

# **CONTROLADOR NEURAL COM APRENDIZAGEM EM TEMPO REAL APLICADO EM UM TANQUE CILÍNDRICO NÃO LINEAR**

MAICO CAMERINO, FILIPE PACHECO, LEIZER SCHNITMAN E MÁRCIO MARTINS

**Resumo:** A maioria dos sistemas que são objeto de estudo da engenharia de controle possuem alguma não linearidade. Um exemplo desta situação é o tanque cilíndrico horizontal, comumente utilizado nas indústrias de processo. Para lidar com casos como este, diversas teorias de controle foram desenvolvidas ao longo do tempo, cada uma apresentando melhores resultados em determinados sistemas. Este trabalho traz uma alternativa para controle de sistemas não lineares, sem que exista a necessidade de modelagem ou informações prévias sobre o sistema, baseado em uma nova lei de otimização para o treinamento da rede neural artificial em tempo real.

**Palavras-Chaves:** Controlador Neural; Controle não linear; Tanque Cilíndrico;

## **NEURAL CONTROLLER WITH ONLINE LEARNING ON CYLINDRICAL TANK NON-LINEAR.**

**Abstract:** Most systems that are the subject of control engineering studies have some non-linearity. An example of this is the horizontal cylindrical tank, commonly used in process industries. To deal with cases like this, several control theories have been developed over time, each one presenting better results in certain systems. This work presents an alternative for the control of nonlinear systems, without necessary modeling or previous information about the system, based on a new optimization law for the artificial neural network training in real time.

**Keywords:** Neural Controller; Nonlinear control; Cylindrical Tank.

## 1. INTRODUÇÃO

Uma das características mais fundamentais em sistemas dinâmicos não lineares é a desobediência ao princípio da superposição, válido apenas em sistemas lineares. O resultado desta observação tem como consequência que para diferentes referências fornecidas ao controlador em um sistema não linear, requer esforços de controle que variam de maneira não linear quando comparados com as variações da referência. Isto explica o porquê controladores lineares, como o PID, apresentam resultados hora eficiente e hora ineficiente em pontos distintos do processo com não linearidade (Åström e Hägglund, [1]).

Uma alternativa bastante recorrente é o desenvolvimento de teorias de controle baseada na linearização do processo em torno de um ponto usual de operação. Como exemplos, é possível citar controladores PID com múltiplas sintonias (Nandong, [2]) e controlador baseado em modelo linear (*MPC*) (Costa *et al.*, [3]). Algumas outras técnicas alternativas as de linearização também foram desenvolvidas e aplicadas, tais como: Controlador *Fuzzy* adaptativo (Khooban & Nikam, [4]), controlador neural baseado em modelo (*NNMPC*) (NARAMANI *et al.*, [5]) que por sua vez podem possuir modelos diretos e/ou indiretos do sistema (Kamalasadan & Ghandakly, [6]), sendo este uma evolução da ideia apresentado por Widrow e Walach [7] em seu livro controle inverso adaptativo.

Na indústria de processo, a maioria dos controles avançados são do tipo baseado em modelo, *MPC* (Forbes *et al.*, [8]). Contudo este tipo de controlador apresenta algumas deficiências em suas aplicações industriais. A maioria dos controladores baseados em modelos que se encontram na indústria possuem modelos lineares dos processos que controlam. A medida que o processo se afaste do ponto de criação do modelo a eficiência do controlador tende a cair devido aos erros de predição atribuídos ao modelo linear. Outro ponto importante é a não atualização periódica dos modelos, já que, os processos sofrem alterações em sua dinâmica ao passo que o tempo de operação da unidade se estende.

A alternativa encontrada por [5] demonstra uma resposta mais robusta para aumentar a eficiência deste tipo de controlador em sistemas não lineares. Contudo esta criação ocorre por lote, *off-line*, e não contempla o segundo problema abordado no parágrafo anterior. Visando apresentar uma solução alternativa aos métodos já existentes, a ideia deste artigo foi desenvolver um controlador neural que pudesse controlar um sistema desconhecido, ou seja, sem modelos, e sem informações prévias, pesos sinápticos iniciados aleatoriamente, e que ainda fosse capaz de se adaptar as mudanças do processo, aprendizado em tempo real. Sendo assim, foi desenvolvido uma nova função de otimização para ser aplicada em conjunto com o algoritmo *Backpropagation*.

