

OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS DE CONTROLE MULTIVARIÁVEL EM UMA APLICAÇÃO DE ROBÓTICA DE MANIPULADORES

HELON VICENTE HULTMANN AYALA E LEANDRO DOS SANTOS COELHO

*Laboratório de Automação e Sistemas, Centro de Ciências Exatas e Tecnologia
Programa de Pós-Graduação em Engenharia de Produção e Sistemas, PUCPR / PPGEPS
Pontifícia Universidade Católica do Paraná
Rua Imaculada Conceição, 1155, CEP 80215-901, Curitiba, PR, Brasil
E-mails: helonayala@gmail.com, leandro.coelho@pucpr.br*

Abstract This paper presents a study of gains optimization for PD and PID controllers dealing the control of a robotic manipulator with two degree-of-freedom. The proposed optimization method is an evolutionary computation approach called the Particle Swarm Optimization (PSO). The optimization objective of PSO algorithm is the minimization of the errors of position and velocity variables and also the effort of control. Simulation results and performance analysis of PD and PID controllers using PSO are presented in paper.

Keywords Robotic manipulator, PID control, particle swarm optimization, optimization.

Resumo Este artigo apresenta um estudo sobre a otimização dos ganhos de controladores PD (Proporcional-Derivativo) e PID (Proporcional-Integral-Derivativo) visando o controle de um manipulador robótico com dois graus de liberdade. O método de otimização proposto é uma abordagem de computação evolutiva denominada enxame (ou nuvem) de partículas (*Particle Swarm Optimization*, PSO). O objetivo da otimização do algoritmo PSO é minimizar o erro das variáveis de posição e velocidade e também o esforço de controle. Os resultados de simulação e uma análise de desempenho dos controladores PD e PID usando PSO são apresentados no artigo.

Palavras-chave Manipulador robótico, controle PID, enxame de partículas, otimização.

1 Introdução

O projeto de controladores PI (Proporcional-Integral), PD (Proporcional-Derivativo) e PID (Proporcional-Integral-Derivativo) são tópicos de pesquisa relevantes na área de controle de processos (Åström e Hägglund, 1995).

Muitos operadores de chão de fábrica e engenheiros, muitas vezes, ajustam os parâmetros dos controladores PI, PD e PID por procedimentos de tentativa e erro, ou seja, a sintonia do controlador consiste em deduzir, partindo da resposta do sistema, quando este é sujeito a entradas específicas, determinados valores que vão permitir o cálculo dos referidos parâmetros. No entanto, os procedimentos de tentativa e erro requerem muita experiência do operador e podem demandar muito tempo para o ajuste dos parâmetros. A vantagem deste procedimento é não existir necessidade de conhecer o modelo matemático do sistema a ser controlado (por vezes, difícil de determinar). Pode-se assim concluir que deve-se recorrer a este tipo de procedimento quando o custo de sintonia do controlador for inferior ao custo associado à análise do sistema e projeto do controlador adequado (Fernandes Júnior, 2006).

Na literatura, diversos métodos para a sintonia de controladores PI, PD e PID têm sido propostos, tais como os métodos clássicos de Ziegler-Nichols (Ziegler e Nichols, 1942), lugar das raízes, alocação de pólos, entre outros (Wang *et al.*, 1995, Åström e Hägglund, 2001, Cominos e Munro, 2002).

Entretanto, no intuito de suprir certas limitações dos métodos clássicos de sintonia que, na maioria das vezes, não permitem especificações de projeto que minimizem alguma função objetivo vinculada a erro em regime permanente, sobre-elevação e tempo de resposta, pode-se utilizar abordagens de otimização da computação evolutiva (Fleming e Purshouse, 2002), tais como os algoritmos genéticos (Dimeo e Lee, 1995; Daley e Liu, 1999), estratégias evolutivas (Coelho e Coelho, 1998) e programação evolutiva (Xiolan *et al.*, 2002).

Neste contexto, uma abordagem interessante de ser avaliada é a de otimização por enxame (ou nuvem) de partículas (*Particle Swarm Optimization*, PSO) para sintonia de parâmetros de controladores PID.

