



## Redes Neurais Artificiais para Modelagem Cinemática de Robôs Manipuladores

**Código de Aceite: AACBJG**

**Resumo:** *Um dentre os diversos problemas existentes atualmente na robótica é a obtenção de modelos matemáticos de robôs manipuladores. Isto deve-se principalmente à complexidade dos robôs, fato que dificulta a obtenção de alguns de seus parâmetros relativos tanto a modelos geométricos quanto a dinâmicos. Este trabalho insere-se na tentativa de diminuir estes problemas, consistindo em um estudo sobre a utilização de redes neurais artificiais para a representação dos modelos cinemáticos direto e inverso de um robô manipulador de três graus de liberdade. Os resultados mostraram que esta é uma boa técnica para a representação destes modelos. Observou-se ainda que a qualidade dos resultados depende fundamentalmente da estrutura da própria rede, bem como da estratégia de aprendizado desta.*

**Palavras-Chave:** *robô manipulador, modelos cinemáticos, redes neurais artificiais*

### 1. INTRODUÇÃO

As conhecidas Redes Neurais Artificiais (RNA) tiveram seu surgimento na década de 40, quando apareceram os trabalhos de McCulloch-Pitts e Hebb, dentre outros. A utilização de RNA em robótica é mais recente. Em Miller III *et al.* (1995) há uma descrição histórica do surgimento das RNA, estando inclusas aplicações em robótica. Experiência análoga é encontrada em Zalzalá *et al.* (1996), especificamente a partir da utilização de *Radial Basis Function Networks* e *MultiLayer Perceptron (MLP)*.

No presente artigo apresenta-se um método alternativo à modelagem cinemática de robôs manipuladores, baseado em uma RNA. A motivação para esta aplicação reside no fato de que, dependendo da complexidade geométrica do manipulador, a determinação de alguns parâmetros físicos é muito difícil. Como exemplo podem ser citados as distâncias entre eixos principais de inércia dos diversos corpos rígidos poli-articulados que compõem o manipulador e ainda, a determinação dos seus centros de massa. Imprecisões na determinação destes parâmetros dificultam principalmente a precisão de modelos cinemáticos, fato que pode comprometer toda uma estratégia de controle.

A estrutura de rede adotada neste trabalho foi a *MLP*. Utilizaram-se três camadas de neurônios (apenas uma intermediária, com 40 neurônios), diferente de Kosko (1992) que adotou duas camadas intermediárias. Objetivou-se com isso mostrar que uma estrutura mais simples poderia proporcionar bons resultados, tendo a vantagem de consumir um menor tempo de processamento durante a fase de aprendizado. Evidentemente, este tempo depende também da velocidade de convergência do algoritmo.

O artigo está dividido em seis seções. Na seção 2 define-se o problema a ser tratado, bem como os modelos do manipulador em estudo. A seção 3 mostra de forma resumida a estrutura de toda a pesquisa realizada que deu origem ao presente artigo. O treinamento da RNA, bem como detalhes da sua estrutura são apresentados na seção 4. Resultados de simulação e conclusões gerais são apresentados nas seções 5 e 6 respectivamente.

## 2. DEFINIÇÃO DO PROBLEMA

Os conhecidos modelos cinemáticos de robôs dividem-se em dois tipos: Modelo Cinemático Direto (MCD); Modelo Cinemático Inverso (MCI). Tome-se como exemplo o manipulador da fig.1, configuração muito utilizada industrialmente e que apresenta apenas juntas rotacionais. Desprezou-se o movimento rotacional da garra com o objetivo de se trabalhar com apenas três graus de liberdade. Os modelos cinemáticos são definidos a seguir.

