



## CONTROLE NEURAL DE TRÊS ELOS DE UM ROBÔ DE CINCO GRAUS DE LIBERDADE

## NEURAL CONTROL OF THREE LINKS OF A FIVE DEGREES OF FREEDOM ROBOT

José Antônio Riul (1), Gustavo de Sá Ferreira (2), Paulo Henrique de M. Montenegro (3)

- (1) Dr. Prof., Universidade Federal da Paraíba, João Pessoa, PB, Brasil, [riul@ct.ufpb.br](mailto:riul@ct.ufpb.br), (P) [Apresentador](#)
- (2) Discente, Programa de Pós Graduação em Engenharia Mecânica, João Pessoa, PB, Brasil
- (3) Dr. Prof., Universidade Federal da Paraíba, João Pessoa, PB, Brasil

**Resumo.** O objetivo do presente trabalho é projetar um controlador neural para três elos de um robô manipulador de cinco graus de liberdade (5 GDL). Os elos do robô são acionados por motores de corrente contínua e trens de engrenagens e as respostas que são as posições angulares são captadas por potenciômetros. Neste trabalho serão projetados controladores neurais para os três elos do robô. Modelos lineares dos três elos do robô obtidos através do algoritmo dos mínimos quadrados recursivo (MQR) serão utilizados em simulações. Finalizando serão apresentados resultados obtidos através simulações.

*Palavras-chave: Robótica, controle, rede neural.*

**Abstract.** The objective of the present work is to design a neural controller for three links of five degrees of freedom (5 DOF) robot manipulator. Direct current motors and gear trains drive the links of the robot and the responses that are the angular positions are picked up by potentiometers. Linear models of the three robot links obtained through the recursive least squares (RLS) algorithm will be used in simulations. The paper ends showing results obtained through simulations with conclusions.

**Keywords: Robotics, control, neural network.**



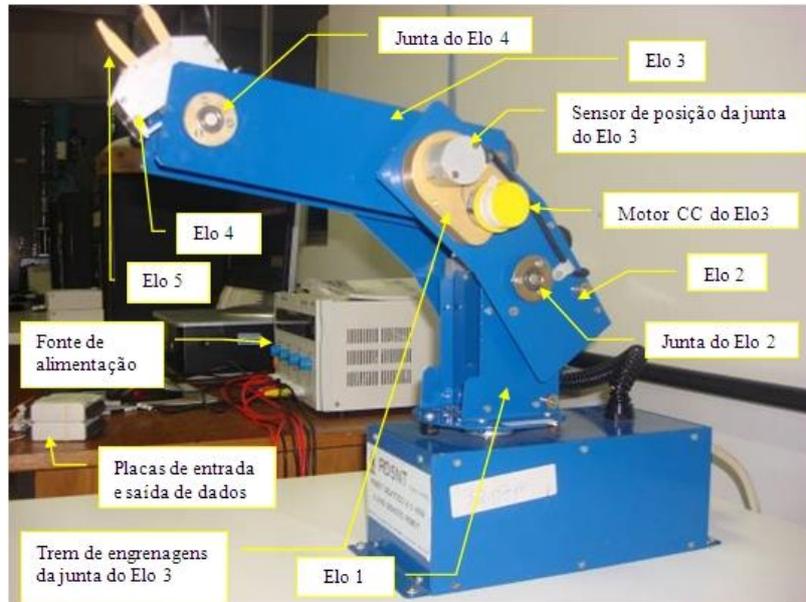
## 1 INTRODUÇÃO

O presente trabalho tem como objetivo projetar um controlador neural para três elos de um robô manipulador de cinco graus de liberdade (5 GDL). Os elos do robô são acionados por motores de corrente contínua e trens de engrenagens e as respostas que são as posições angulares são captadas por potenciômetros. Neste trabalho modelos dos três elos do robô obtidos considerando o acoplamento dinâmico entre eles, e usando o algoritmo dos mínimos quadrados recursivo (MQR) são utilizados em simulações, na obtenção das respostas dos três elos controlados por uma rede neural artificial denominada de controlador neural. O controlador neural é projetado em duas etapas, na primeira denominada etapa de treinamento, os pesos da rede são ajustados para obtenção de respostas adequadas para os elos; e na segunda denominada de etapa de controle, os pesos obtidos no treinamento são usados no início da ação de controle e são ajustados durante a ação de controle. Finalizando serão apresentados resultados de simulação, obtidos para os três elos do robô, nas etapas de treinamento e de controle.