A PSO foi desenvolvida, inicialmente, por Kennedy e Eberhart (1995) baseada nos estudos do sócio-biologista Edward Osborne Wilson (Brandstätter e Baumgartner, 2002). A PSO constitui uma técnica da inteligência coletiva (*swarm intelligence*) baseada em uma população de soluções e transições aleatórias. A PSO apresenta características similares a técnicas da computação evolutiva (ou evolucionária), que são baseadas em uma população de soluções. Entretanto, a PSO é motivada pela simulação de comportamento social e cooperação entre agentes em vez da sobrevivência do indivíduo mais apto. Na PSO, cada solução candidata (denominada partícula) possui associada uma velocidade. A velocidade é ajustada através de uma equação de atualização que considera a experiência

da partícula correspondente e a experiência das outras partículas da população.

A literatura menciona algumas pesquisas sobre a utilização do algoritmo PSO na otimização de controle *fuzzy* (Ghoshal, 2004), controle adaptativo (Coelho e Guerra, 2007) e controle preditivo (Coelho *et al.*, 2005).

A contribuição deste trabalho é propor uma abordagem de PSO para otimização dos ganhos de dois controladores PID para uma aplicação em robótica de manipuladores. Neste contexto, os ganhos dos controladores PID são atrelados em uma função objetivo a ser minimizada com os seguintes objetivos ponderados: (i) minimização de erro de trajetória da posição do manipulador e também (ii) minimização o esforço de controle.

O restante deste artigo é organizado da seguinte forma. O modelo matemático do manipulador robótico e a configuração do projeto de controle multivariável do tipo PD e PID são apresentados na seção 2. Uma breve descrição do PSO e as etapas para sua aplicação na otimização dos ganhos de controladores PID é apresentada na seção 3. Na seção 4, os resultados de simulação são analisados. A conclusão é apresentada na seção 5.

2 Descrição do Manipulador Robótico

As equações dinâmicas do robô constituem um conjunto de equações diferenciais acopladas contendo termos, tais como: inércia variante, torque centrífugo e Coriolis, termos de carregamento e gravidade. O movimento do efetuador final em uma particular trajetória com particular velocidade requer um conjunto complexo de funções de torque para serem aplicadas a atuadores na(s) junta(s) do manipulador robótico. A seguir apresenta-se a descrição do modelo matemático do robô e as configurações de controle otimizadas.

O modelo do manipulador usualmente considera o vetor de representação da dinâmica de um manipulador robótico de Q -graus de liberdade (neste caso, $Q=2$) regido pela equação:

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

onde $M(\mathbf{q}) \in \hat{\mathbf{A}}^{n \times n}$ é a matriz de inércia do sistema definida positiva, $C(\mathbf{q}, \dot{\mathbf{q}}) \in \hat{\mathbf{A}}^{n \times 1}$ é um vetor que representa os efeitos do torque centrífugo e de Coriolis, $G(\mathbf{q}) \in \hat{\mathbf{A}}^{n \times 1}$ é o vetor relativo ao efeito dos torques gravitacionais, $\mathbf{t} \in \hat{\mathbf{A}}^{n \times 1}$ é o vetor do torque da(s) junta(s), e, \mathbf{q} , $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ são os termos de posição angular, velocidade e aceleração da(s) junta(s). O modelo dinâmico para o manipulador robótico de dois graus de liberdade utilizado, conforme apresentado na Fig. 1, tal que (Craig, 1996):

$$\begin{aligned} \mathbf{t}_1 = & m_2 l_2^2 (\ddot{\mathbf{q}}_1 + \ddot{\mathbf{q}}_2) + m_2 l_1 l_2 c_2 (2\ddot{\mathbf{q}}_1 + \ddot{\mathbf{q}}_2) + \\ & (m_1 + m_2) l_1^2 \ddot{\mathbf{q}}_1 - m_2 l_1 l_2 s_2 \ddot{\mathbf{q}}_2^2 \\ & - 2m_2 l_1 l_2 s_2 \dot{\mathbf{q}}_1 \dot{\mathbf{q}}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1 \end{aligned} \quad (2)$$

$$\begin{aligned} \mathbf{t}_2 = & m_2 l_1 l_2 c_2 \ddot{\mathbf{q}}_1 + m_2 l_1 l_2 s_2 \dot{\mathbf{q}}_1^2 + \\ & m_2 l_1 g c_{12} + m_2 l_2^2 (\ddot{\mathbf{q}}_1 + \ddot{\mathbf{q}}_2) \end{aligned} \quad (3)$$