## 2. METODOLOGIA

Um dos algoritmos mais conhecidos para o treinamento de uma rede neural é o *Backpropagation*, tal algoritmo pode ser utilizado para treinamento de um controlador neural por lote (Heertjes & Tso, [9]) ou em tempo real (Zhang *et al.*, [10]).

O algoritmo de treinamento de uma rede neural é baseado na ideia do erro mínimo quadrático (*MSE*), cuja função é definida como sendo:

$$E = \frac{(y - \hat{y})^2}{2} \quad (1)$$

Em que o valor de  $\hat{y}$  corresponde a saída da rede neural. Este valor pode ser definido pela seguinte equação:

$$\hat{y} = \varphi(M[X]^T) \quad (2)$$

No qual a função  $\varphi$  representa a função de ativação do neurônio,  $M$  representa a matriz de pesos sinápticos que ligam os dados da camada de entrada aos seus respectivos neurônios, e por fim,  $X$  que representa o vetor de entradas neste neurônio. Assumindo-se que a equação (1) se trata de um problema de otimização e assumindo como a matriz  $M$  as variáveis de decisão do problema, chega-se a:

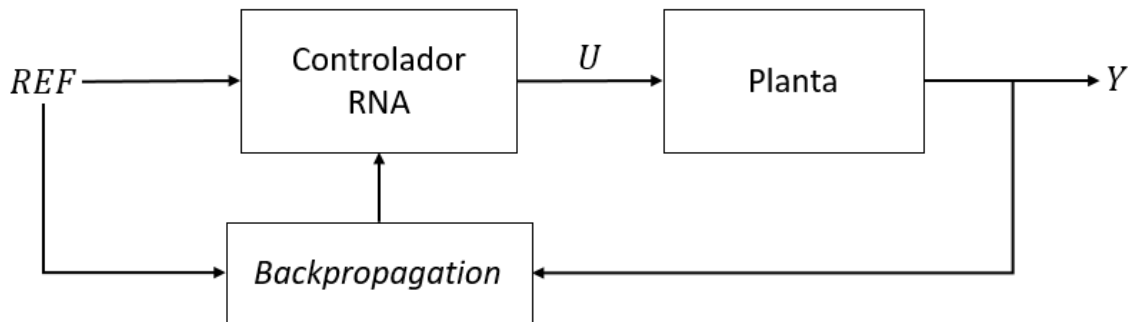
$$\frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \varphi} \frac{\partial \varphi}{\partial M} = (y - \hat{y}) \varphi'(M[X]^T)[X]^T \quad (3)$$

Escrevendo a equação (3) através do método de otimização do gradiente, tem-se que:

$$\Delta M = -\alpha \frac{\partial E}{\partial M} \quad (4)$$

A equação 3 e 4 resumem em síntese o algoritmo *Backpropagation* para apenas um neurônio. O tipo de controlador que este artigo se propõe a desenvolver, tal algoritmo escrito desta maneira, não é compatível com a definição do problema, em que não se tem um modelo conhecido da planta e que possa aprender em tempo real. A figura 1 ilustra o diagrama de bloco da estrutura de controle conforme é conhecido na literatura.

Figura 1. Diagrama de blocos do controlador neural conhecido na literatura.



Fonte: Própria (2018).

Da maneira como está apresentado acima, a saída da rede, agora representada por  $U$  e não mais por  $\hat{y}$ , não pode ser escrita da maneira que está na equação 1, pois a saída  $Y$  do processo que será comparado com o valor da *REF* é uma função ( $P$ ) dependente de  $U$ , logo a equação 1 deve ser rescrita como sendo:

$$E = \frac{(y - P(U))^2}{2} \quad (5)$$

A literatura atual propõe dois modos de treinar uma rede neural para controle de sistemas através desta maneira. O primeiro modo é o desenvolvimento de um modelo dinâmico que represente a planta ( $P$ ), as vantagens deste método é a possibilidade de treinamento em tempo real, sem a necessidade de dados prévios e a grande desvantagem é a ineficiência do controlador até que um modelo adequado da planta ( $P$ ) seja criado. O segundo modo é o levantamento de dados do processo para realização de um treinamento por lote, de maneira *off-line*.

