

Controlador Instantâneo Adaptativo Baseado em Rede Neural aplicado a um Conversor Boost

Carlúcio Pereira da Silva* Luiz Fernando Marquez Arruda*
Diego Santos Greff*

* GREEN - Grupo de Energias Renováveis, Universidade Federal de Santa Catarina, SC, (e-mail: carluciopsilva@gmail.com, eng.luizarruda@gmail.com, diego.greff@ufsc.br).

Abstract: The proposal for this work deals with a technique for controlling a Boost-type DC-DC converter, called Instantaneous Adaptive Neural Network Controller (IANNC). The proposed control method uses a Perceptron neural network that seeks to maintain the voltage converter output. Through IANNC it is possible to establish the appropriate working cycle for the Pulse Width Modulation (PWM) control signal. The proposed method was confronted with a typical Proportional Integral (PI) controller aiming to validate the model. The results were obtained using the PSIM simulator, where, in particular, the stability of the converter's output voltage against an R load was considered.

Resumo: A proposta para este trabalho trata de uma técnica para controle de um conversor CC-CC do tipo Boost, denominada Instantaneous Adaptive Neural Network Controller (IANNC). O método de controle proposto utiliza de uma rede neural do tipo Perceptron que busca manter estável a tensão de saída do conversor. Por meio do IANNC é possível estabelecer o ciclo de trabalho adequado para o sinal de controle Pulse Width Modulation (PWM). O método proposto foi confrontado com um controlador Proporcional Integral (PI) clássico objetivando validar o modelo. Os resultados foram obtidos por meio do simulador PSIM, onde, especialmente, considerou-se a estabilidade da tensão de saída do conversor frente a uma carga R.

Keywords: DC-DC Converter, Neural Controller, Predictive Controller, Artificial Neural Network, Boost Converter.

Palavras-chaves: Conversor CC-CC, Controlador Neural, Controlador Preditivo, Rede Neural Artificial, Conversor Boost.

1. INTRODUÇÃO

As fontes de alimentação de Corrente Contínua (CC) são amplamente empregadas com o objetivo de processar energia elétrica para equipamentos eletroeletrônicos, como notebooks, equipamentos hospitalares, dispositivos industriais, bem como para veículos elétricos e produtos aeroespaciais (Saadatmand et al., 2020). Um dos principais requisitos de uma fonte de tensão CC diante de uma carga é a sua estabilidade a variações de tensão, os conversores CC-CC são frequentemente utilizados para atender esse requisito. Tais conversores também apresentam destaque diante de aplicações que exigem adequação de potência elétrica gerada por sistemas alternativos de energia, como sistemas fotovoltaicos, células à combustível, por fim e não menos importante, os geradores eólicos (Utomo et al., 2011).

As metodologias de controle para conversores CC-CC permeiam dois modos: controle em modo de tensão (VMC) e controle em modo de corrente (CMC). Os controladores em modo de tensão, tradicionalmente, utilizam compensadores Proporcionais Integrais (PI) com apenas uma malha de controle para a tensão de realimentação, como sugerido em (Barbi and Martins, 2000; Greff and Barbi, 2006; Mohan, 2011). De acordo com Dong et al. (2021) a implementação

de um VMC é simples, porém a habilidade deste em lidar com rejeição de perturbações na carga é deficiente em relação ao CMC.

Neste trabalho é apresentado o IANNC controlado em modo de tensão, ainda que o modelo preconize alguma instabilidade diante de distúrbios relativos a carga.

Devido às incertezas dos parâmetros físicos e dinâmica do sistema na utilização de controladores PI Oviedo et al. (2006), propõe-se a utilização de um neurônio artificial conhecido na literatura como Perceptron, o qual foi sugerido inicialmente por Rosenblatt (1958) como modelo para aprendizagem supervisionada (Haykin, 2001). Como a tensão de saída do conversor é controlada sob a supervisão desta frente uma referência, faz do Perceptron um modelo ideal para o IANNC.