onde $s_1 = \text{sen}(\mathbf{q}_1)$, $s_2 = \text{sen}(\mathbf{q}_2)$, $c_1 = \text{cos}(\mathbf{q}_1)$, $c_2 = \text{cos}(\mathbf{q}_2)$, e $c_{12} = \text{cos}(\mathbf{q}_1 + \mathbf{q}_2)$. Os subscritos 1 e 2 denotam os parâmetros das juntas 1 e 2, respectivamente. Os parâmetros adotados nas simulações são comprimentos — $l_1=0,8$ m e $l_2=0,4$ m, massas — $m_1=m_2=0,1$ kg, e aceleração da gravidade de $g=9,81$ m/s² (Mital e Chin, 1995). As restrições impostas aos torques \mathbf{t}_1 e \mathbf{t}_2 são [-1000; 1000] Nm. Um período de amostragem, T_s , de 10 ms é adotado neste trabalho.

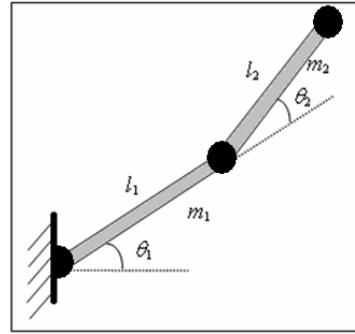


Fig. 1. Manipulador robótico de dois graus de liberdade.

Define-se \mathbf{q}_{dj} , $\dot{\mathbf{q}}_{dj}$ e $\ddot{\mathbf{q}}_{dj}$ como os valores de posição angular, velocidade e aceleração desejados à junta robótica. Os vetores dos erros de posição, velocidade e aceleração são dados por

$$d_j(t) = \mathbf{q}_j(t) - \mathbf{q}_{dj}(t), \text{ onde } j \in \hat{\mathbf{A}} / j = \{1, 2\} \quad (4)$$

$$v_j(t) = \dot{\mathbf{q}}_j(t) - \dot{\mathbf{q}}_{dj}(t), \text{ onde } j \in \hat{\mathbf{A}} / j = \{1, 2\} \quad (5)$$

$$a_j(t) = \ddot{\mathbf{q}}_j(t) - \ddot{\mathbf{q}}_{dj}(t), \text{ onde } j \in \hat{\mathbf{A}} / j = \{1, 2\} \quad (6)$$

2.1 Projeto de controlador PD

Um projeto avaliado, neste artigo, é o controle do tipo PD *multiloop* na forma digital. Em termos práticos a ação de controle é regida pelos torques \mathbf{t}_1 e \mathbf{t}_2 , aplicados ao elo das juntas. Contudo, em controle de manipuladores reais é necessária a multiplicação dos torques por uma constante que de a relação adequada para obtenção da ação de controle, $u_j(t)$, a ser aplicada ao motor (atuador) considerando o circuito de atuação e os parâmetros inerentes ao experimento prático.

A forma do controle PD analógico multivariável é dada pela seguinte equação (Coelho e Coelho, 1998):

$$\mathbf{t}_j(t) = K_{p,j} d_j(t) + K_{v,j} v_j(t) \quad (7)$$

onde $\mathbf{t}_j(t)$ é o torque aplicado as juntas do manipulador robótico.

Após a discretização da equação (7) obtém-se a equação a diferenças do controlador PD regida por

$$\begin{aligned} \mathbf{t}_j(t) = & (K_{p,j} + K_{v,j}/T_s) \mathbf{q}(t) - (K_{v,j}/T_s) \mathbf{q}(t-1) - \\ & (K_{p,j} + K_{v,j}/T_s) \mathbf{q}_{d,j}(t) + K_{v,j}/T_s \mathbf{q}_{d,j}(t-1), \end{aligned} \quad (8)$$

onde $j \in \hat{A} / j = \{1, 2\}$

As matrizes de ganho $K_{p,j}$ e $K_{v,j}$ têm propriedades diagonais e definidas positivas e, são proporcionais aos erros de posição e velocidade.