- O MCD informa as posições  $x$ ,  $y$  e  $z$  (geralmente tomadas em um referencial inercial fixo à base do manipulador) do centro de massa da carga terminal ou garra, em função dos ângulos nas juntas do manipulador:

$$\begin{aligned} x &= f_1(\theta_1, \theta_2, \theta_3, p_i) \\ y &= f_2(\theta_1, \theta_2, \theta_3, p_i) \\ z &= f_3(\theta_1, \theta_2, \theta_3, p_i) \end{aligned} \tag{1}$$

- O MCI informa os ângulos nas juntas em função das posições  $x$ ,  $y$  e  $z$  da carga terminal ou garra:

$$\begin{aligned} \theta_1 &= g_1(x, y, z, p_i) \\ \theta_2 &= g_2(x, y, z, p_i) \\ \theta_3 &= g_3(x, y, z, p_i) \end{aligned} \tag{2}$$

Os  $n$  parâmetros geométricos tais como comprimento dos braços, *etc.*, estão representados por  $p_i$ ,  $i = 1, \dots, n$  na Eq.(1) e Eq.(2). Neste artigo concentrou-se o interesse na determinação de posições, embora se pudesse trabalhar também com velocidades.

Duas redes foram desenvolvidas, uma para o MCD e outra para o MCI. Os modelos representados pela Eq.(1) e pela Eq.(2) foram utilizados para a obtenção dos dados de treinamento das redes. Evidentemente, caso se dispusesse de um manipulador real, estes dados seriam fornecidos por sensores. Os modelos das equações (1) e (2) se aplicam ainda na verificação da eficiência das redes implementadas, uma vez que os resultados de simulação podem ser comparados aos oriundos dos modelos.

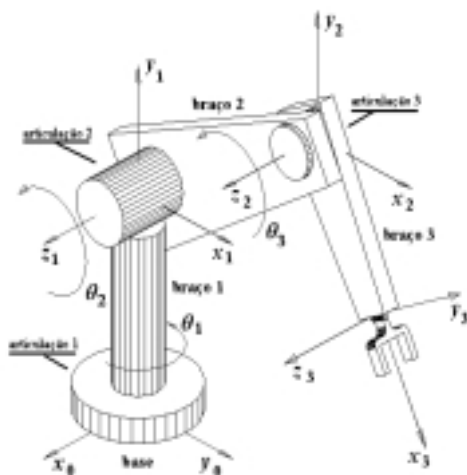


Figura 1 - Robô manipulador em estudo.

### 3. ESTRUTURA DA PESQUISA DESENVOLVIDA

A pesquisa aqui resumidamente exposta foi dividida em três etapas:

- Desenvolvimento dos modelos cinemáticos (direto e inverso);
- Desenvolvimento e implementação computacional dos algoritmos das redes, bem como o ensinamento destas (uma rede para cada modelo);
- Simulações.

Na primeira etapa foram desenvolvidos os modelos matemáticos, os quais foram utilizados para gerar dados oriundos de uma planta real supostamente desconhecida. A partir destes modelos geraram-se na segunda etapa as redes neurais e os valores que seriam ensinados a estas. Treinou-se então uma rede para modelar o MCD e outra para o MCI. Faz-se necessário colocar que foi utilizada a *Toolbox Neural Network* do software *MatLab* para a implementação das redes e dos algoritmos de treinamento. Na última etapa as redes foram submetidas a simulações, visando verificar a eficiência destas.

### 4. ARQUITETURA DA REDE NEURAL

Utilizou-se para a aquisição dos modelos cinemáticos do manipulador RNA com neurônios de estrutura similar<sup>1</sup> à da fig. 2,

<sup>1</sup> Exceto pela função de transferência

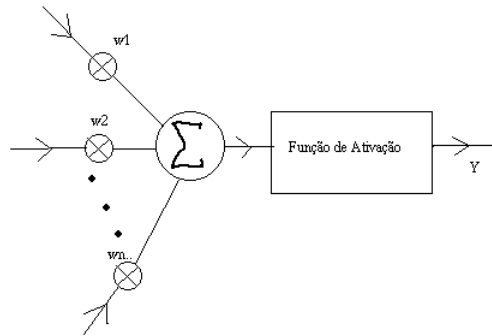


Figura 2 - Modelo de neurônio.

A camada de entrada da rede possui três neurônios, com função de ativação descrita pela equação seguinte:

$$f_x = \tanh(x) \quad (3)$$

Uma única camada de neurônios separa a entrada da saída da rede (1 camada intermediária), conforme ilustrado na fig. 3. Esta camada foi objeto de testes, variando-se o número de neurônios desta, buscando uma arquitetura que apresentasse capacidade de aprender o modelo, dentro de um tempo computacional minimizado. Simulações foram realizadas com 50 neurônios na camada interna.