## 2 DESCRIÇÃO DO SISTEMA

O Robô Manipulador, mostrado na Figura 1 é um robô didático, pesando aproximadamente 7 kg, referência RD5NT, fabricado pela empresa Didacta Itália, é composto de cinco juntas rotativas, quatro elos e uma garra. A primeira junta rotativa refere-se ao movimento angular da base, com deslocamento máximo de  $293^\circ$ , a segunda junta rotativa refere-se ao ombro, com deslocamento angular máximo de  $107^\circ$ , a terceira junta rotativa refere-se ao cotovelo, com deslocamento angular máximo de  $284^\circ$ , a quarta junta rotativa refere-se ao pulso com deslocamento angular máximo de  $360^\circ$  e a quinta junta rotativa refere-se a um sistema corôa / parafuso sem fim, responsável pelo curso da garra, de no máximo 22 mm, capacidade de fixação de carga de 350 gramas e parada automática por meio de uma micro chave operando com velocidade de fechamento ajustável. Os elos do robô manipulador representam o tronco, o braço, o antebraço e o punho. A transmissão de cada movimento é feita por meio de bloco motor redutor, com dois estágios de redução, e com relação de transmissão total de 1/500. Os motores dos blocos são de corrente contínua, referência 2139.906-22.112-050, fabricados pela Maxon Motor, com potência de 2,5 watts e com capacitor de longa vida. A voltagem nominal dos motores CC é de 12 volts e a rotação máxima sem carga é de 6480 rpm. A reprodução dos deslocamentos angulares das juntas e movimento da garra é assegurada por meio de potenciômetros rotativos lineares, referência 78CSB502, fabricados pela Sfernice, com resistência de 5 k $\Omega$ .

Um computador HP Compaq com processador AMD Athlon dual core de 985 Mhz e 786 MB de RAM é utilizado para enviar comando de acionamento aos motores CC e receber os sinais dos sensores potenciométricos. A comunicação do robô com o computador, é realizada através de duas placas de entrada e saída de dados, NI USB-6009 e de um programa computacional nas plataformas LabView e Matlab.



**Figura 1. Robô manipulador de 5 GDL**

Considerando as características de tensão e capacidade máxima de corrente das placas de entrada e saída de dados, houve a necessidade de introdução de um amplificador de potência para servir como fonte de suprimento aos motores de corrente contínua do robô manipulador. Este amplificador além de fornecer a potência necessária ao acionamento de cada motor, fornece a polaridade adequada para que o seu funcionamento se dê no sentido desejado. A decisão do sentido de rotação depende da tensão de excitação aplicada em seus terminais de entrada.

### 3 CONTROLADOR NEURAL

Redes neurais (Haykin, 1994) podem ser usadas como controladores adaptativos (Åstrom, 1995), (Nascimento Júnior, 2000) de sistemas e de processos. Existem vários modelos de sistemas de controle usando redes neurais, como o controlador neural inverso direto (Miller III, 1995), o controlador neural adaptativo direto (Cavalcanti, 1994), (Riul, 2017a) o controlador híbrido (Miller III, 1995). No controlador neural adaptativo direto, mostrado na Figura 2, o controlador é treinado para uma determinada referência. O alvo da rede depende do erro entre a referência e a saída do sistema e do jacobiano do sistema, que é determinado de forma direta ou indireta.

