

Desempenho de Seguimento de Trajetória no Espaço da Tarefa de Robôs Manipuladores: Um Projeto de Controlador Neural Adaptativo

Nardênio Almeida Martins, Maycol de Alencar, and Francisco Alexandre Ribeiro de Alencar, Douglas Wildgrube Bertol, Marcos Rafael Kaissi Barbosa

Laboratório de Eletrônica Digital e Hardware
Departamento de Informática, Universidade Estadual de Maringá
Avenida Colombo, 5790, 87020-900, Maringá, Paraná, Brasil
E-mail: {namartin,kaissi}@din.uem.br

Abstract - An Adaptive neural network control design for robot manipulators in task space coordinates is provided in this paper. This controller design and an direct adaptive control strategy (passivity-based controller) are simulated for the same trajectory, considering the presence of the friction torques and the influence of payload. Performances are evaluated according to behavior of position tracking, and to trajectory tracking accuracy. The adaptive neural network controller is developed based on a neural network modeling technique which neither requires the evaluation of inverse dynamical model nor the time-consuming training process, and does not require the inverse of the Jacobian matrix.

Index Terms - Adaptive Control, Neural Networks, Robot Manipulators, Task Space, Friction Torques.

Resumo - Um projeto de controle neural adaptativo para robôs manipuladores no espaço da tarefa é fornecido neste artigo. Este projeto de controlador e uma estratégia de controle adaptativo direto (controlador baseado na passividade) são simulados para a mesma trajetória, considerando a presença de torques de atrito e a influência de carga. Desempenhos são avaliadas de acordo com comportamento de seguimento de posição, e a exatidão do seguimento da trajetória. O controlador neural adaptativo é desenvolvido com base em uma técnica de modelagem de rede neural que não requer a avaliação do modelo dinâmico inverso nem o tempo de consumo para o processo de treinamento, e não requer a inversa da matriz Jacobiana.

Palavras-Chave - Controle Adaptativo, Redes Neurais, Robôs Manipuladores, Espaço da Tarefa, Torques de Atrito.

I. INTRODUÇÃO

Várias técnicas de controle foram desenvolvidas durante as últimas décadas para executar o seguimento de trajetória com precisão e desempenho satisfatórios. As técnicas clássicas de controle direto (PD e PID), empregadas no controle de seguimento de trajetória, exibem um erro mínimo limitado, devido ao fato de que não fazem a análise dos sistemas não-lineares típicos da dinâmica do robô manipulador [1]. Para resolver estes problemas, as técnicas de controle adaptativo direto para os robôs manipuladores foram desenvolvidas [2], as quais não requerem um conhecimento a priori de parâmetros desconhecidos.

Diversos resultados de controle adaptativo globalmente convergente para os robôs manipuladores rígidos apareceram [3], os

quais são descritos em [2] e divididos em dois grupos distintos. O primeiro grupo é uma versão adaptativa da dinâmica inversa ou do esquema de controle de torque computado que tenta linearizar e desacoplar globalmente as equações do movimento do robô. O segundo grupo confia fundamental na passividade da dinâmica rígida do robô e não tenta linearizar as equações do movimento. Como em todo o esquema de controle, existem vantagens, desvantagens e os compromissos nestes esquemas de controle.

Os pesquisadores têm sugerido o uso de redes neurais para o controle de robôs manipuladores [4]. Como regra geral, os projetos de controle empregam uma rede neural para empreender uma aproximação off-line suficientemente exata do modelo dinâmico do sistema e, para determinar conseqüentemente uma estratégia de controle adequada como de tal aproximação. Entretanto, estas aproximações não manipulam e não tratam corretamente quando das mudanças no sistema, incorporando conseqüentemente o uso do controle adaptativo. Algumas experiências dão bons resultados com a aplicação do controle adaptativo e de uma rede neural adequada para padronizar diretamente a regra de controle. Estes controladores direcionam-se com grande estabilidade para um sistema global em uma malha fechada, não requerendo a avaliação do modelo dinâmico inverso e nem o tempo necessário para o processo de treinamento [5, 6, 7, 8]. Neste caso, as redes neurais podem apenas ser inicializadas em zero, presumindo nenhum conhecimento do sistema.

Em muitas aplicações práticas, o movimento do robô é basicamente definido pelo movimento de seu efetuador final, isto é, no controle do robô o maior interesse é que o efetuador final siga o seu movimento desejado, definida no espaço da tarefa, ou no espaço cartesiano. Assim, muitos controladores foram sugeridos para o seguimento adaptativo de robôs manipuladores no espaço da tarefa, os quais requerem a inversa da matriz Jacobiana [9]. Entretanto, isto consome tempo e dificulta totalmente a obtenção da inversa da matriz Jacobiana. Além disso, as dificuldades devem-se às singularidades cinemáticas. Parametrizando diretamente a lei de controle, elimina-se a necessidade dessa inversão.

Neste artigo, o método de controle apresentado em [10] é estendido para o espaço da tarefa. Para aplicar o robô manipulador em uma ampla gama de classes de tarefas, é necessário controlar, não somente a posição do efetuador

final, mas também a força exercida por ele em um determinado objeto. Pelo projeto da lei de controle no espaço da tarefa, o controle de força pode ser facilmente formulado.