Três neurônios constituem a camada de saída da rede. A mesma estrutura de rede foi aplicada tanto para o modelo direto quanto para o inverso.

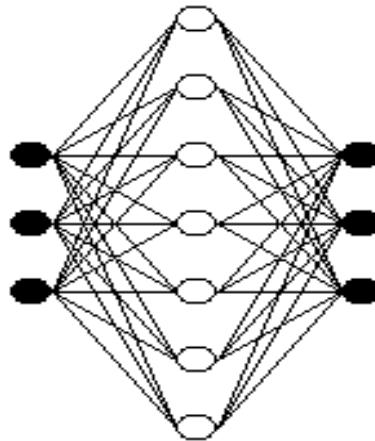


Figura 3 - Estrutura da rede neural aplicada

Como algoritmo de treinamento utilizou-se o *backpropagation*, apresentando variações da taxa de aprendizado e momento. Estas duas alterações ao *backpropagation* padrão são para obter-se melhores resultados na saída (momento) e fazer com que se alcance o mínimo erro mais rapidamente (taxa de aprendizado).

## 5. RESULTADOS OBTIDOS

Por questões de economia de espaço, apresentam-se apenas resultados obtidos com a rede representativa do MCD, ressaltando-se entretanto que estes são semelhantes aos resultados obtidos com a rede para o MCI. Antes de apresentar os resultados, é importante ressaltar qual a estratégia de aprendizado adotada, bem como quais foram as trajetórias de referência utilizadas para a realização das simulações, trajetórias estas em termos dos ângulos das juntas.

### 5.1 Estratégia de Aprendizado

Observando-se a fig. 1, percebe-se que  $\theta_1$  corresponde ao ângulo de rotação da base, enquanto que  $\theta_2$  e  $\theta_3$  correspondem aos ângulos do primeiro e segundo braço respectivamente. Propõe-se aqui uma estratégia de aprendizado que venha a permitir a utilização da rede de forma generalizada, ou seja, não restrita a um único tipo de trajetória de referência em termos dos ângulos das juntas. Fez-se variações nos ângulos das juntas e utilizou-se as saídas do MCD (eq. (1)) como referências para o aprendizado da rede. As variações adotadas foram as seguintes:

$$\theta_1 \Rightarrow [-\pi, \pi]; \quad \theta_2 \Rightarrow \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]; \quad \theta_3 \Rightarrow \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]; \quad (4)$$

sendo que para os três ângulos adotaram-se o mesmo incremento igual a  $0.1\pi$ . Para cada variação em  $\theta_1$ , fez-se uma varredura em  $\theta_2$  e para cada variação neste último, fez-se uma varredura em  $\theta_3$ , isto respeitando os intervalos acima.

### 5.2 Trajetórias de Referência para as Simulações

Para a realização de simulações, adotou-se duas trajetórias de referência em termos dos ângulos das juntas, descritas em Gomes *et al.* (1998). A primeira exponencial, conforme a equação seguinte:

$$\theta_r(t) = \left( \theta_f - \theta_i \right) \left( 1 - e^{-\left( \frac{b t^N}{(t_f + \varepsilon - t)^M} \right)} \right) + \theta_i \quad (5)$$

onde  $\theta_f$  é o ângulo final,  $\theta_i$  o inicial,  $t_f$  o tempo final (tempo inicial suposto nulo). Adotou-se ainda  $b = N = M = 1$  e  $\varepsilon = 0$ . A eq (5) é aplicada três vezes, para cada um dos ângulos do manipulador.

A segunda trajetória adotada é do tipo polinomial, sob a forma :

$$\theta_r(t) = a_1 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (6)$$

cujos coeficientes são obtidos a partir da solução do seguinte sistema:

$$\begin{bmatrix} 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 6t_f & 12t_f^2 & 20t_f^3 \\ t_f^3 & t_f^4 & t_f^5 \end{bmatrix} \begin{bmatrix} a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \theta_f - \theta_i \end{bmatrix} \quad (7)$$

As figuras 4 e 5 mostram estas trajetórias, para  $\theta_i = 0$ ,  $\theta_f = 60^\circ$  e  $t_f = 1s$ .



Figura 4 - Trajetória exponencial.