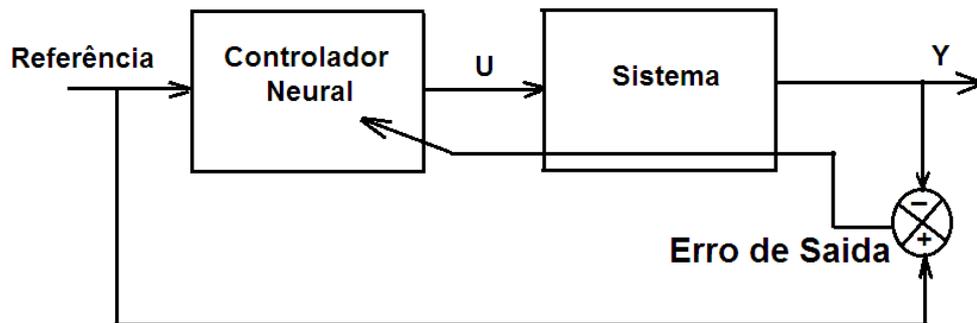


Figura 2. Diagrama de blocos do controlador neural e de um sistema

A rede neural usada para o controlador adaptativo direto é composta por uma camada de entrada (E) com neurônios lineares (L), por uma camada intermediária (I) com neurônios não lineares (NL), que usa a função de ativação tangente hiperbólica e por uma camada de saída (S) com neurônios lineares (L), conforme mostrado na Figura 3. A entrada da rede é o vetor composto pelas referências, pelas variáveis de controle e pelos erros de saída dos três elos, num total de 9 (nove) neurônios. A camada intermediária contém 4 (quatro) neurônios e a de saída 3 (três), que são as variáveis de controle dos elos a serem controlados.

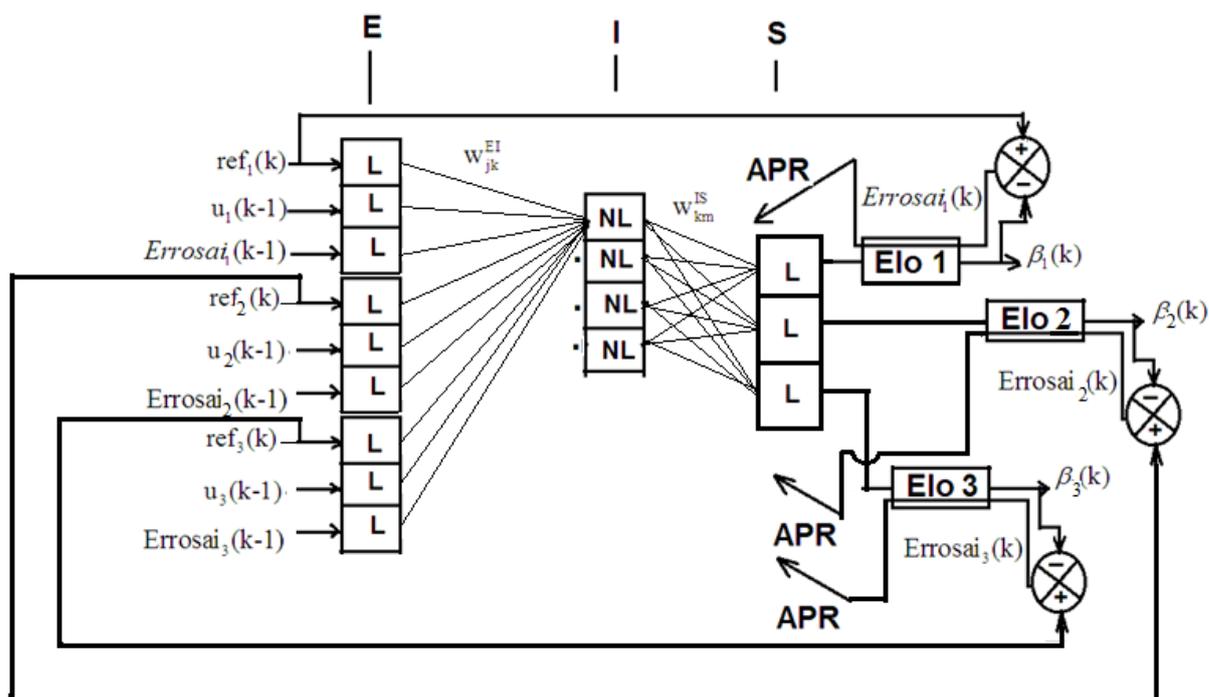


Figura 3. Controlador Neural e Elos 1, 2 e 3 – Fase de Treinamento

As Equações (1) à (6) são utilizadas para o cálculo da saída de cada neurônio, de cada camada da rede.

$$y_j^E = x_j^E \quad (1)$$

$$x_k^I = \sum_j w_{jk}^{EI} y_j^E = \sum_j (w_{jk}^{EI} y_j^E) + T_k^I \quad (2)$$