Em ambos os casos se faz necessário a aquisição de dados para levantamento de modelos. Para evitar tal problemática, foi necessário desenvolver uma nova função objetivo para que o controlador pudesse com o algoritmo *Backpropagation* controlar o sistema em tempo real. Tal equação é demonstrada na forma matricial como sendo:

$$\min_M J = \begin{bmatrix} e_{[1xn]} \\ \Delta u_{[1xm]} \end{bmatrix}_{[1xi]}^T \begin{bmatrix} Q_{[nxn]} & \Psi_{[nxm]} \\ \Psi_{[mxn]} & R_{[mxm]} \end{bmatrix}_{[ixj]} \begin{bmatrix} e_{[nx1]} \\ \Delta u_{[mx1]} \end{bmatrix}_{[jx1]} \quad (6)$$

Em que:

$n$  = número de entradas do controlador

$m$  = número de saídas do controlador

$i = n + m$

$j = n + m$

$e$  = diferença entre referência e saída do processo ( $REF - Y$ )

$\Delta u$  = diferença entre a ação de controle atual e a anterior ( $u_k - u_{k-1}$ )

$Q, R$  e  $\Psi$  = variáveis de ponderação do controlador (sintonia)

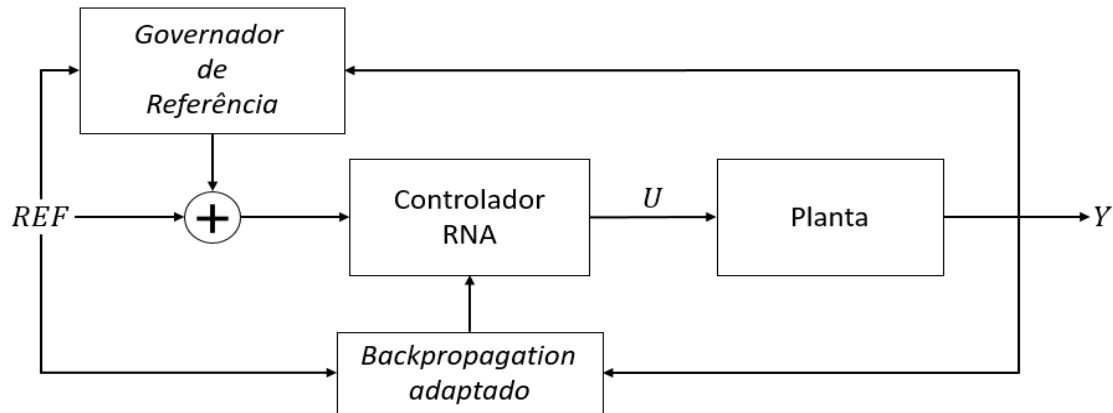
A equação 1 deve ser substituída pela equação 6 como função objetivo a ser aplicado com forma de controle pelo algoritmo *Backpropagation*. O diagrama de blocos que ilustra o controlador e o fechamento do laço *feedback* do controlador permanece o mesmo que o demonstrado na figura 1. Com a utilização desta função objetivo, é possível controlar o processo, ainda que o sistema mude, como acontece em sistemas variantes no tempo, o que é uma grande vantagem em comparação com os controladores que possuem modelo estático.

Uma desvantagem desta abordagem com relação aos controladores que possuem modelo, ainda que linearizado, é a impossibilidade de prever o efeito de suas ações de controle que foram implementadas ou que serão implementadas, ou ainda, em controladores PID onde existe a ação derivativa que busca antecipar a correção. Como resultado desta limitação, em sistemas dinâmicos cuja resposta do sistema ao estímulo de uma entrada demora a estabilizar, maiores serão as oscilações do sistema até estabilizar no valor desejado de referência. Tal comportamento se assemelha a de um sistema sub amortecido de segunda ordem, como poderá ser visto na seção seguinte.

De forma a tentar acelerar a correção do sistema e minimizar as possíveis oscilações, este trabalho faz a utilização da ideia do governador de referência. Tal implementação não é novidade em sistemas de controle, mas continua sendo um ótimo artifício para suavizar correções em sistemas não lineares. Com essa

adaptação, o diagrama de bloco que ilustra o controlador apresentado por este trabalho na figura 2 é alterado para seguinte figura.