Na Seção II, a dinâmica e a modelagem da rede neural para robôs manipuladores no espaço da tarefa são analisados. Na Seção III o projeto do controlador é apresentado enquanto os resultados da simulação numérica são exibidos na Seção IV. Finalmente, na Seção V a conclusão é dada.

II. DINÂMICA E MODELAGEM DE REDES NEURAIS PARA ROBÔS MANIPULADORES

Usualmente, a especificação da tarefa do manipulador é dada relativamente ao efetuador final. Isto é natural na tentativa de derivar, preferencialmente, o algoritmo de controle diretamente no espaço da tarefa, do que no espaço de juntas. Assim, denotando a posição e a orientação do efetuador final no espaço da tarefa por $x \in \mathfrak{R}^n$. A dinâmica do espaço da tarefa pode ser escrita como

$$H_x(q)\ddot{x} + C_x(q, \dot{q})\dot{x} + G_x(q) + F_x(\dot{q}) = F \quad (1)$$

onde $q \in R^n$, $F \in R^n$, $H_x(q) \in R^{n \times n}$, $C_x(q, \dot{q}) \in R^{n \times n}$, $G_x(q) \in R^n$, e $F_x(\dot{q}) \in R^n$ são definidas em [11].

A equação dinâmica (1) tem as seguintes propriedades estruturais necessárias para a compreensão dos controladores diretos que têm que ser analisados.

Propriedade 1: Matriz de inércia $H_x(q)$ é simétrica e positiva definida.

Propriedade 2: Matriz $N = \dot{H}_x(q) - 2C_x(q, \dot{q})$ é anti-simétrica, se $C(q, \dot{q})$ é definida pelos símbolos de *Christoffel* [11].

$H_x(q)$, $G_x(q)$ e $F_x(\dot{q})$ são funções de q e \dot{q} , respectivamente, assim, redes neurais estáticas são suficientes para modelagem delas. Assumindo que $h_{xkj}(q)$, $g_{xk}(q)$ e $f_{xk}(\dot{q})$ podem ser modeladas como

$$\begin{aligned} h_{xkj}(q) &= \sum_l \theta_{kjl} \xi_{kjl}(q) + \varepsilon_{hkj}(q) = \theta_{kj}^T \xi_{kj}(q) + \varepsilon_{hkj}(q) \\ g_{xk}(q) &= \sum_l \beta_{kl} \eta_{kl}(q) + \varepsilon_{gk}(q) = \beta_k^T \eta_k(q) + \varepsilon_{gk}(q) \\ f_{xk}(\dot{q}) &= \sum_l \gamma_{kl} \rho_{kl}(\dot{q}) + \varepsilon_{fk}(\dot{q}) = \gamma_k^T \rho_k(\dot{q}) + \varepsilon_{fk}(\dot{q}) \end{aligned} \quad (2)$$

onde $\theta_{kjl}, \beta_{kl}, \gamma_{kl} \in R$ são pêsos das redes neurais; $\xi_{kjl}(q), \eta_{kl}(q), \rho_{kl}(\dot{q}) \in R$ são funções das bases Gaussianas com vetores de entrada q e \dot{q} , respectivamente, assim como $\varepsilon_{hkj}(q), \varepsilon_{gk}(q), \varepsilon_{fk}(\dot{q}) \in R$ são erros de modelagem de $h_{xkj}(q)$, $g_{xk}(q)$ e $f_{xk}(\dot{q})$, respectivamente, e são assumidos serem limitados. Sabendo que $C_x(q, \dot{q})$, é uma rede neural dinâmica devido ser uma função de q e \dot{q} , sua modelagem é requerida. Assumindo que $c_{xkj}(q, \dot{q})$ pode ser modelada como

$$\begin{aligned} c_{xkj}(q, \dot{q}) &= \sum_l \alpha_{kjl} \zeta_{kjl}(z) + \varepsilon_{ckj}(z) \\ &= \alpha_{kj}^T \zeta_{kj}(z) + \varepsilon_{ckj}(z) \end{aligned} \quad (3)$$

onde $z = [q^T \ \dot{q}^T]^T \in R^{2n}$, $\alpha_{kjl} \in R$ é o vetor de pêsos; $\zeta_{kjl}(z) \in R$ é a função da base Gaussianas com vetor de entrada z , e $\varepsilon_{ckj}(z)$ é o erro de modelagem de $c_{xkj}(q, \dot{q})$, o qual é também assumido ser limitado.

Em (1), a dinâmica do robô manipulador no espaço da tarefa pode ser descrito como

$$\begin{aligned} h_{xkj}(q) &= \theta_{kj}^T \xi_{kj}(q) + \varepsilon_{hkj}(q) \\ c_{xkj}(q, \dot{q}) &= \alpha_{kj}^T \zeta_{kj}(z) + \varepsilon_{ckj}(z) \\ g_{xk}(q) &= \beta_k^T \eta_k(q) + \varepsilon_{gk}(q) \\ f_{xk}(\dot{q}) &= \gamma_k^T \rho_k(\dot{q}) + \varepsilon_{fk}(\dot{q}) \end{aligned} \quad (4)$$

portanto, a estabilidade das redes neurais pode ser analisadas, na qual a matriz Ge-Lee (GL) [12], definida por $\{\cdot\}$ e seu operador de produto “ \bullet ” é empregada, e a matriz ordinária e o vetor são denotados por $[\cdot]$.

