]$. O valor da função objetivo foi reduzido de 19.9513 para 6.1008.

A trajetória ótima e o torque correspondente são apresentados na Figura 4.

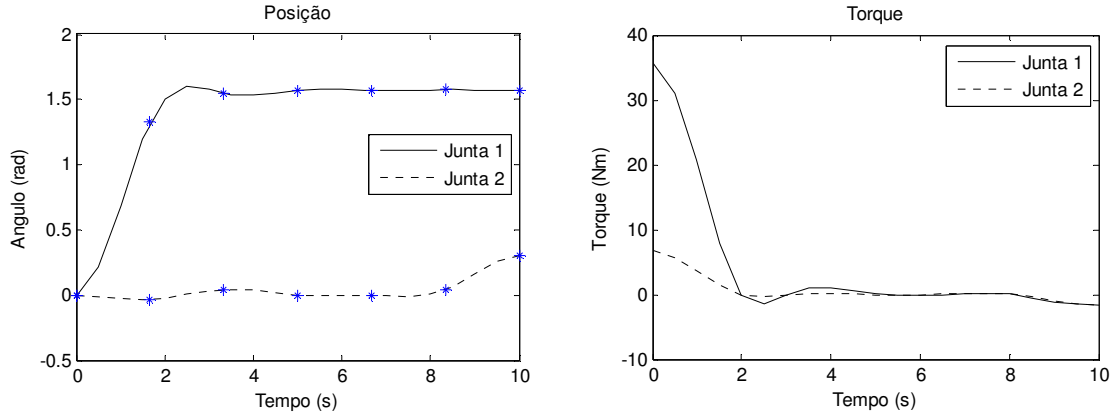


Figura 4: Trajetória e torque da configuração ótima obtida pela estratégia QSP.

Nota-se que a solução obtida por esta estratégia é bastante próxima daquela obtida pelos métodos heurísticos. A função objetivo, que no caso anterior apresentou o valor $f=6.1173$, agora tem o valor $f=6.1008$.

Entretanto, não é possível garantir que o método clássico terá desempenho superior aos métodos heurísticos em todos os problemas, uma vez que ele é eficiente apenas para a determinação do mínimo local. Para o caso aqui apresentado, a configuração ótima é semelhante àquela obtida pelas outras estratégias, e é, possivelmente, o mínimo global para o problema em questão.

6. CONCLUSÃO

Neste trabalho foi apresentada uma metodologia de obtenção da trajetória ótima para o movimento de robôs.

Inicialmente faz-se uma revisão sobre o planejamento de trajetórias de robôs manipuladores. Em seguida, juntamente com a definição dos nós intermediários da interpolação como variáveis de projeto, apresenta-se o equacionamento da dinâmica do sistema e a correspondente função objetivo a ser considerada no processo de otimização.

O uso de diversos nós intermediários para a especificação da interpolação permite a geração de trajetórias com relativa complexidade, ao mesmo tempo em que provê suavidade ao movimento. Como consequência, os picos de velocidade, torque e movimentos bruscos são evitados de forma eficiente.

Na seção sobre otimização são discutidas características gerais de diferentes métodos heurísticos comumente utilizados para a obtenção de minimizadores globais. Considera-se também o uso de um método clássico de programação não linear.

São realizados diferentes experimentos numéricos a partir da proposição de um movimento de referência.

Foram apresentados dados tabelados sobre o valor mínimo obtido pelos diferentes métodos, bem como o número de chamadas à função objetivo e o tempo computacional requerido.

No presente contexto, o algoritmo *Pattern Search* apresentou os melhores resultados, tendo exigido um tempo computacional moderado. É importante salientar que a análise atual não tem a

preocupação de estabelecer a “melhor” estratégia, mas servir como referência sobre “estratégias preferenciais” para o planejamento e otimização de trajetórias. Neste sentido, nota-se que todas as estratégias oferecem valores próximos para a função objetivo.