2.2 Projeto de controlador PID

O controle PID analógico multivariável é dado por (Coelho e Coelho, 1998):

$$\mathbf{t}_j(t) = K_{p,j} d_j(t) + K_{v,j} v_j(t) + K_{i,j} \int_0^t d(s) ds \quad (9)$$

onde $K_{i,j}$ é uma matriz diagonal definida positiva responsável pelo termo integral do PID e $\mathbf{t}_j(t)$ é o torque aplicado as juntas do manipulador robótico. A discretização da equação (9) conduz a lei de controle PID discreto

$$\begin{aligned} D\mathbf{t}_j(t) = & (K_{p,j} + 0,5T_s K_{i,j}) d_j(t) - K_{p,j} \\ & - 0,5T_s K_{i,j} d_j(t-1) + K_{v,j} v_j(t) + K_{v,j} v_j(t-1) \end{aligned} \quad (10)$$

$$\begin{aligned} v_j(t) = & [\mathbf{q}_{d,j}(t) - \mathbf{q}_{d,j}(t-1)] / T_s - \\ & [\mathbf{q}_j(t) - \mathbf{q}_j(t-1)] / T_s, \text{ onde } j \in \hat{A} / j = \{1, 2\} \end{aligned} \quad (11)$$

3 Otimização Baseada em PSO

A PSO é um procedimento de otimização estocástica inspirado em princípios de cooperação e comportamento em sociedade de enxames, cardume de peixes e bandos de pássaros. O PSO foi proposto por Kennedy e Eberhart (1995) e rapidamente o algoritmo popularizou-se como um otimizador global devido sua eficiência e custo computacional relativamente baixo (Coello e Reyes-Serra, 2006), pois o PSO utiliza poucas funções matemáticas para realizar o procedimento de busca e otimização para obtenção de uma solução N -dimensional.

Como o algoritmo PSO é uma técnica da computação evolutiva (obs.: alguns autores a definem como uma abordagem da inteligência coletiva), este trabalha a partir de uma população de partículas fazendo-os evoluir até que se ache uma solução aceitável. Entretanto, diferentemente dos algoritmos evolutivos clássicos, que utilizam uma população inicial de soluções (indivíduos) gerada aleatoriamente com uma distribuição uniforme, o algoritmo PSO trabalha com uma população de partículas em um enxame, onde cada partícula i tem sua posição x_i e velocidade v_i atualizados a cada geração t (iteração do algoritmo) através de fatores de sua própria memória (fator cognitivo), $co_i(t)$, e de seu conhecimento social (fator social), $so_i(t)$. Estes fatores são regidos respectivamente pelas ponderações: social c_1 e cognitiva c_2 .

Resumindo, o comportamento de cada partícula i é definido segundo as equações que se seguem:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot co_i(t) + c_2 \cdot r_2 \cdot so_i(t) \quad (12)$$

onde

$$co_i(t) = pbest_i - x_i(t) \quad (13)$$

$$so_i(t) = gbest - x_i(t) \quad (14)$$

$$x_i(t+1) = x_i(t) + \Delta t \cdot v_i(t+1) \quad (15)$$

onde $i=1, \dots, npop$, $npop$ é o tamanho da população, x_i e v_i denotam respectivamente a posição e a velocidade da i -ésima partícula, $pbest$ (*personal best*) e $gbest$ (*global best*) são, respectivamente, a melhor posição obtida por uma partícula em uma determinada posição e de toda população em uma determinada vizinhança (enxame); w é a constante de inércia, r_1 e r_2 são números aleatórios gerados usando uma distribuição uniforme no intervalo $[0, 1]$ e $\Delta t = 1$.

Na Fig. 2 é apresentado o pseudocódigo do PSO com uma topologia de vizinhança global, testado neste trabalho, para um problema de minimização do valor da função objetivo.