Pode-se, assim, expressar $H_x(q)$ como

$$H_x(q) = \left[\{\Theta\}^T \bullet \{\Xi(q)\} \right] + E_H(q) \quad (5)$$

onde $\{\Theta\}$ e $\{\Xi(q)\}$ são matrizes GL e seus elementos θ_{kj} e $\xi_{kj}(q)$, respectivamente. $E_H(q) \in R^{n \times n}$ é a matriz e seus elementos são erros de modelagem $\varepsilon_{hkj}(q)$. Similarmente, a expressão para $C_x(q, \dot{q})$, $G_x(q)$ e $F_x(\dot{q})$ é

$$C_x(q, \dot{q}) = \left[\{A\}^T \bullet \{Z(z)\} \right] + E_C(z) \quad (6)$$

$$G_x(q) = \left[\{B\}^T \bullet \{D(q)\} \right] + E_G(q) \quad (7)$$

$$F_x(\dot{q}) = \left[\{W\}^T \bullet \{P(\dot{q})\} \right] + E_F(\dot{q}) \quad (8)$$

onde $\{A\}$ e $\{Z(z)\}$ são matrizes GL, enquanto $\{B\}$, $\{W\}$, $\{D(q)\}$ e $\{P(\dot{q})\}$ são vetores GL, com seus respectivos elementos α_{kj} , $\zeta_{kj}(z)$, β_k , $\eta_k(q)$, γ_k , e $\rho_k(\dot{q})$. $E_C(z) \in R^{n \times n}$, $E_G(q) \in R^n$ e $E_F(\dot{q}) \in R^n$ são matriz e vetores, com seus elementos (erros de modelagem) $\varepsilon_{ckj}(z)$, $\varepsilon_{gk}(q)$ e $\varepsilon_{fk}(\dot{q})$, respectivamente.

III. PROJETO DO CONTROLADOR DA REDE NEURAL

Assumindo que $x_d(t)$ é a trajetória desejada nas coordenadas do espaço de tarefa, e $\dot{x}_d(t)$ e $\ddot{x}_d(t)$ são a velocidade e aceleração, respectivamente; as seguintes expressões são definidas

$$\begin{aligned} e(t) &= x_d(t) - x(t) \\ \dot{x}_r(t) &= \dot{x}_d(t) + \Lambda e(t) \\ \ddot{x}_r(t) &= \ddot{x}_d(t) + \Lambda \dot{e}(t) \\ r(t) &= \dot{x}_r(t) - \dot{x}(t) = \dot{e}(t) + \Lambda e(t) \end{aligned} \quad (9)$$

onde Λ é uma matriz diagonal positiva definida. Por meio do seguinte lema, a estabilidade de e e \dot{e} pode ser definida por r .

Lema 1: Assumindo que $e(t) = h(t) * r(t)$, no qual $*$ denota o produto de convolução e $h(t) = L^{-1}(H(s))$ mais $H(s)$ é uma função de transferência $n \times n$ exponencialmente estável e estritamente própria. Conseqüentemente, $r \in L_2^n$ o que implica que $e \in L_2^n \cap L_\infty^n$, $\dot{e} \in L_2^n$, e é contínuo e $e \rightarrow 0$ quando $t \rightarrow \infty$. Ainda mais, se $r \rightarrow 0$ quando $t \rightarrow \infty$, então $\dot{e} \rightarrow 0$ [13].

Denotando a estimativa de (\cdot) por $(\hat{\cdot})$, e definindo $(\tilde{\cdot}) = (\hat{\cdot}) - (\cdot)$ como o vetor de erro nos parâmetros. Conseqüentemente, $\{\hat{\Theta}\}$, $\{\hat{A}\}$, $\{\hat{B}\}$ e $\{\hat{W}\}$ representam as estimativas dos parâmetros verdadeiros das matrizes $\{\Theta\}$, $\{A\}$, $\{B\}$ e $\{W\}$ de (5) a (8), respectivamente.

Analisando o controlador no espaço da tarefa, tem-se

$$\begin{aligned} F = & \left[\{\hat{\Theta}\}^T \bullet \{\Xi(q)\} \right] \ddot{x}_r + \left[\{\hat{A}\}^T \bullet \{Z(z)\} \right] \dot{x}_r \\ & + \left[\{\hat{B}\}^T \bullet \{D(q)\} \right] + \left[\{\hat{W}\}^T \bullet \{P(\dot{q})\} \right] \\ & + (K_v + K_a \|e\|^2) r + K_p e + K_s \operatorname{sgn}(r) \end{aligned} \quad (10)$$

onde $K_v \in R^{n \times n} > 0$, $K_a \in R^{n \times n} > 0$, $\|e\|^2 = \sum_{i=1}^g e_i^2$, com g sendo o grau de liberdade do robô manipulador, $K_p \in R^{n \times n} > 0$ e $K_s > \|E\|$ com $E = E_H(q) \ddot{x}_r + E_C(z) \dot{x}_r + E_G(q) + E_F(\dot{q})$.

De (10), fica claro que o controlador não requer a inversa da matriz Jacobiana $J(q)$, a qual é uma característica muito atrativa sabendo do problema da singularidade cinemática, onde $J^{-1}(q)$ não existe, é evitado.