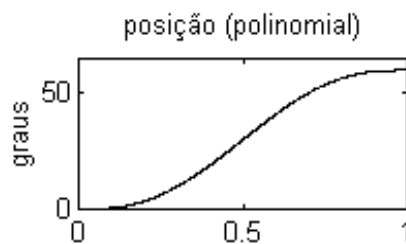


Figura 5 - Trajetória polinomial.

### 5.3 Simulações

Os resultados das simulações podem ser vistos nas figuras 6 a 9. Estas simulações foram realizadas tendo-se como dados de entrada na rede as trajetórias obtidas com as equações (5) e (6). Na fig. 6 vêem-se os erros nas coordenadas  $x$ ,  $y$  e  $z$ , obtidos a partir da diferença entre as referidas posições oriundas do modelo (eq. (1)) e as obtidas com a simulação da rede. Na fig. 7 vêem-se as trajetórias do modelo (linha pontilhada) e da rede (linha cheia) no espaço. A fig. 8 contém os erros em  $x$ ,  $y$  e  $z$ , desta feita com a trajetória exponencial, enquanto que a fig. 9 contém as respectivas trajetórias do modelo (linha pontilhada) e da rede (linha cheia) no espaço. Pode-se perceber que os erros são pequenos, mostrando assim que a rede aprendeu bem o MCD. O treinamento da rede efetuado foi genérico o suficiente para permitir a utilização desta em diferentes trajetórias dos ângulos das juntas. Este constitui um fato importante, haja visto que outras trajetórias foram testadas além das aqui mostradas exponencial e polinomial, ficando sempre os erros restritos aos mesmos valores mostrados nas últimas figuras. Os erros cresceram bastante quando a amplitude dos deslocamentos ( $\theta_f - \theta_i$  nas equações (5) e (7)) ultrapassava os limites impostos aos ângulos

das juntas (eq. (4)) na fase de aprendizado da rede, fato já esperado e apenas confirmado em outras simulações realizadas aqui omitidas por razões de espaço.

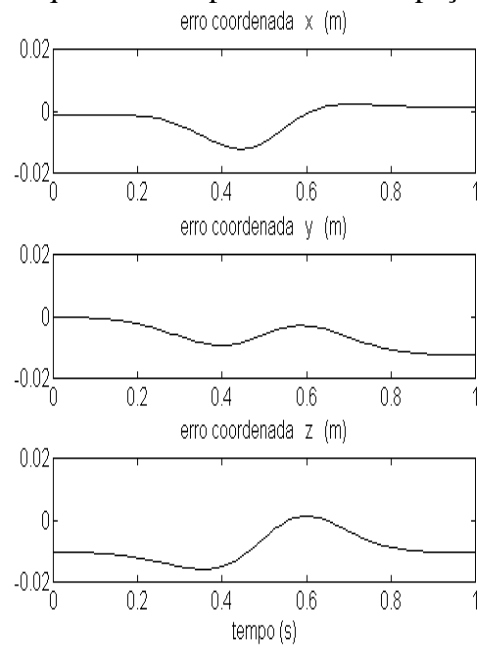


Figura 6 - Curvas de erro para trajetória polinomial.

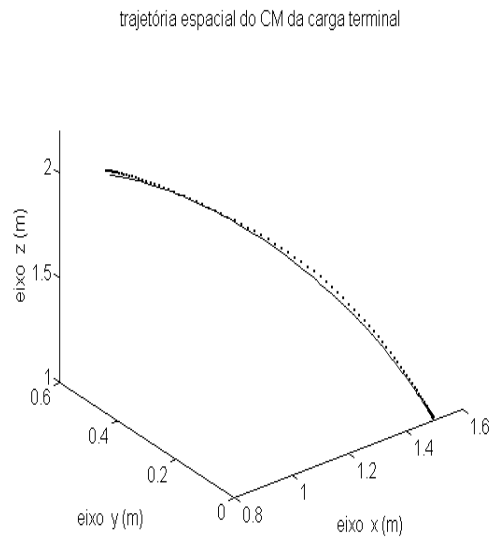


Figura 7 - Curvas espaciais para trajetória polinomial.