Substituindo (1) em (2), tem-se:

$$x_k^I = \sum_j w_{jk}^{EI} x_j^E = \sum_j (w_{jk}^{EI} x_j^E) + T_k^I \quad (3)$$

$$y_k^I = f(x_k^I) \quad (4)$$

$$x_m^S = \sum_k w_{km}^{IS} y_k^I \quad (5)$$

$$y_m^S = x_m^S = \sum_k w_{km}^{IS} y_k^I \quad (6)$$

onde:

j – j-ésimo neurônio da camada de entrada E;

k – k-ésimo neurônio da camada intermediária I;

m – m-ésimo neurônio da camada de saída S;

$x_j^E$  - j-ésima entrada da camada E;

$y_j^E$  - j-ésima saída da camada E;

$x_k^I$  - k-ésima entrada da camada I;

$y_k^I$  - k-ésima saída da camada I;

$x_m^S$  - m-ésima entrada da camada S;

$y_m^S$  - m-ésima saída da camada S;

$w_{jk}^{EI}$  - elemento da matriz de pesos da camada I;

$w_{km}^{IS}$  - elemento da matriz de pesos da camada de saída;

$f(\cdot)$  – função de ativação;

$T_k^I$  - bias do k-ésimo neurônio da camada intermediária.

O objetivo do modelo de controle é otimizar o índice de desempenho dado pela Equação (7), representado na forma discreta, que é função das referências e das saída dos elos.

$$I_{U,w}(k) = \frac{1}{2} (Ref(k) - Y(k))^2 \quad (7)$$

A otimização é realizada como segue: primeiro, o alvo do controlador neural é determinado usando-se o índice de desempenho  $I_U$  e o algoritmo da propagação retroativa do erro (APR) (Kovács, 1996) conforme Equações (8), (9) e (10).

$$I_U(k) = \frac{1}{2} (Ref(k) - Y(k))^2 \quad (8)$$

$$U(k) = U(k-1) - \eta \nabla I_U(k-1) \quad (9)$$

E,

$$U(k) = U(k-1) + \eta Erros_{ai}(k-1) \frac{\partial Y(k-1)}{\partial U(k-1)} \quad (10)$$

onde:

$Ref_1(k)$ ,  $Ref_2(k)$  e  $Ref_3(k)$  - são as referências dos elos 1, 2 e 3 do robô;

$u_1(k)$ ,  $u_2(k)$  e  $u_3(k)$  - são as variáveis de controle dos elos 1, 2 e 3 do robô;

$\beta_1(k)$ ,  $\beta_2(k)$  e  $\beta_3(k)$  - são saídas dos elos 1, 2 e 3 do robô;

$Erros_{ai_1}(k)$ ,  $Erros_{ai_2}(k)$  e  $Erros_{ai_3}(k)$  - são os erros de saída dos elos 1, 2 e 3 do robô;

$Errosai(k) = Ref(k) - Y(k);$

$$Re f(k) = \begin{bmatrix} Re f_1(k) \\ Re f_2(k) \\ Re f_3(k) \end{bmatrix}, \quad Y(k) = \begin{bmatrix} \beta_1(k) \\ \beta_2(k) \\ \beta_3(k) \end{bmatrix}, \quad Errosai(k) = \begin{bmatrix} Errosai_1(k) \\ Errosai_2(k) \\ Errosai_3(k) \end{bmatrix}, \quad U(k) = \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix}$$

$\frac{\partial Y(k)}{\partial U(k)}$  - jacobiano dos elos do robô;

$U(k-1)$  – saídas do controlador neural;

$U(k)$  – alvos do controlador neural;

$\eta$  - fator de correção

Com a determinação do alvo do controlador, Equação (10), usa-se o índice de desempenho dado pela Equação (11), que é função do alvo obtido para o controlador e da saída do controlador, e através do algoritmo de propagação retroativa do erro, os parâmetros da rede são ajustados conforme as Equações (11), (12) e (13).

$$I_w(k) = \frac{1}{2} (U(k) - U(k-1))^2 \quad (11)$$

$$w_{km}^{IS}(k) = w_{km}^{IS}(k-1) - \mu (\nabla I_w(w_{km}^{IS}))(k-1) \quad (12)$$

$$w_{jk}^{EI}(k) = w_{jk}^{EI}(k-1) - \mu (\nabla I_w(w_{jk}^{EI}))(k-1) \quad (13)$$

Das Equações (12) e (13), obtém-se as Equações (14) e (15), respectivamente.

$$w_{km}^{IS}(k+1) = w_{km}^{IS}(k) + \mu (E_m^S y_k^I)(k) \quad (14)$$

$$w_{jk}^{EI}(k+1) = w_{jk}^{EI}(k) + \mu \left\{ \left( \frac{\beta_k^I}{2} \left[ 1 - (y_k^I)^2 \right] x_j^E \right) \left( \sum_{m=1}^3 E_m^S w_{km}^{IS} \right) \right\} (k) \quad (15)$$

onde:

$\mu$  - fator de treinamento do controlador;

$k = t = nT_s$  – tempo discreto.