Algoritmo PSO
<p>Início do Algoritmo</p> <p>$t = 1$;</p> <p>% Gera a população inicial de partículas</p> <p>Faça $j = 1$ a N % N: dimensão do problema</p> <p> Faça $i = 1$ a $npop$</p> <p> $x_{i,j}(t) =$ Gera aleatório $\in [limite_{min}, limite_{max}]$;</p> <p> $v_{i,j}(t) =$ Gera aleatório $\in [v_{min}, v_{max}]$;</p> <p> Fim Faça i</p> <p> $f_i(t) =$ Avalia ($x_i(t)$); % minimização</p> <p> Fim Faça j</p> <p>Faça $t = 1$ a Máximo de gerações (t_{max})</p> <p> $pbest_i = x_i(t)$; % melhor da posição i</p> <p> $gbest = x_{melhor}(t)$; % melhor de todas</p> <p> Faça $i = 1$ a $npop$</p> <p> Calcula $co_i(t)$ e $so_i(t)$, $i = 1, npop$</p> <p> Calcula $x_i(t+1)$ e $v_i(t+1)$, $i = 1, npop$</p> <p> $f_i(t+1) =$ Avalia ($x_i(t+1)$);</p> <p> Se $f_i(t+1) < f(pbest_i)$ então $pbest_i = x_i(t+1)$;</p> <p> Se $f_i(t+1) < f(gbest)$ então $gbest = x_i(t+1)$;</p> <p> Fim Faça i</p> <p> $t = t + 1$;</p> <p> Fim Faça t</p> <p>Fim do Algoritmo</p> <p>Melhor solução do algoritmo PSO: $gbest$</p>

Fig. 2. Pseudocódigo para o algoritmo PSO.

Na literatura (Ratnaweera *et al.*, 1996; Kennedy *et al.*, 2001) são mencionadas diversas formas de definir os parâmetros que ponderam a interferência cognitiva social e cognitiva em cada etapa do algoritmo PSO, bem como da ponderação w , que controla a convergência das partículas.

O valor definido para w afeta o tipo de busca (Shi e Eberhart, 2000), pois para um valor pequeno de w , a busca é local e para um valor alto de w a busca torna-se global. Este parâmetro pode variar

linearmente de um valor alto para um baixo de forma que a busca seja global no início e depois local no final do procedimento.

Os parâmetros c_1 e c_2 são, muitas vezes, definidos, tal que $c_1 = c_2 = 1$, ou mesmo seguindo a relação regra $c_1 + c_2 = 4$ (Kennedy e Eberhart, 1995). Pode-se mencionar que as constantes c_1 , c_2 e w podem ser interpretadas como a confiança que cada partícula tem em sua posição corrente, sua própria experiência e na experiência da vizinhança, respectivamente (Coelho *et al.*, 2005).

A velocidade das partículas em cada dimensão são limitadas a um valor máximo de velocidade, v_{max} .

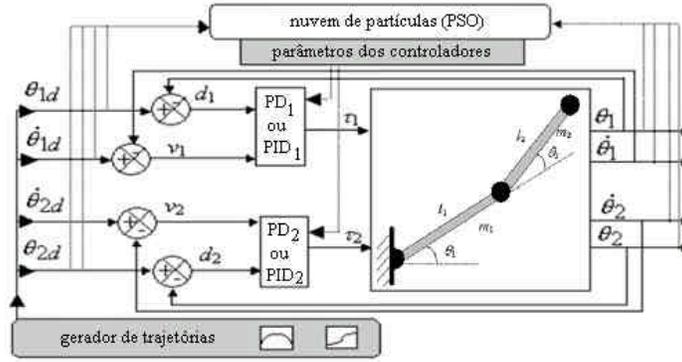


Fig. 3. Representação da otimização dos controladores baseada em PSO.

4 Resultados das Simulações

A dinâmica do robô manipulador é simulada pelo método Runge-Kutta de 4ª ordem. O algoritmo PSO otimiza o projeto dos controladores PD ou PID visando satisfazer um polinômio de 5ª ordem que representa a trajetória a ser seguida pelo manipulador para posição, velocidade e aceleração (Craig, 1996) é regida por

$$\mathbf{q}_{d,j}(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5, \quad \text{onde } j \in \{1, 2\} \quad (16)$$

onde as restrições são:

$$\begin{aligned} \mathbf{q}_{0,j} &= a_0 \\ \mathbf{q}_{df,j} &= a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \\ \mathbf{q}_{0,j} &= a_1 \\ \dot{\mathbf{q}}_{df,j} &= a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4 \\ \ddot{\mathbf{q}}_{0,j} &= 2a_2 \\ \ddot{\mathbf{q}}_{df,j} &= 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3 \\ &\text{onde } j \in \{1, 2\} \quad (17) \end{aligned}$$