2. CONVERSOR BOOST

O conversor CC-CC elevador de tensão apresentado na Figura 1, também denominado conversor Boost, tem como objetivo ampliar a tensão de saída cujo o valor médio é no mínimo igual ao valor médio da tensão de entrada. Outro aspecto que se observa é a condição da corrente média de saída, que apresenta-se menor que a corrente média de entrada (Barbi and Martins, 2000).

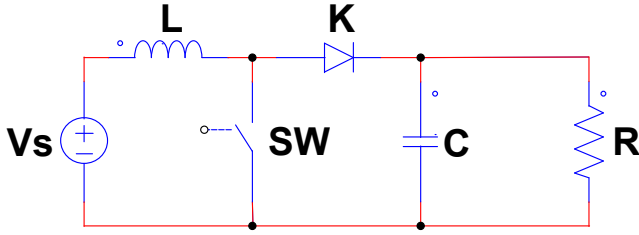


Figura 1. Conversor Boost.

Onde segundo Barbi (2014), resolvendo o sistema de equações por espaço de estados tem-se a função de transferência:

$$\frac{\bar{V}_o(s)}{\bar{d}(s)} = \frac{V_s}{(1-D)^2} \frac{\left(1 - \frac{sL}{R(1-D)^2}\right)}{\left[\frac{CL}{(1-D)^2}s^2 + \frac{L}{R(1-D)^2}s + 1\right]} \quad (1)$$

Utilizando a ferramenta MATLAB juntamente com o recurso pidtune é possível encontrar os ganhos do controlador PI, os quais foram ajustados empiricamente para ($K_p = 1,5$, $K_i = 2,5$), observado na simulação.

3. MODELAGEM DO IANNC

Não é recente a utilização de técnicas de otimização baseadas em algoritmos inteligentes como o observado em Maddaleno (1994), que apresenta uma simulação de uma rede neural para o controle do conversor Buck. Devido ao alcance em termos de otimização dessas técnicas, outros inúmeros estudos podem ser avaliados e uma breve amostra pode ser encontrada em (Kurokawa et al., 2011; Maruta et al., 2012; Nizami and Mahanta, 2014; Arruda et al., 2020), dentre os quais destaca-se Arruda et al. (2020) pelo estudo do conversor Ćuk aplicado ao rastreamento do ponto de máxima potência utilizando redes neurais.

3.1 Caracterização

A política utilizada como sugestão para o controle do conversor Boost deu origem ao IANNC, o qual trabalha com as tensões V_s , V_o e V_{ref} , onde V_s e V_o são a tensão de entrada e de saída do Boost, e V_{ref} uma referência para o ciclo ativo. V_{ref} representa a tensão de saída desejada, e funciona como guia do IANNC para fazer $V_o = V_{ref}$ a partir da atualização de D .

As tensões de entrada e saída do Boost, V_s e V_o , nessa ordem, consideram os intervalos $[10 \ 20]$ e $[10 \ 100]$ volts. Já a tensão V_{ref} encontra-se naturalmente distribuída no intervalo $[0 \ 5]$ volts. Portanto, na entrada do IANNC, tanto V_s quanto V_o encontram-se normalizadas no mesmo intervalo de V_{ref} . As tensões foram distribuídas dessa forma devido a prototipação do IANNC, especialmente no que diz respeito ao ADC (Analog to Digital Converter) do microprocessador utilizado em sua implementação física.

3.2 Estrutura Neural

Para facilitar o entendimento a despeito do modelo matemático do neurônio vinculado ao IANNC, este será identificado por k , como apontado na Figura 2. Portanto,

dada a natureza ininterrupta do controlador, o bias tem a função de aplicar uma transformação afim ao combinador linear, e o potencial de ativação da rede fica submetido ao direcionamento dado pelo bias (Haykin, 2001).

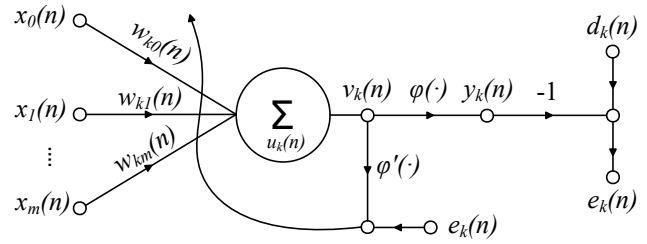


Figura 2. RNA (Rede Neural Artificial) tradicional. Adaptado de Haykin (2001).