O jacobiano dos elos do robô foi calculado indiretamente usando-se modelos lineares dos elos obtidos através do método de identificação least mean squares (LMS) (Aguirre, 2000), (Coelho, 2004), (Riul, 2015), (Riul, 2016) dados pelas Equações (16), (17) e (18). Foi dada uma sequência de entrada impulsiva  $u_i(k)$  em cada elo e determinada a saída  $y_i(k)$  de cada modelo, e através dos resultados obtidos, as funções  $y_i(u_i)$  foram determinadas e conseqüentemente suas derivadas que são os jacobianos dos elos 1, 2 e 3.

$$\beta_1(k) = 0,1401\beta_1(k-1) - 0,1331\beta_1(k-2) + 0,03216u_1(k) - 0,06925u_1(k-1) \quad (16)$$

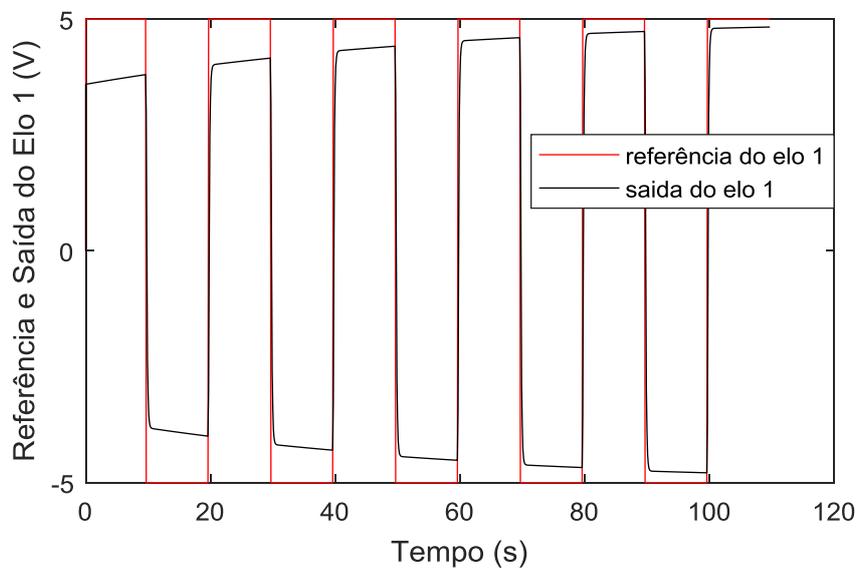
$$\beta_2(k) = 0,08801\beta_2(k-1) + 0,07410\beta_2(k-2) + 0,03488u_2(k) + 0,09685u_2(k-1) \quad (17)$$

$$\beta_3(k) = 0,04080\beta_3(k-1) + 0,03354\beta_3(k-2) + 0,05978u_3(k) + 0,1133u_3(k-1); \quad (18)$$