onde $\mathbf{q}_{d,j}$ é a posição instantânea desejada, \mathbf{q}_0 , $\dot{\mathbf{q}}_0$, $\ddot{\mathbf{q}}_0$ são os valores adotados para a posição, velocidade e aceleração inicial, respectivamente. De forma análoga $\mathbf{q}_{df,j}$, $\dot{\mathbf{q}}_{df,j}$, $\ddot{\mathbf{q}}_{df,j}$ são os valores desejados para a posição ($\mathbf{q}_{df,1} = 3$ rad e $\mathbf{q}_{df,2} = 1$ rad do tempo t de 0 a 2 s, e $\mathbf{q}_{df,1} = 2$ rad e $\mathbf{q}_{df,2} = 4$ rad do tempo t_f de 2 a 4 s), velocidade ($\dot{\mathbf{q}}_{df,1} = \dot{\mathbf{q}}_{df,2} = 0$ rad/s) e aceleração final ($\ddot{\mathbf{q}}_{df,1} = \ddot{\mathbf{q}}_{df,2} = 0$ rad/s²), respectivamente.

O v_{max} é importante, pois determina a resolução que a região próxima a soluções atuais são procuradas. Se v_{max} é alto, a PSO facilita a busca global, enquanto um valor v_{max} pequeno enfatiza as buscas locais. Neste trabalho adota-se o v_{max} como 20% da diferença entre o limite máximo e mínimo de cada partícula i para determinada dimensão N .

Na Fig. 3 é apresentada uma representação da do algoritmo PSO para otimização dos parâmetros dos controladores PI ou PID na aplicação de controle de um manipulador robótico com dois graus de liberdade.

Os testes realizados objetivaram a aplicação de controle PD e também PID multivariável. O espaço de busca dos parâmetros (ganhos) dos controladores, ou seja, $K_{p,1}$, $K_{p,2}$, $K_{v,1}$ e $K_{v,2}$ para o controle PD e $K_{p,1}$, $K_{p,2}$, $K_{v,1}$, $K_{v,2}$, $K_{i,1}$ e $K_{i,2}$ para o PID foi de $[\limite_{min}, \limite_{max}] = [0; 350]$ para $K_{p,1}$, $K_{p,2}$ e $[0; 30]$ para $K_{v,1}$, $K_{v,2}$, $K_{i,1}$ e $K_{i,2}$. O tamanho de população, n_{pop} , escolhida para o algoritmo PSO é de 20 partículas e o ciclo evolutivo tem critério de parada estipulado em t_{max} com 100 gerações. Neste trabalho foi adotado $c_1 = c_2 = 2$ e fator de inércia, w , com decréscimo linear de 0,9 a 0,4 durante o ciclo evolutivo.

A função objetivo f utilizada no PSO visa considerar os ganhos dos controladores que melhor minimizem os erros de posição e velocidade do elo 1 e 2 e também o esforço de controle (torque), estes dados pela seguinte equação ponderada a ser minimizada:

$$f = \sum_{t=1}^{t_f} [d_1(t)]^2 + 10^{-25} \sum_{t=1}^{t_f} [v_1(t)]^2 + 0,01 \sum_{t=1}^{t_f} |t_1(t)| + \sum_{t=1}^{t_f} [d_2(t)]^2 + 10^{-25} \sum_{t=1}^{t_f} [v_2(t)]^2 + 0,01 \sum_{t=1}^{t_f} |t_2(t)| \quad (18)$$

Deve enfatizar que as ponderações (função com múltiplos objetivos) para os erros de posição, velocidade e torque na equação (18) foram obtidos por tentativa e erro.

Os resultados da otimização usando PSO para a sintonia dos controladores multivariáveis é resumido na Tabela 1.

Tabela 1. Resumo dos resultados da função objetivo f para 30 simulações com cada abordagem de controle.

controle	média	mínimo	máximo
PD	109,3968	41,5154	709,1879
PID	41,6504	41,3628	41,8630

Na Fig. 4 é apresentado o melhor resultado (valor mínimo de f para o PD da Tabela 1) com controle PD obtendo função objetivo $f=41,5154$. Os ganhos do controle PD foram: $K_{p,1}=191,6429$; $K_{v,1}=3,6419$; $K_{p,2}=22,3729$ e $K_{v,2}=0$.