Figura 2. Diagrama de blocos do controlador neural com governador de referência.



Fonte: Própria (2018).

A função utilizada pelo Governador de Referência é apresentada a seguir:

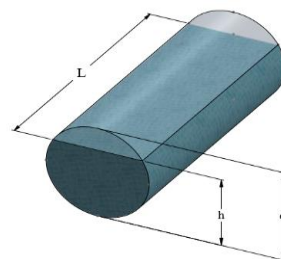
$$G = k(\Delta Y) = k(Y_k - Y_{k-1}) \quad (6)$$

A escolha desta função está baseada na ideia da taxa de variação da saída da planta com relação ao tempo. Quando o resultado da subtração, contida dentro dos parênteses, é maior que zero, significa que a variável de saída está em elevação, o que para um sistema estável, deve significar que a referência está em um valor acima da variável controlada. Para  $k < 0$ , implica que o governador de referência deverá reduzir o valor de entrada para que o controlador possua um erro menor do que o real, tornando a correção mais suavizada. O inverso ocorre quando a subtração resulta em um valor negativo.

### 3. RESULTADOS E DISCUSSÃO

O sistema escolhido para a aplicação deste controlador foi o de um tanque cilíndrico horizontal. Tal sistema possui características não lineares, tendo em vista que o volume no interior do tanque não varia de maneira linear com a variação da altura. Este sistema teve sua modelagem analítica descrita em (Pacheco, Camerino & Cajueiro [11]). Abaixo está a figura que representa o sistema.

Figura 3. Ilustração gráfica do tanque cilíndrico horizontal.



Fonte: Própria (2018).

Este sistema tem como variável manipulada a vazão de entrada, enquanto que a drenagem de água ocorre de maneira natural pela ação da gravidade. A equação 7 representa a dinâmica do sistema em função do tempo, permitindo deste modo simular numericamente os efeitos das ações de controle implementadas pelo controlador neural bem como um controlador PID utilizado para comparar os resultados do controlador desenvolvido por este trabalho.

$$\frac{dh}{dt} = \frac{1}{2r\sqrt{\frac{2hr-h^2}{r^2}}} \frac{F_i(t) - C_g C_v \sqrt{gh(t)}}{L} \quad (7)$$

A Tabela 1 apresenta os valores do modelo real utilizado para construir o modelo analítico.

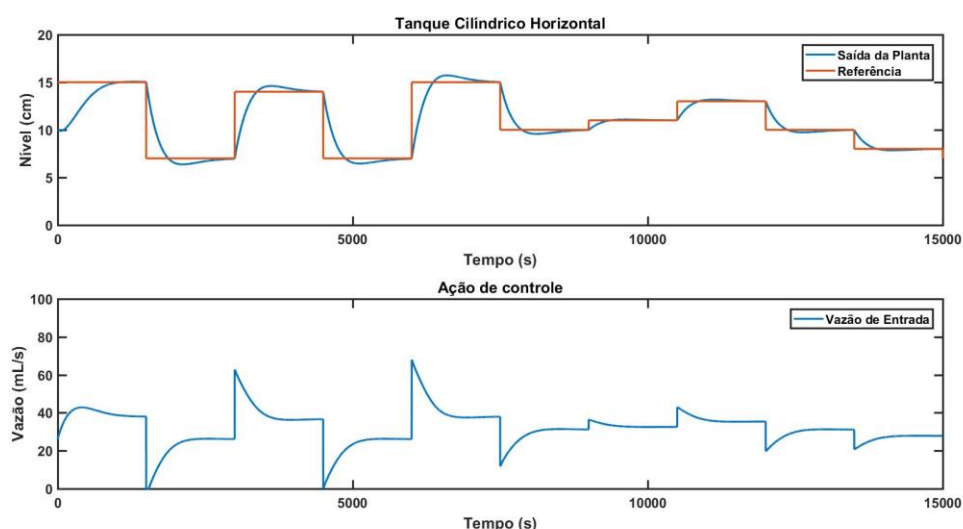
Tabela 1. Descrições das constates e variáveis.