Entretanto, recomenda-se a análise também da média e desvio padrão da solução como critérios para a escolha de um método de otimização. O método preferencial é dependente também do recurso computacional disponível.

Em alguns casos o numero excessivo de cálculos da função objetivo (e o tempo computacional envolvido) inviabiliza o uso do método para a obtenção do projeto ótimo. Neste caso, a prioridade é dada aquele método que oferece uma boa solução em um tempo razoável. O Algoritmo Genético é o mais adequado dentre aqueles apresentados.

Por outro lado, caso a prioridade seja o projeto ótimo, recomenda-se o *Pattern Search* para os casos onde não há idéia sobre uma possível configuração ótima.

Entretanto, para aqueles casos onde há uma estimativa sobre uma possível solução ótima, recomenda-se o uso da Programação Quadrática Sequencial.

Quanto ao projeto ótimo apresentado, nota-se um pequeno aumento do torque requerido pelo robô durante os instantes iniciais do movimento, e a seguir uma redução significativa a valores próximos de zero. Este perfil reflete o objetivo considerado, que é a redução do torque total utilizado para o movimento. Adicionalmente, podem-se estabelecer parâmetros para a redução do torque máximo, ou uma combinação entre estes e outros aspectos.

Finalmente, conclui-se que as estratégias apresentam desempenho semelhante na resolução do problema proposto, ao se considerar limites máximos e mínimos para as variáveis de projeto. Por este motivo, acredita-se que a formulação proposta é robusta no sentido de prover uma configuração ótima que atende a critérios de suavidade da trajetória e respeita os limites físicos do robô, com certa independência quanto a estratégia de otimização aplicada.

A perspectiva de continuidade deste trabalho consiste na utilização de recursos de processamento paralelo para a redução do tempo computacional requerido pelos métodos heurísticos, além de uma estimativa do melhor momento para a utilização conjunta de um método clássico na busca do mínimo global.

7. AGRADECIMENTOS

O primeiro autor é grato à FAPEMIG (TEC 00214/06) e o segundo autor é grato ao CNPq (Proc. 470346/2006-0) pelo financiamento parcial deste trabalho de pesquisa.

8. REFERÊNCIAS

- Conn, A. R., Gould, N. I. M. and Toint, P. L., 1991, “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds”, *SIAM Journal on Numerical Analysis*, Vol. 28, N. 2, pp. 545–572.
- Conn, A. R., Gould, N. I. M. and Toint, P. L., 1997, “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds”, *Mathematics of Computation*, Vol. 66, N. 217, pp 261–288.
- Dueck, G. and Scheuer, T., 1990, “Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing”, *J. Comp. Phys.* 90, pp. 161-175.
- Fletcher, R., 1987, “Practical Methods of Optimization”, John Wiley and Sons.
- Fu, K. S., Gonzales, R. C. and Lee, C. S. G., 1987, “Robotics: control, sensing, vision and intelligence”, McGraw-Hill.
- Gill, P.E., Murray, W. and Wright, M.H., 1981, “Practical Optimization”, London, Academic Press.
- Hwang, Y.K. and Ahuja, N., 1992, “Gross motion planning - A survey”, *ACM Computing Surveys (CSUR) archive*. ACM Press (ISSN:0360-0300). Vol. 24, N. 3, p. 219-291.
- Ingber, L., 1993, “Simulated Annealing: Practice Versus Theory”, *Math. Comput. Modelling* 18, pp. 29-57.
- Khatib, O., 1986, “Real-time obstacle avoidance for manipulators and mobile robots”, *Int. J. Robot. Res.*, pp. 90-98.
- Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, “Optimization by Simulated Annealing”, *Science*, Vol. 220, pp. 671-680.
- Lewis, R. M. and Torczon, V., 2002, “A Globally Convergent Augmented Lagrangian Pattern Search Algorithm for Optimization with General Constraints and Simple Bounds”, *SIAM Journal on Optimization*, Vol. 12, N. 4, pp. 1075–1089.