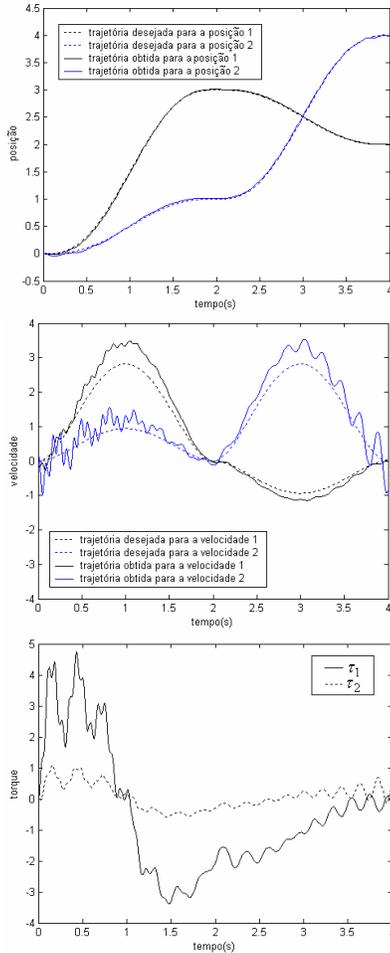


Fig. 4. Resposta em malha fechada com controle PD.

E na Fig. 5 é mostrado o melhor resultado (valor mínimo de f para o PID da Tabela 1) com controle PID, onde $f=41,3628$. Os ganhos do PID, neste caso, foram: $K_{p,1}=176,4534$; $T_{i,1}=12,1300$; $K_{v,1}=3,8194$; $K_{p,2}=22,2029$; $T_{i,1}=11,1390$ e $K_{v,2}=0$.

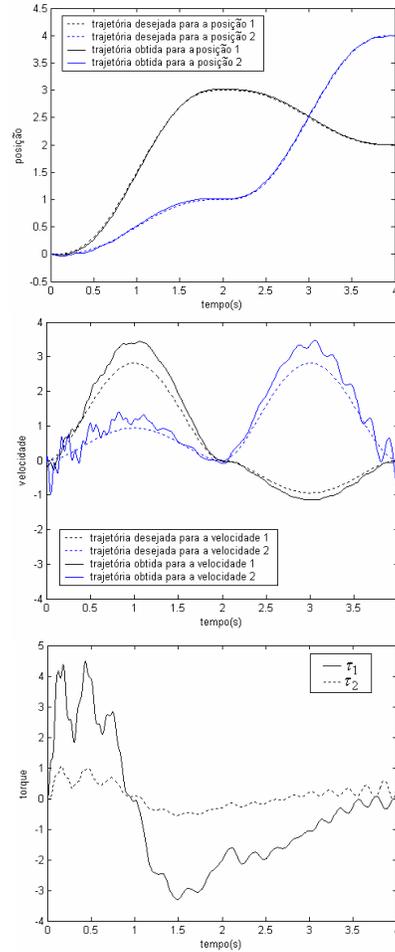


Fig. 5. Resposta em malha fechada com controle PID.

Nota-se pelos resultados apresentados nas Figs. 4 e 5 que tanto para o melhor resultado de controle PD quanto para o PID houve uma boa aproximação no seguimento da trajetória desejada para a posição dos elos 1 e 2. No entanto, o projeto das ponderações adotadas na função objetivo para o sinal de controle (torques) e trajetória das velocidades deva ser aprimorado ou mesmo adotar-se uma abordagem de otimização com múltiplos objetivos, similar a apresentada em Ayala e Coelho (2006).

5 Conclusão

Este artigo apresentou e avaliou a aplicação do algoritmo PSO na otimização de duas abordagens de controle multivariável (PD e PID *multiloop*) aplicados ao controle de um manipulador robótico de dois graus de liberdade.

De acordo com o resultado apresentados na Tabela 1 e Figs. 4 e 5, nota-se que a otimização baseada em PSO foi apropriada no seguimento da trajetória para a posição dos elos 1 e 2.

Futuros trabalhos objetivam otimização e análise de estabilidade e robustez de controladores PD e PID em robótica combinados a métodos de controle

adaptativo baseado em modelo e otimização com múltiplos objetivos.

Agradecimentos

Ao Programa Insitucional de Bolsas de Iniciação Científica – PIBIC da PUCPR pela bolsa concedida (projeto número 60452) ao primeiro autor deste artigo.