Se a lei de controle (10) é aplicada na dinâmica do robô manipulador, em (1) e, se (9) é usado, a equação dinâmica de erro fica

$$\begin{aligned} H_x(q) \dot{r} + C_x(q, \dot{q}) r + (K_v + K_a \|e\|^2) r \\ + K_p e + K_s \operatorname{sgn}(r) = & \left[\{\tilde{\Theta}\}^T \bullet \{\Xi(q)\} \right] \ddot{x}_r \\ & + \left[\{\tilde{A}\}^T \bullet \{Z(z)\} \right] \dot{x}_r + \left[\{\tilde{B}\}^T \bullet \{D(q)\} \right] \\ & + \left[\{\tilde{W}\}^T \bullet \{P(\dot{q})\} \right] + E. \end{aligned} \quad (11)$$

A estabilidade do sistema (11) é dada pelo seguinte teorema.

Teorema 1: Para o sistema em malha fechada (11), se $K_v > 0$, $K_a > 0$, $K_p > 0$, $K_s > \|E\|$ e os parâmetros são atualizados por

$$\dot{\hat{\theta}}_k = \Gamma_k \bullet \{\xi_k(q)\} \ddot{x}_r r_k \quad (12)$$

$$\dot{\hat{\alpha}}_k = Q_k \bullet \{\zeta_k(z)\} \dot{x}_r r_k \quad (13)$$

$$\dot{\hat{\beta}}_k = N_k \eta_k(q) r_k \quad (14)$$

$$\dot{\hat{\gamma}}_k = M_k \rho_k(\dot{q}) r_k \quad (15)$$

onde $\Gamma_k = \Gamma_k^T > 0$, $Q_k = Q_k^T > 0$, $N_k = N_k^T > 0$ e $M_k = M_k^T > 0$, $\hat{\theta}_k$ e $\hat{\alpha}_k$ são os vetores coluna com seus respectivos elementos $\hat{\theta}_{kj}$ e $\hat{\alpha}_{kj}$, então $\hat{\theta}_k$, $\hat{\alpha}_k$, $\hat{\beta}_k$, $\hat{\gamma}_k \in L_\infty$ e $e \in L_2^n \cap L_\infty^n$, e é contínuo, e e $\dot{e} \rightarrow 0$ quando $t \rightarrow \infty$.

Prova: Considere a função escalar positiva V como

$$\begin{aligned} V = & \frac{1}{2} r^T H_x(q) r + \frac{1}{2} e^T K_p e + \frac{1}{2} \sum_{k=1}^n \tilde{\theta}_k^T \Gamma_k^{-1} \tilde{\theta}_k \\ & + \frac{1}{2} \sum_{k=1}^n \tilde{\alpha}_k^T Q_k^{-1} \tilde{\alpha}_k + \frac{1}{2} \sum_{k=1}^n \tilde{\beta}_k^T N_k^{-1} \tilde{\beta}_k \\ & + \frac{1}{2} \sum_{k=1}^n \tilde{\gamma}_k^T M_k^{-1} \tilde{\gamma}_k \end{aligned} \quad (16)$$

onde Γ_k , Q_k , N_k e M_k são matrizes simétricas e positivas definidas compatíveis. Se a derivada de (16) é calculada com (11), e as necessárias simplificações são feitas, tem-se

$$\begin{aligned} \dot{V} = & r^T (H_x(q) \dot{r} + C_x(q, \dot{q}) r) + e^T K_p \dot{e} \\ & + \sum_{k=1}^n \tilde{\theta}_k^T \Gamma_k^{-1} \dot{\tilde{\theta}}_k + \sum_{k=1}^n \tilde{\alpha}_k^T Q_k^{-1} \dot{\tilde{\alpha}}_k \\ & + \sum_{k=1}^n \tilde{\beta}_k^T N_k^{-1} \dot{\tilde{\beta}}_k + \sum_{k=1}^n \tilde{\gamma}_k^T M_k^{-1} \dot{\tilde{\gamma}}_k \end{aligned} \quad (17)$$

na qual a propriedade anti-simetria é usada. Substituindo a dinâmica de erro (11) e observando que

$$\begin{aligned} r^T \left[\{\tilde{\Theta}\}^T \bullet \{\Xi(q)\} \right] \ddot{x}_r &= \sum_{k=1}^n \{\tilde{\theta}_k\}^T \bullet \{\xi_k(q)\} \ddot{x}_r r_k \\ r^T \left[\{\tilde{A}\}^T \bullet \{Z(z)\} \right] \dot{x}_r &= \sum_{k=1}^n \{\tilde{\alpha}_k\}^T \bullet \{\zeta_k(z)\} \dot{x}_r r_k \\ r^T \left[\{\tilde{B}\}^T \bullet \{D(q)\} \right] &= \sum_{k=1}^n \tilde{\beta}_k^T \eta_k(q) r_k \\ r^T \left[\{\tilde{W}\}^T \bullet \{P(\dot{q})\} \right] &= \sum_{k=1}^n \tilde{\gamma}_k^T \rho_k(\dot{q}) r_k \end{aligned} \quad (18)$$