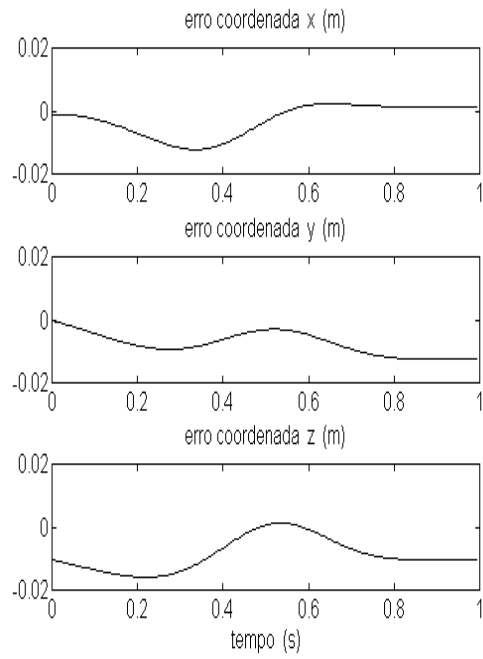


Figura 8 - Curvas de erro para trajetória exponencial.

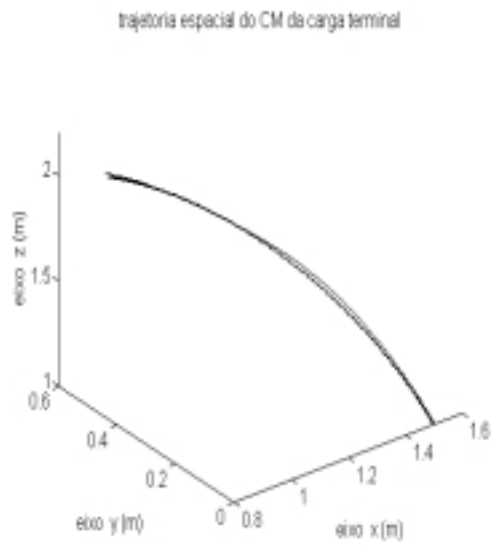


Figura 9 - Curvas espaciais para trajetória exponencial.



## 6. CONCLUSÕES

Investigou-se neste trabalho a possibilidade de se utilizar redes neurais artificiais como substituição a modelos cinemáticos de robôs manipuladores. Propôs-se ainda uma estratégia de aprendizado para a rede que permitisse à mesma devolver resultados generalizados, ou seja, para quaisquer trajetórias em termos dos ângulos das juntas como parâmetros de entrada. Comparando-se os resultados obtidos com a rede e aqueles oriundos do modelo que serviu de base para o treinamento da rede, verificaram-se pequenos erros e isto, para diversas trajetórias em termos dos ângulos das juntas. Estes erros poderiam ser ainda menores se o erro máximo admissível para o critério quadrático de minimização, no processo de ajuste dos parâmetros da rede, fosse inferior ao adotado. Neste sentido, futuras pesquisas serão desenvolvidas objetivando melhorar os resultados, mas ao custo provável do acréscimo da complexidade da estrutura da rede. Outros algoritmos de treinamento também deverão ser investigados.

## REFERENCES

- GOMES, S.C.P. , Bier, C.C (1998) Estudo sobre trajetórias de referência para controle de robôs manipuladores. Anais XII Congresso Brasileiro de Automática
- KOSKO, B., Neural network for signal processing. Prentice-Hall. 1992.
- MILLER III, W.T., Sutton, R.S., Werbos, P.J., Neural networks for control. MIT Press. 1995.
- Neural Network Toolbox – User Guide
- ZALZALA, A. M. S., Morris, A S., Neural networks for robotic control. Prentice-Hall - 1996.

## **Artificial Neural Networks to Kinematics Modeling of Robot Manipulators**

**Abstract:** *One of the problems in robotics, nowadays, is the acquisition of mathematical models of robotic manipulators. It is due, mainly, to the complexity of the manipulator, that makes difficult the way to obtain of some parameters related with the geometrical and dynamical models. This work presents an alternative approach to decrease these problems, consisting in the use of artificial neural networks to represent the forward and inverse kinematics models of a robotic manipulator with 3 degrees of freedom. The results show this is a good technique to represent those models. Furthermore, the quality of results depends on the structure of the neural network and on the learning strategy also.*

**Keywords:** *robotic manipulator, kinematics model, artificial neural networks*