Referências Bibliográficas

- Åström, K. J.; Hägglund T. (1995). *PID Controllers: Theory, Design and Tuning*, Instrument Society of America, Research Triangle Park, North Carolina, USA.
- Åström, K. J.; Hägglund T. (2001). The future of PID control, *Control Engineering Practice*, **9**(11): 1163-1175.
- Ayala, H. V. H. e Coelho, L. S. (2006). Algoritmo genético multiobjetivo aplicado à otimização do controle de um manipulador robótico, *I Workshop on Computational Intelligence*, Internaitonal Joint Conference IBERAMIA-SBIA-SBRN, Ribeirão Preto, SP.
- Brandstätter, B. e Baumgartner, U. (2002). Particle swarm optimization – mass-spring systems analogon, *IEEE Transactions on Magnetics*, **38**(2): 997-1000.
- Coelho, J. P., Oliveira, P. B. M. e Cunha, J. B. (2005). Greenhouse air temperature predictive control using the particle swarm optimization algorithm, *Computers and Electronics in Agriculture*, **49**(3): 330-344.
- Coelho, L. S. e Coelho, A. A. R. (1998). Projeto e sintonia evolutiva de controladores PD e PID com aplicação em um manipulador robótico, *Anais do XII Congresso Brasileiro de Automática*, Uberlândia, MG, pp. 380-386.
- Coelho, L. S. e Guerra, F. (2007). Applying particle swarm optimization to adaptive controller, *Soft Computing in Industrial Applications: Recent and Emerging Methods and Techniques*, Saad, A.; Avineri, E., Dahal, K., Sarfraz, M., Roy, R. (eds.), Springer-Verlag, Germany, pp. 82-91.
- Coello, C. A. C. e Reyes-Serra, M. (2006). Multi-objective particle swarm optimizers: a survey of the state-of-the-art, *International Journal of Computational Intelligence Research*, **2**(3): 287-308.
- Cominos, P. e Munro, N. (2002). PID controllers: recent tuning methods and design to specification, *IEE Proceedings-Control Theory Appl.*, **149**(1): 46-53.
- Craig, J. J. (1996). *Introduction to Robotics: Mechanics & Control*, Addison-Wesley.
- Daley, S. e Liu, G. P. (1999). Optimal PID tuning using direct search algorithms, *Computers & Control Engineering Journal*, **10**(2), pp. 51-56.
- Dimeo, R. e Lee, K. Y. (1995). Boiler-turbine control system design using a genetic algorithm, *IEEE Transactions on Energy Conversion*, **10**(4): 752-759.
- Fernandes Júnior, F. G. (2006). Metodologia para resintonia de controladores industriais, *Dissertação de mestrado*, Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal, RN.
- Fleming, P. J. e Purshouse, R. C. (2002). Evolutionary algorithms in control systems engineering: a survey, *Control Engineering Practice*, **10**(11): 1223-1241.
- Ghoshal, S. P. (2004). Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control, *Electric Power Systems Research*, **72**(3): 203-212.
- Kennedy, J. e Eberhart, R. C. (1995). Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*. Perth, Australia, pp. 1942-1948.
- Kennedy, J. F., Eberhart, R. C. e Shi, Y. (2001). *Swarm intelligence*, Morgan Kaufmann Pub, San Francisco, CA, USA.
- Mital, D. P. e Chin, L. (1995). Intelligent control applications with neural networks, *Intelligent control systems: theory and applications*, Gupta, M. M. e Sinha, N. K. (eds), chapter 18, Piscataway: IEEE Press, pp. 479-514.
- Ratnaweera, A., Halgamuge, S. K. e Watson, H. C. (1996). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration, *IEEE Transactions on Evolutionary Computation*, **8**(3) 240-255.
- Shi, Y. e Eberhart, R.C. (2000). Experimental study of particle swarm optimization, *Proceedings of 4th World Conference on Systems, Cybernetics and Informatics*, Orlando, FL, USA.
- Wang, L., Barnes, T. J. D. e Cluett, W. R. (1995). New Frequency-domain design method for PID controllers, *IEE Proc.- Control Theory Appl.*, **142**(4): 265-271.
- Xiolan, W., Jianghui, D. e Debao, C. (2002). PID self-tuning control based on evolutionary programming, *Proceedings of the 4th World Congress on Intelligent Control and Automation*, Shanghai, P. R. China, pp. 3132-3135.
- Ziegler, J. G. e Nichols, N. B. (1942). Optimum settings for automatic controllers, *ASME Transaction*, **64**: 759-768.