a (17) torna-se

$$\begin{aligned} \dot{V} = & -r^T (K_v + K_a \|e\|^2) r - e^T K_p \Lambda e \\ & + \sum_{k=1}^n \{\tilde{\theta}_k\}^T \bullet \{\xi_k(q)\} \ddot{x}_r r_k + \sum_{k=1}^n \{\tilde{\alpha}_k\}^T \bullet \{\zeta_k(z)\} \dot{x}_r r_k \\ & + \sum_{k=1}^n \tilde{\beta}_k^T \eta_k(q) r_k + \sum_{k=1}^n \tilde{\gamma}_k^T \rho_k(\dot{q}) r_k + \sum_{k=1}^n \tilde{\theta}_k^T \Gamma_k^{-1} \dot{\tilde{\theta}}_k \\ & + \sum_{k=1}^n \tilde{\alpha}_k^T Q_k^{-1} \dot{\tilde{\alpha}}_k + \sum_{k=1}^n \tilde{\beta}_k^T N_k^{-1} \dot{\tilde{\beta}}_k + \sum_{k=1}^n \tilde{\gamma}_k^T M_k^{-1} \dot{\tilde{\gamma}}_k \end{aligned} \quad (19)$$

Substituindo a lei de atualização de parâmetro dada em (12) por (15) em (19), através de $K_s > \|E\|$, tem-se

$$\dot{V} \leq -r^T (K_v + K_a \|e\|^2) r - e^T K_p \Lambda e \leq 0. \quad (20)$$

a) De (20) e, sabendo que $K_v > 0$, $K_a > 0$, $K_p > 0$ e $\Lambda > 0$, tem-se $r \in L_2^n$. Conseqüentemente, a partir do *Lema* 1, $e \in L_2^n \cap L_\infty^n$, e é contínuo e $e \rightarrow 0$ quando $t \rightarrow \infty$, e $\dot{e} \in L_2^n$.

b) De $\dot{V} \leq -r^T(K_v + K_a\|e\|^2)r - e^T K_p \Lambda e \leq 0$, conseqüentemente $0 \leq V(t) \leq V(0)$, $\forall t \geq 0$. Assim, de $V(t) \in L_\infty$, isto implica que r , $\tilde{\theta}_k$, $\tilde{\alpha}_k$, $\tilde{\beta}_k$, $\tilde{\gamma}_k \in L_\infty$, ou melhor, $\hat{\theta}_k$, $\hat{\alpha}_k$, $\hat{\beta}_k$, $\hat{\gamma}_k \in L_\infty$.

Sabendo que $r \in L_2^n$, x_d , \dot{x}_d e $\ddot{x}_d \in L_\infty^n$, e $\{\Xi(q)\}$, $\{Z(z)\}$, $\{D(q)\}$, e $\{P(\dot{q})\}$ são funções de base limitadas, pode-se concluir que $\dot{r} \in L_\infty^n$ a partir de (11). Assim, r é uniformemente contínuo. A prova é definida quando o seguinte é considerado [13]: r é uniformemente contínuo, $r \in L_2^n \Rightarrow r \rightarrow 0$ quando $t \rightarrow \infty$. Conseqüentemente, $\dot{e} \rightarrow 0$ quando $t \rightarrow \infty$.

IV. RESULTADOS DE SIMULAÇÕES

O objetivo das simulações é conduzido para demonstrar o desempenho do controlador sugerido para robôs manipuladores no espaço da tarefa.

Para o teste de simulação é usado um robô manipulador planar de dois elos, considerando a presença de torques de atrito [14].

Para esse robô, a matriz Jacobiana é dada por

$$J(q) = \begin{bmatrix} -l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{bmatrix}$$

onde os comprimentos dos elos 1 e 2 são 0.4318 m e 0.6491 m, respectivamente.

As trajetórias desejadas no espaço cartesiano são escolhidas como

$$\begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix} = \begin{bmatrix} 0.35 + 0.15 \cos(\pi t) \\ 0.35 + 0.15 \sin(\pi t) \end{bmatrix}$$

onde representam um círculo de raio 0,15 m e seu centro localizado em $(x, y) = (0.35, 0.35)$ m.

O robô é inicialmente assumido estar parado com seu efetuador final posicionado nas seguintes coordenadas do círculo, i.e.

$$[x(0) \ y(0)] = [0.5 \ 0.35]; \quad [\dot{x}(0) \ \dot{y}(0)] = [0.0 \ 0.47].$$

Para o teste de rejeição à perturbação de carga, uma carga adicional de $p_l = 20.0$ kg é aplicada em um tempo $t = 1.0$ s.

Para comparação, um controlador adaptativo padrão é dado a seguir.