As entradas da rede são identificadas por x_0, x_1, \dots, x_m , os pesos por $w_{k0}, w_{k1}, \dots, w_{km}$, e uma dada amostra por n . Como o IANNC possui apenas três entradas, x_m fica à caráter da entrada fixa e w_{km} como o peso relativo ao b_k (m arremete à entrada 2 da rede, portanto $x_m = x_2$ e $w_{km} = w_{k2}$). Sendo assim o resultado do combinador linear da rede, u_k , será

$$u_k(n) = \sum_{j=0}^{m-1} w_{kj}(n)x_j(n) \quad (2)$$

o potencial de ativação será v_k e b_k o bias

$$v_k(n) = u_k(n) + b_k \quad (3)$$

ao incorporar a entrada relativa a b_k à rede, considerando as equações (2) e (3), sem perda de generalidade, tem-se

$$v_k(n) = \sum_{j=0}^m w_{kj}(n)x_j(n) \quad (4)$$

a função de ativação $\varphi_k(\cdot)$ trata de uma ELU “Clevrt et al. (2015)”, $\varphi'_k(\cdot)$ a sua derivada e α é o ajuste da inclinação da função

$$\varphi_k(v_k)(n) = \begin{cases} v_k, & \text{if } v_k > 0 \\ \alpha(\exp^{v_k} - 1), & \text{if } v_k \leq 0 \end{cases} \quad (5)$$

$$\varphi'_k(v_k)(n) = \begin{cases} 1, & \text{if } v_k > 0 \\ \varphi_k(v_k) + \alpha, & \text{if } v_k \leq 0 \end{cases} \quad (6)$$

o sinal de saída será y_k

$$y_k(n) = \varphi(v_k)(n) \quad (7)$$

o erro será e_k e d_k a saída desejada

$$e_k(n) = d_k(n) - y_k(n) \quad (8)$$

o gradiente local será δ_k

$$\delta_k(n) = e_k(n)\varphi'_k(v_k)(n) \quad (9)$$

a energia total do erro será ε (C representa todos os neurônios da camada de saída)

$$\varepsilon(n) = \frac{1}{2} \sum_{j \in C} e_{kj}^2(n) \quad (10)$$

por fim tem-se a atualização dos pesos por meio da regra delta generalizada, onde se observa o termo η , que representa a taxa de aprendizado da rede

$$\sum_{j=0}^m w_{kj}(n+1) = w_{kj}(n) + \eta \delta(n) x_j(n) \quad (11)$$

3.3 Algoritmo Proposto

A rede neural proposta para o IANNC encontra-se na Figura 3, trata-se de uma adaptação do modelo observado na Figura 2. A entrada da rede é composta por três estímulos, V_e , V_s , e D_{ref} os quais deseja-se que a rede aprenda o comportamento. V_e representa o erro entre V_{ref} e V_o , e V_s a tensão de entrada do Boost. D_{ref} representa tanto o ciclo ativo desejado que é adquirido por meio de V_{ref} quanto a entrada fixa relativa ao bias. D_{err} representa o erro calculado a partir de D_{ref} e D_o . De modo que $D_o^*(n)$ e D_o , representam o ciclo ativo obtido, o primeiro trata do ciclo ativo antes da sua ativação e o segundo após a sua ativação.

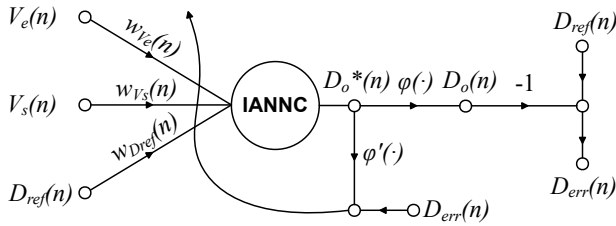


Figura 3. IANNC.