Sigla	Descrição	Valor
$h$	Altura de coluna líquida	– cm
$F_i(t)$	Vazão de entrada	– mL/s
$r$	raio do tanque cilíndrico	0,1 m
$C_g$	Coeficiente linear da perda de carga	0,4
$C_v$	Coeficiente equivalente a abertura da válvula	$7,854 \times 10^{-5} \text{ m}^2$
$g$	Aceleração da gravidade local	$9,7833 \text{ m/s}^2$
$L$	Comprimento horizontal do tanque	0,5 m

Fonte: Própria (2018).

A sistemática durante a simulação dos dois tipos de controle foram as mesmas, o tempo de simulação foi de 15000 segundos e a cada 1500 segundos a referência era alterada aleatoriamente. Na parte superior das figuras se encontram as curvas de resposta do sistema enquanto que na parte inferior as ações de controle. A Figura 4 representa a resposta do sistema sob a ação do controlador PID.

Figura 4. Resposta do sistema com controlador PID.

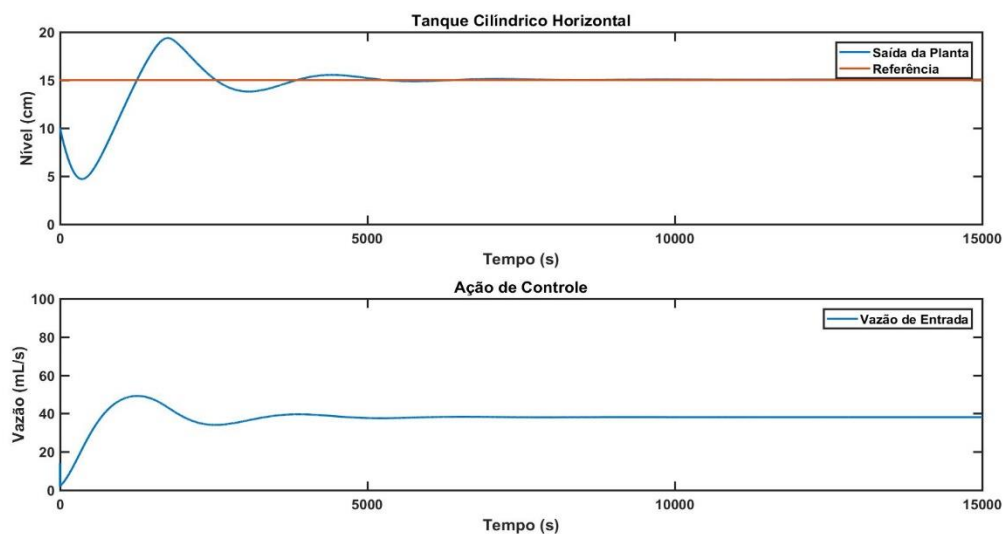


Fonte: Própria (2018).

É possível perceber que para grandes alterações na referência, este controlador linear sofre com sobressinais em sua ação de controle. Também é notável que em alguns momentos com grandes variações a sua ação de controle chega a ser zerada o que causa maior oscilação devido ao valor que é acumulado na ação de integrativa do algoritmo.

A Figura 5 representa a resposta do sistema com o controlador neural sem a ação antecipatória do Governador de Referência. Observa-se que é necessário muito tempo para que o sistema estabilize, ainda que a ação de controle sempre esteja próxima do valor de regime permanente.

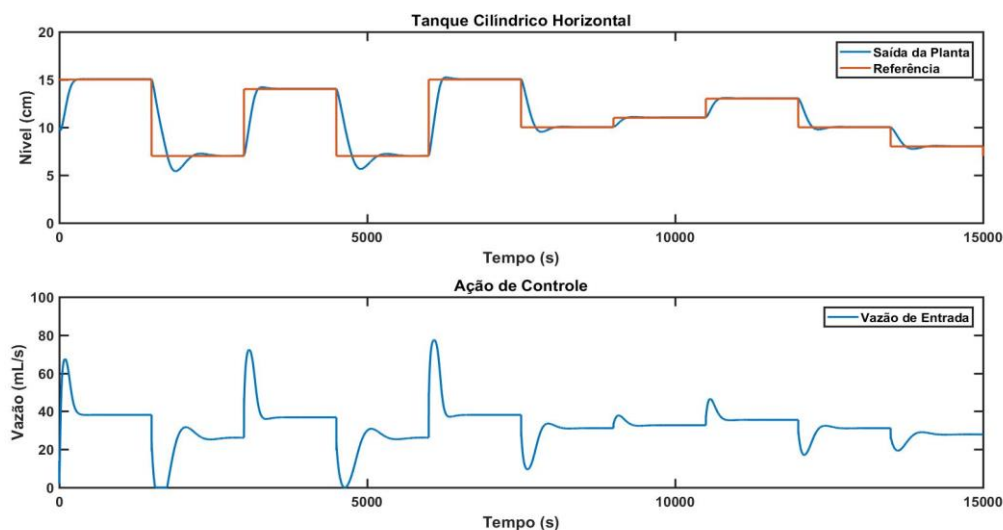
Figura 5. Resposta do sistema com controlador Neural sem Governador de Referência.