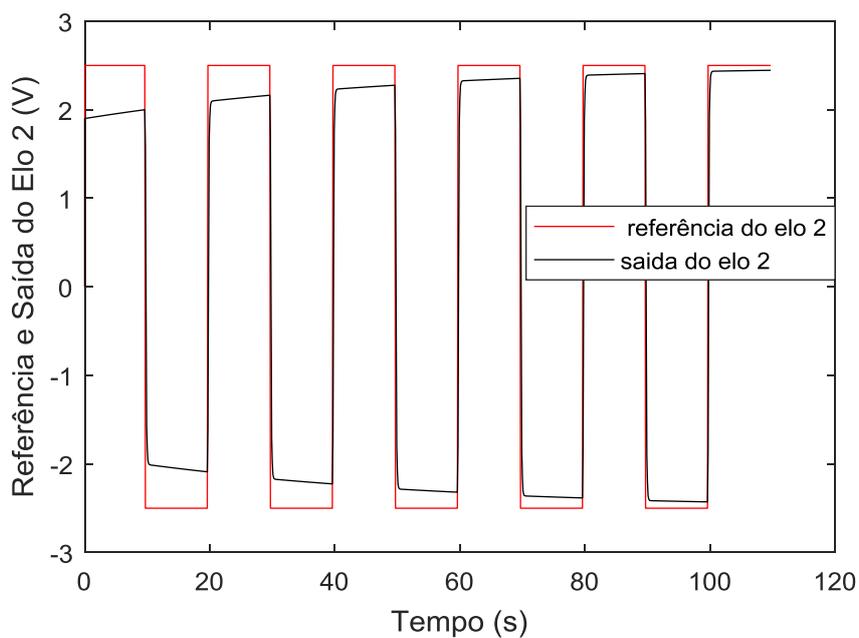
O aprendizado da variável de controle  $U(k)$  para o controlador neural direto é concluído quando o índice de desempenho  $I_{U,W}$  atinge um valor mínimo; ou seja, quando  $Y(k) \cong \text{Ref}(k)$ . Concluída a fase de treinamento os pesos  $w_{jk}^{EI}$  e  $w_{km}^{IS}$  são armazenados e são utilizados na segunda etapa para simulação dos elos 1, 2 e 3 do robô sob controle

## 4 RESULTADOS

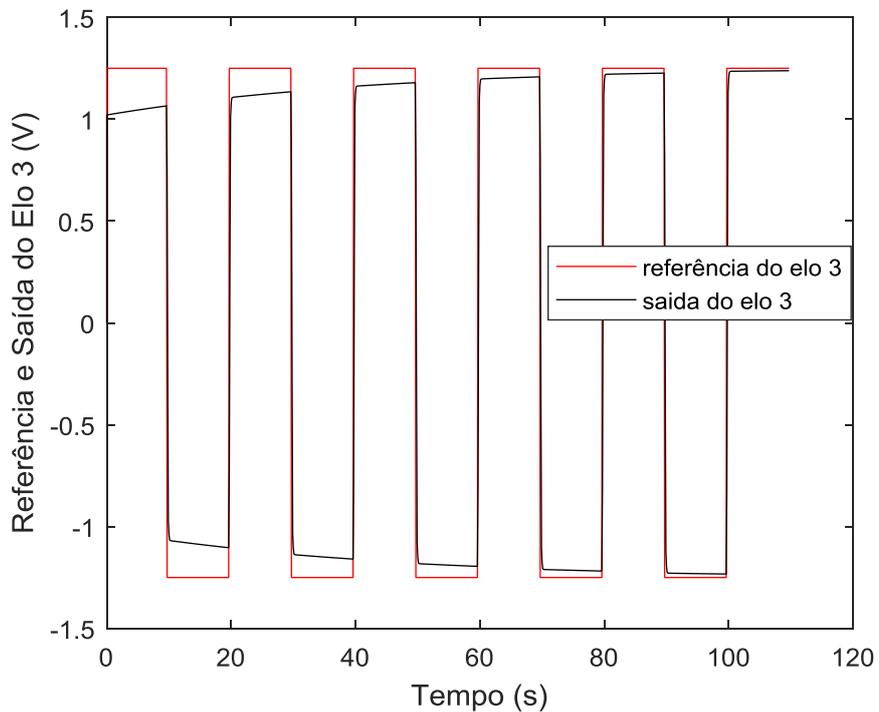
Os resultados apresentados envolvem as etapas de treinamento do controlador neural e de controle. Nas Figuras 4, 5 e 6 são mostradas as referências e as respostas obtidas para os três elos do robô, na etapa de treinamento do controlador neural. As curvas são mostradas em função do tempo discreto  $t = nT_s$ , sendo  $T_s$  o tempo de amostragem e  $n$  as amostras. O tempo de amostragem utilizado nas simulações foi de 100 milissegundos. Foram usadas 1100 amostras de cada elo do robô, a rede neural de 3 camadas mostrada na Figura 3, treinada com: fator de treinamento  $\mu = 0,3$  e 2 épocas (iterações). Na fase de treinamento os pesos da última época (iteração) foram gravados e usados na fase de controle. Os pesos treinados pela rede foram usados na etapa de controle, em simulações realizadas considerando outros sinais de referência, e os resultados são mostradas nas Figuras 7 à 9. Nesta fase os pesos da rede continuam sendo corrigidos a cada instante de tempo, o que representa um controlador adaptativo.