Controlador baseado na Passividade de Slotine e Li: O problema do projeto de controle adaptativo [apud 2, 15] pode ser facilmente exemplificado através das leis de controle e adaptação:

$$\begin{aligned} F &= \hat{H}(q)\ddot{x}_r + \hat{C}(q, \dot{q})\dot{x}_r + \hat{G}(q) - K_v r \\ &= \bar{Y}(q, \dot{q}, \ddot{x}_r)\hat{\theta} - K_v r \\ \hat{\phi} &= -\Gamma_{pc}^{-T} \bar{Y}^T(q, \dot{q}, \ddot{x}_r)r \end{aligned}$$

com $\Gamma_{pc} = \Gamma_{pc}^T > 0$ uma matriz de parâmetros de projeto, $\bar{Y}(q, \dot{q}, \ddot{x}_r)$ uma matriz razoavelmente complicada de funções do robô que deve ser explicitamente derivada da dinâmica do robô manipulador, e ϕ o vetor de parâmetros desconhecidos. Este vetor está definido em [14]:

$$\phi = [a_{01} \ a_{02} \ a_{03} \ a_{04} \ a_{05} \ a_{06} \ a_{07} \ a_{08} \ a_{09}]^T$$

Os ganhos das leis de controle e adaptação são escolhidas respectivamente como $K_v = \text{diag}[100]$; $\Lambda = \text{diag}[10]$; $\Gamma_{pc} = \text{diag}[20]$. A resposta deste controlador é exibida na Fig. 1.

A entrada (2,9) da matriz de função do robô $\bar{Y}(q, \dot{q}, \ddot{x}_r)$ é:

$$p_2 [[1 + \cos(q_2)] \ddot{q}_{r1} + \ddot{q}_{r2} + [\sin(q_2) \dot{q}_1 \dot{q}_{r1}]] + p_5 [\cos(q_1 + q_2)]$$

onde p_2 e p_5 são parâmetros definidos em [14]; \dot{q}_r e \ddot{q}_r variáveis definidas no espaço da tarefa com base na matriz Jacobiana.

Para demonstrar o efeito insatisfatório das dinâmicas não-modeladas no controle adaptativo, o termo $p_5 [\cos(q_1 + q_2)]$ é retirado do controlador. Os resultados aparecem na Fig. 2. É enfatizado que, no controlador neural adaptativo, todas as dinâmicas são não-modeladas.

Controlador Neural Adaptativo: Para cada elemento de $\hat{H}_x(q)$, $\hat{G}_x(q)$, e $\hat{F}_x(\dot{q})$, uma rede neural de função de base radial Gaussiana (FBR) com entrada x e \dot{x} , respectivamente, é usada, visto que, para cada elemento de $\hat{C}_x(q, \dot{q})$, uma rede neural FBR Gaussiana com ambas as entradas x e \dot{x} é escolhida. Simplificando, os centros μ localizados das RBF Gaussianas são distribuídas uniformemente para estender o espaço de entrada da rede. A variância σ^2 da função Gaussiana é fixada em 10.0. Os pesos da rede neural são todos inicializados em zero, assumindo nenhum conhecimento sobre o sistema e nenhuma aplicação de carga (sem carga adicional).

Os ganhos da porção PD da lei de controle foi escolhida como $K_v = K_a = K_p = \text{diag}[100]$; $\Lambda = \text{diag}[10]$.

O termo de controle de modo deslizante é excluído ajustando $K_s = \text{diag}[0]$, mostrando assim a efetividade do controlador neural.

Para o caso do controle não-adaptativo, a lei de adaptação dos pesos do controlador neural não é ativada. A Fig. 3 mostra o desempenho do seguimento de posição do robô e a precisão do seguimento de trajetória. Pode ser observado, a partir desses resultados, que o controlador neural não-adaptativo tem um desempenho insatisfatório.

Para o caso do controle adaptativo, os algoritmos de adaptação são ativados com $\Gamma_k = \text{diag}[5.0]$, $Q_k = \text{diag}[1.0]$,

$N_k = \text{diag}[20.0]$, e $M_k = \text{diag}[20.0]$. O desempenho do seguimento de posição e a precisão do seguimento de trajetória do robô é mostrado na Fig. 4. Pode ser visto que o desempenho é satisfatório em relação ao caso não-adaptativo, por causa do mecanismo de aprendizagem. A comparação com o desempenho do controle adaptativo padrão nas Figuras 1 e 2 é visível, mesmo que a dinâmica do robô manipulador não seja requerida para implementar o controlador neural adaptativo.

V. CONCLUSÕES

Um projeto de controle neural adaptativo de robôs manipuladores no espaço de tarefa é apresentado neste artigo.

O projeto do controlador estendido ao espaço da tarefa é baseado no método de controle apresentado em [10].

Um comportamento satisfatório no seguimento da trajetória é obtido dos resultados de simulação deste controlador sugerido, considerando a presença de torques de atrito.

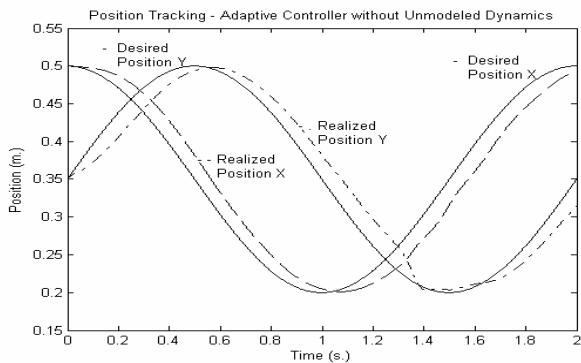
Mostra-se, através destas simulações, como a implementação do controle neural adaptativo sugeriria foi conseguida particionando-se a rede neural em várias subredes pequenas para obter uma computação mais eficiente. A principal vantagem desta abordagem de rede neural é que permite dividir o controlador em termos de redes neurais particionadas ou subredes neurais. Isto simplifica o projeto, dá uma estrutura do controlador adicionada, e também direciona para contribuir em ajustes mais rápidos dos algoritmos de adaptação. No exemplo do controle neural adaptativo sugerido, este procedimento resultou em quatro subredes neurais, que foi descrita como uma rede neural estruturada ou particionada. Mostrou-se diretamente que as redes neural individuais particionadas podem ser separadamente ajustadas, contribuindo para um procedimento rápido de atualização de pesos. Outras vantagens desta rede neural estruturada é que se alguns termos na dinâmica do robô são bem conhecidas (por exemplo, a matriz de inércia $H_x(q)$ e a matriz de torques gravitacionais $G_x(q)$), então as suas respectivas redes neurais podem ser substituídas por equações que as computem. Estas redes neurais podem ser usadas para reconstruir somente os termos desconhecidos ou aqueles muitos complicados para ser computados, os quais provavelmente incluem os termos da matriz de torques de atrito $F_x(\dot{q})$ e os termos da matriz de torques de Coriolis e centrípetos $C_x(q, \dot{q})$.