- Powell, M.J.D., 1983, "Variable Metric Methods for Constrained Optimization", Mathematical Programming: The State of the Art, (A. Bachem, M. Grottschel and B. Korte, eds.) Springer Verlag, pp 288-311.
- Santos, R. R., Saramago, S. F. P. e Steffen Jr, V., 2005c, "Otimização do Torque Aplicado pelos Atuadores de Robôs Usando Técnicas de Controle Ótimo", In: Simpósio do Programa de Pós-Graduação em Engenharia Mecânica, 2005, Uberlândia – MG.
- Santos, R. R., Saramago, S. F. P. e Steffen Jr, V., 2006c, "Trajetória ótima de robôs manipuladores ao interceptar objetos em movimento", In: Simpósio do Programa de Pós-Graduação em Engenharia Mecânica, 2006, Uberlândia – MG.
- Santos, R. R., Steffen Jr, V. e Saramago, S. F. P., 2006a, "Técnicas de Controle Ótimo Multicritério Aplicadas à Otimização de Aspectos Dinâmicos no Planejamento de Trajetórias de Robôs Manipuladores", In: VII Simpósio de Mecânica Computacional, 2006, Araxá. Anais do VII Simpósio de Mecânica Computacional, pp. 1-10.
- Santos, R. R., Steffen Jr, V. and Saramago, S. F. P., 2005, "Robot Path Planning: Avoiding Obstacles", In: 18th International Congress of Mechanical Engineering (COBEM), 2005, Ouro Preto – MG.
- Santos, R. R., Steffen Jr, V. and Saramago, S. F. P., 2006b, "Robot Path Planning in a Constrained Workspace by using Optimal Control Techniques", In: III European Conference on Computational Mechanics Solids, Structures and Coupled Problems in Engineering, Lisboa.
- Saramago, S. F. P and Steffen Jr, V., 1998, "Optimization of the Trajectory Planning of Robot Manipulators Taking Into-Account The Dynamics of the System", Mechanism and Machine Theory, Vol. 33, N. 7, pp. 883-894.
- Schmitt, L. M., 2001, "Theory of Genetic Algorithms", Theoretical Computer Science, Vol. 259, pp. 1-61.
- Schmitt, L. M., 2004, "Theory of Genetic Algorithms II: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling", Theoretical Computer Science, Vol. 310, pp. 181-231.
- Seraji, H., Steele, R. and Ivlev, R., 1996, "Sensor-based collision avoidance: Theory and experiments". J. Robot. Syst. Vol. 13, N. 9, p. 571-586.
- Vose, M. D., 1999, "The Simple Genetic Algorithm: Foundations and Theory", MIT Press, Cambridge, MA.

HEURISTIC STRATEGIES APPLIED TO THE OPTIMAL PATH PLANNING OF ROBOTS

Rogério Rodrigues dos Santos

School of Mechanical Engineering, Federal University of Uberlândia
Av. João Naves de Ávila, 2160 - Campus Santa Mônica - Bloco 1M - Uberlândia/MG
rrsantos@mecanica.ufu.br

Valder Steffen, Jr.

School of Mechanical Engineering, Federal University of Uberlândia
vsteffen@mecanica.ufu.br

Abstract: *Robotic manipulators are programmable devices developed to perform a great variety of tasks in a repetitive way. In the industrial environment, the productivity increase and the operational cost decrease can be achieved through the minimization of dynamics efforts, required to perform a given task. Furthermore, movements that require the maximum system capacity can be successfully performed by means of a suitable adjust of the path. In this work a purpose of total torque reduction is presented, by using global optimization techniques and the dynamics characterization of the robot. A methodology to adjust the trajectory through the interpolation of intermediate nodes of joint coordinates is proposed. Numerical results show the performance of optimization strategies and demonstrate the viability of the proposed methodology.*

Keywords: *path planning, global optimization, robotic manipulator.*