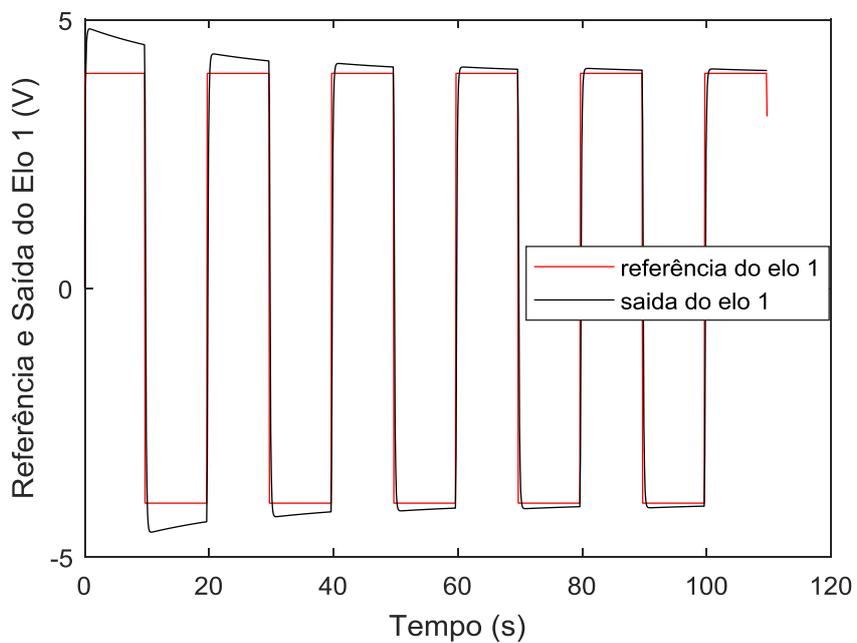
**Figura 4. Referência e saída do elo 1 na fase de treinamento**