Fonte: Própria (2018).

A Figura 6 representa a curva do sistema sob a ação do controlador Neural com governador de referência. É possível perceber que a resposta de controle ocorre de maneira muito mais suave que o controlador anterior, evitando em alguns casos o sobressinal e nos demais, pequenos valores.

Figura 6. Resposta do sistema com controlador Neural com Governador de Referência.

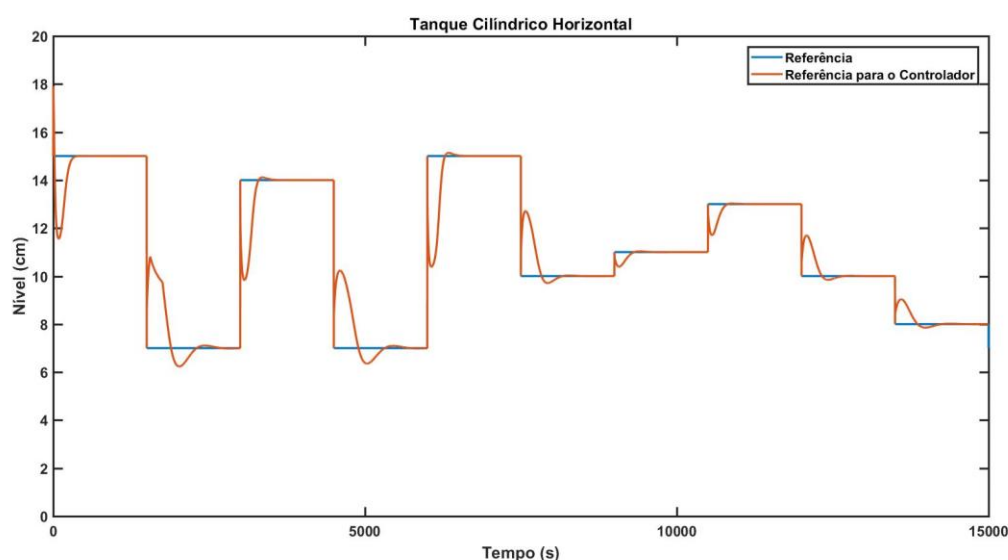


Fonte: Própria (2018).

As amplitudes de controle também se mostram ser menores. Toda esta suavidade que somente pode ser alcançada por conta da ação do controlador de referência já que o controlador neural com aprendizagem em tempo real não possui nenhuma ação que possa antecipar a sua correção.

A figura 7 representa a referência desejada para o nível em azul e em vermelho a referência que é apresentada ao controlador após a ação do controlador de referência agindo segundo a equação (6).

Figura 7. Ação do Governador de Referência.



Fonte: Própria (2018).

Na tabela 2 se encontram os valores utilizados na sintonia dos controladores nas simulações. A sintonia do controlador PID foi realizado através da ferramenta *Auto Tune* do *Matlab*, enquanto que as demais foram ajustadas por tentativa e erro.

Tabela 2. Tabela de sintonia dos controladores

Controlador	Tipo	Valor
<b>Controlador PID</b>	<i>P</i>	7706
	<i>I</i>	21,22
	<i>D</i>	-1378
<b>Controlador Neural sem Gov. de referência</b>	<i>Q</i>	0
	$\psi$	100
	<i>R</i>	0,1
	<i>K</i> (Gov. Referência)	0
	$\alpha$ (taxa de aprendizagem)	0,01
<b>Controlador Neural Com Gov. de referência</b>	<i>Q</i>	0
	$\psi$	100
	<i>R</i>	0,1
	<i>K</i> (Gov. Referência)	-100
	$\alpha$ (taxa de aprendizagem)	0,2

Fonte: Própria (2018).