Pode-se também concluir que o controlador proposto é uma alternativa no seguimento de trajetória onde um alto

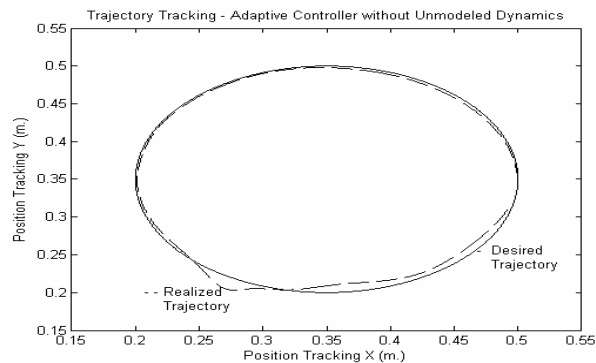
grau de desempenho e precisão são requeridos, sendo que não requer a avaliação do modelo dinâmico inverso nem o tempo necessário para o processo de treinamento. Adicionalmente, eliminando a necessidade para cálculo da inversa da matriz Jacobiana, é possível controlar efetivamente os sistemas dinâmicos não-lineares desconhecidos, bem como as perturbações de carga.

REFERÊNCIAS

- [1] L. Sciavicco, and B. Siciliano, *Modeling and Control of Robot Manipulators*, New York: The McGraw-Hill Companies Inc., 1996.
- [2] M. W. Spong, "Control of Robots and Manipulators", in *The Control Handbook*, CRC Press Inc., 1996, pp. 1339-1351.
- [3] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of Robot Manipulators*, New York: Maxwell Macmillan Publishing Company, 1993.
- [4] T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Trajectory Control of Robotic Manipulators Using Neural Networks", *IEEE Trans. On Ind. Electron.*, vol. 38, pp. 195-202, June 1991.
- [5] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural Net Robot Controller with Guaranteed Tracking Performance", *IEEE Trans. On Neural Networks*, vol. 6, pp. 703-715, May 1995.
- [6] F. L. Lewis, A. Yesildirek, and K. Liu, "Neural Net Robot Controller: Structure and Stability Proofs", *Journal Intelligent and Robotic Sys.*, vol. 12, pp. 277-299, 1995.
- [7] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer Neural Net Robot Controller: Structure and Stability Proofs", *IEEE Trans. On Neural Networks*, vol. 7, n.º 2, pp. 1-12, March 1996.
- [8] F. L. Lewis, A. Yesildirek, and K. Liu, "Multilayer Neural Net Robot Controller with Guaranteed Tracking Performance", *IEEE Trans. On Neural Networks*, vol. 7, n.º 2, pp. 388-399, March 1996.
- [9] J.-J. E. Slotine, and W. Li, "On the Adaptive Control of Robot Manipulators", *Int. Journal Robotics Research*, vol. 6, n.º 3, pp. 49-59, 1987.
- [10] N. A. Martins, M. de Alencar, "A Proposed Adaptive Neural Controller and Direct Controllers for Robot Manipulators", in *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, vol. 3, pp. 1174-1179, Japan, July 2003.
- [11] C. Canudas de Wit, B. Siciliano, and G. Bastin, *Theory of Robot Control*, New York: Springer-Verlag, January 1996.
- [12] S. S. Ge, "Robust Adaptive NN Feedback Linearization Control of Nonlinear Systems", *Int. Journal Systems Science*, vol. 27, n.º 12, pp. 1327-1338, 1996.
- [13] C. Desoer, and M. Vidyasagar, *Feedback Systems Input-Output Properties*, New York: Academic Press, 1975.
- [14] M. Erlic, and W.-S. Lu, "A Reduced-Order Adaptive Velocity Observer for Manipulator Control", *IEEE Trans. On Robotics and Automation*, vol. 11, pp. 293-303, April 1995.
- [15] J.-J. E. Slotine, and W. Li, "Adaptive Manipulator Control: A Case Study", *IEEE Trans. On Automatic Control*, vol. 33, n.º 11, pp. 995-1003, November 1988.