**Figura 5. Referência e saída do elo 2 na fase de treinamento**



**Figura 6. Referência e saída do elo 3 na fase de treinamento**



**Figura 7. Referência e saída do elo 1 na fase de controle**

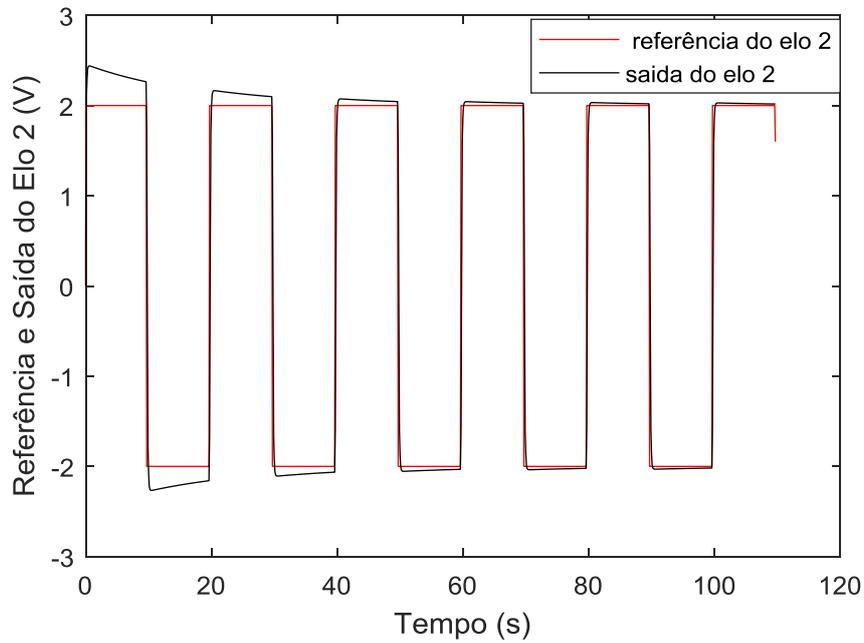


Figura 8. Referência e saída do elo 2 na fase de controle

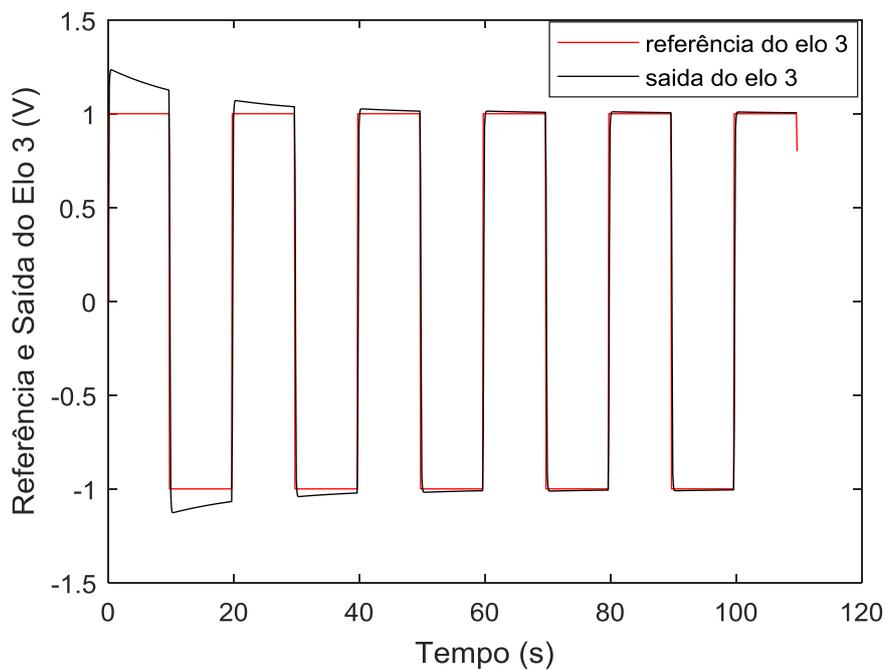


Figura 9. Referência e saída do elo 3 na fase de controle



## 5 CONCLUSÃO

Este trabalho apresentou a técnica de controle neural direto em três elos de um robô manipulador de 5 GDL. O controle da posição dos elos do robô foi realizado utilizando-se modelos dos elos do robô. O controlador neural foi inicialmente treinado para determinadas referências e posteriormente atuou nos modelos dos elos considerando-se outras referências. Os resultados indicam que as saídas na etapa de controle em comparação com as referências são satisfatórias, dados os níveis de sobressinal e tempo de resposta apresentados.

## REFERÊNCIAS

- Aguirre, L. A., 2000. *Introdução à Identificação de Sistemas: Técnicas lineares e não-lineares aplicadas a sistemas reais*, 2ª edição, Belo Horizonte, MG, UFMG.
- Åström, K. J., Wittenmark, B., 1995. *Adaptive Control*, 2ª ed., New York: Ed. Addison Wesley Publishing Company, Inc.
- Cavalcanti, J. H. F., 1994. *Controladores Neurais Adaptativos*, Tese de Doutorado, Universidade Federal da Paraíba, Campina Grande, Paraíba, Brasil.
- Coelho, A.A.R., Coelho, L.S., 2004. *Identificação de Sistemas Dinâmicos Lineares*, 1ª ed., Florianópolis: Ed. Universidade Federal de Santa Catarina.
- Haykin, S., 1994. *Neural Networks – A Comprehensive Foundation*, Macmillan, New York.
- Kovács, Z. L., 1996. *Redes Neurais Artificiais: Fundamentos e Aplicações*. São Paulo: Edição Acadêmica São Paulo.
- Miller III, W. T. & Sutton, R. S. & Werbos, P. J., 1995. *Neural Networks for Control*, The MIT Press, London.
- Nascimento Júnior, C. L. & Yoneyama, T., 2000. *Inteligência Artificial em Controle e Automação*, Editora Edgard Blucher Ltda, 1ª Edição.
- Riul, J. A. e Montenegro, P. H. M., 2015. Modelamento Não Linear de Dois Elos de um Robô Eletromecânico de Cinco Graus de Liberdade, *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3: pp. 1-7.
- Riul, J. A., Costa Junior A. G. e Montenegro, P. H. M., 2016. Nonlinear Neural Model of a Robot Five Degrees of Freedom. *IEEE Latin America Transactions*, 14: pp. 1652-1655.
- Riul, J. A., Montenegro, P. H. M. e Ferreira, G. S., 2017a. Controle Neural de Dois Elos de um Robô de Três Graus de Liberdade. *Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering (CILAMCE)*, Florianópolis.