Na tabela 3 está elencado os dois parâmetros para comparação dos desempenhos de ambos os controladores. No primeiro parâmetro está o *IAE*, integral absoluta do erro, enquanto que a segunda representa a somatória da utilização da MV durante toda a simulação. Os dados são significativos por si só. É notável que a utilização do governador de referência em auxílio ao controlador neural acelera em muito a sua correção com a utilização de praticamente a mesma quantidade de MV.

Tabela 3. Desempenho dos controladores

Controlador	IAE	Utilização da MV
<b>Controlador PID</b>	16,1880	0,4802
<b>Controlador Neural Com Gov. de referência</b>	3,8682	0,4830

Fonte: Própria (2018).

#### 4. CONCLUSÃO

Ao final da análise dos resultados é possível inferir que a utilização do controlador neural possui vantagens em seu desempenho frente ao controlador linear em sistemas não lineares. Contudo, o controlador proposto se apresentou muito lento em sua correção, tendo em vista que este não possui nenhuma maneira de antecipar sua correção como faz a ação derivativa no controlador PID, por isso, foi necessário se utilizar de um artifício conhecido como governador de referência, em que, parte do trabalho que seria exercido pelo controlador é previamente realizado pelo governador, a fim que o controlador possa reproduzir um esforço de controle mais suave. Os próximos passos na linha de pesquisa deste tipo de controlador é analisar seu desempenho em sistemas variantes no tempo em que existe uma necessidade real de parâmetros flexíveis para controladores de sistemas como este.

#### 5. REFERÊNCIAS

- [1] ÅSTRÖM, K.J.; HÄGGLUND, T. **Revisting the Ziegler-Nichols step response method for PID control**. Journal of Process Control, v. 14, n. 6, p. 635 – 650 – 2004.
- [2] NANDONG, J. **A multi-scale Control Approach for a PID Controller Tuning based on Complex Models**. CHEMECA. Australia, 2014.
- [3] COSTA, ERBERT A.; PATARO, IGOR M.; PACHECO, FILIPE S.; MARTINS, ANDRÉ A. F.. **Controladores Preditivo com Estabilidade Garantida em duas Camadas para Sistemas com Polos Integradores Repetidos**. CBA – 2018.
- [4] KHOOBAN, M. H.; NIKNAM, T.. **A new intelligent online Fuzzy tuning approach for multi-area load frequency control: Self adaptive Modified Bat Algorithm**. Elsevier – Eletrical Power and Energy Systems. v. 71, p. 254 – 261 – 2015.
- [5] NARIMANI, M.; WU, B.; YARAMASU, V.; CHENG, Z.; ZARGARI, N. R. **Finite Control-Set Model-Predictive Control (FCS-MPC) of Neasted Neural Point-Clamped (NNPC) Converter**. IEEE Transactions on Power Electronics, v. 30, n. 12, 2015.

- [6] KAMALASADAN, S.; GHANDAKLY, A.A.. ***A neural network based intelligent model reference adaptive controller***. IEEE International conference on Computational Intelligence for Measurement Systems and Applications, 2004.
- [7] WIDROW, B.; WALACH, E.. ***Adaptive Inverse Control***. Prentice Hall – New Jersey – 1996.
- [8] FORBES, M.; PATWARDHAN, R.; HAMADAH, H.; GOPALUNI, R. B.; ***Model Predictive Control in Industry: Challenges and Opportunities***. IFAC, v. 48, n. 8, 2015.
- [9] HEERTJES, M., TSO, T.. ***Robustness, convergence, and Lyapunov Stability of a nonlinear iterative learning control applied at a Wafer Scanner***. American Control Conference, 2007.
- [10] ZHANG, Y., SEN, P., HEARN, G.. ***An on-line trained adaptative neural controller***. IEEE Control Systems, v, 15, n, 5, 1995.
- [11] PACHECO, FILIPE; CAMERINO, MAICO; CAJUEIRO, EMANUEL B.. ***Identificação de sistema não linear em tempo real através de redes neurais artificiais***. SIINTEC, v. 4, 2018.