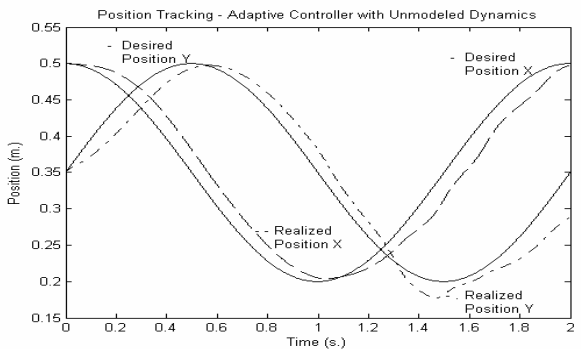


(a)

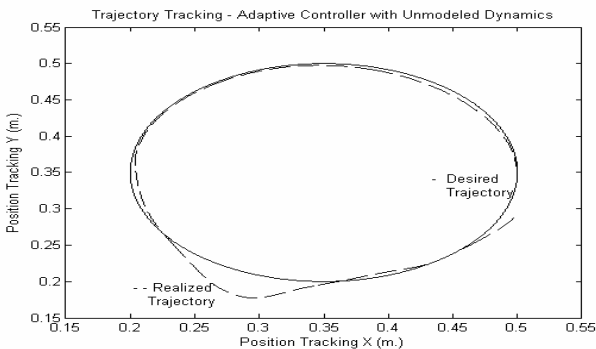


(b)

Fig. 1 Resposta do Controlador Adaptativo. (a) Posições Desejadas e Realizadas (b) Seguimentos de Trajetória Desejado e Realizado

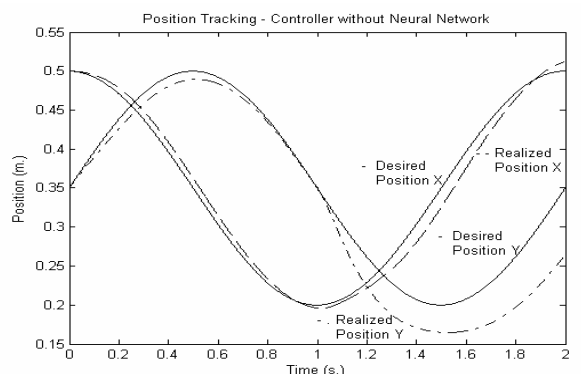


(a)

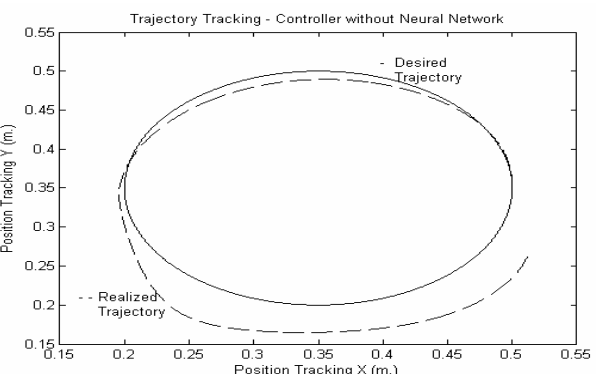


(b)

Fig. 2 Resposta do Controlador Adaptativo com Dinâmica Não-Modelada . (a) Posições Desejadas e Realizadas (b) Seguimentos de Trajetória Desejado e Realizado

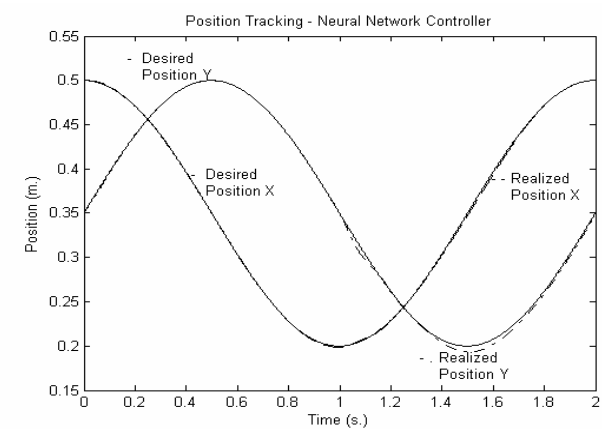


(a)

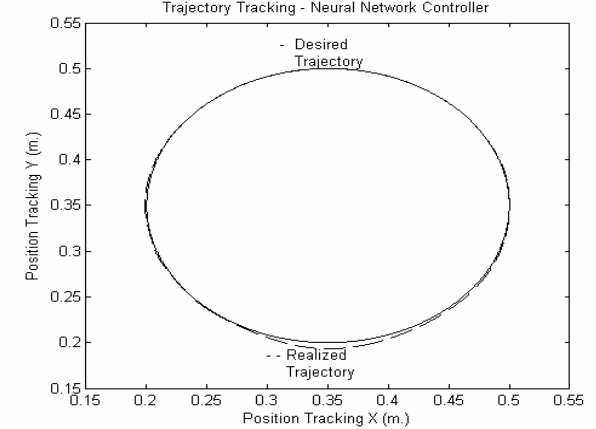


(b)

Fig. 3 Resposta do Controlador sem Rede Neural. (a) Posições Desejadas e Realizadas (b) Seguimentos de Trajetória Desejado e Realizado



(a)



(b)

Fig. 4 Resposta do Controlador Neural. (a) Posições Desejadas e Realizadas (b) Seguimentos de Trajetória Desejado e Realizado