O algoritmo proposto para o controle do conversor Boost considera dois níveis de computação. Inicia-se o processamento propagando-se os estímulos pela rede, método identificado como *forward* (computação para frente). Em seguida executa-se a computação no sentido contrário, ou seja, tem-se a atualização dos pesos, método conhecido como *backward* (computação para trás). Como é um algoritmo de natureza online, não há distinção entre o momento de treinamento e o de processamento, pelo contrário, desde que o algoritmo entra em ação os pesos estão sempre sendo corrigidos e atualizados sob o prisma de um supervisor (Haykin, 2001).

A supervisão do treinamento fica a caráter da retropropagação do erro (*error back-propagation*), a qual pode ser encarada como uma generalização do algoritmo LMS (*-Least Mean Square*), que de igual modo baseia-se na regra de aprendizagem por correção de erro, também presente em filtros adaptativos (Haykin, 2001). A diferenciação entre o aprendizado proposto pelo algoritmo LMS frente ao que ocorre pelo *back-propagation*, observado no IANNC, se dá especialmente pela Equação (12) (onde se calcula o gradiente local de acordo com o LMS). Segundo Haykin (2001), de modo geral, a equação (12) representa apenas

uma estimativa para o gradiente local, diferentemente do que ocorre com a equação (9), que utiliza a derivada da função de ativação aplicada à saída do combinador linear para o cálculo do gradiente.

$$\delta_k^*(n) = (x_k)(n)e_k(n) \quad (12)$$

Os passos que executam a computação pelos métodos *forward* e *backward* realizam o treinamento da rede neural, onde o método *backward* é responsável pela execução do *back-propagation*. O alvo do treinamento é minimizar o erro entre o objetivo desejado, D_{ref} , e o calculado, D_o , o que significa fazer V_o se aproximar de V_{ref} de tal modo que o erro seja mínimo.

A presença de D_{ref} na entrada da rede confere a esta um aspecto de extrema relevância, a realimentação, haja visto que D_{ref} representa o ciclo ativo da tensão de saída desejada. Essa característica particular na configuração do IANNC resulta em um comportamento dinâmico não-linear. Assim, amplia-se a capacidade do algoritmo em resolver problemas de ordem dinâmica, o que é fortemente indicado para o Boost. De acordo com Haykin (2001), laços de realimentação tem forte influência na capacidade de aprendizagem da rede, bem como no seu desempenho.

Os dados de entrada da rede neural encontram-se normalizados no intervalo [0 1] e parametrizados de acordo com seus devidos ganhos. A exigência da função de ativação ELU frente a normalização das entradas da rede não é tão crítica quanto, por exemplo, a função logística ou tangente hiperbólica. Porém, como o aprendizado do IANNC gira em torno de fazer a rede aprender D_{ref} , a normalização fez-se especialmente necessária segundo o intervalo assumido por D .

Inicialmente, para se definir D_{ref} , precisa-se ajustar V_{ref} à magnitude de V_o mantendo a escala de variação entre ambas. Portanto, a partir de D_{max} (Tabela 1), considerando $V_{s(max)}$, $V_{o(min)}$, $V_{o(max)}$ e $V_{ref(max)}$ como definido na seção 3.1, é possível encontrar V_{ref} . De sorte que V_{ref} encontra-se escalada segundo V_o e identificada por $V_{o(ref)}$, como observado na equação 13.

$$V_{o(ref)} = \frac{V_{ref}}{V_{ref(max)}} (V_{o(max)} - V_{o(min)}) + V_{o(min)} \quad (13)$$

Consequentemente se obtém D_{ref} pela equação (14). Com isso habilita-se o IANNC a se recuperar de oscilações em V_s , haja visto que D_{ref} atualiza-se sempre que houver alteração em V_s .

$$D_{ref} = 1 - V_s/V_{o(ref)} \quad (14)$$

Assume-se que o potencial de ativação da rede, v_k , equipara-se ao ciclo ativo não ativado, $D_o^*(n)$, obtido por meio da equação (15), a qual representa a expansão da equação (4).

$$D_o^*(n) = w_{V_e}(n)V_e(n) + w_{V_s}(n)V_s(n) + w_{D_{ref}}(n)D_{ref}(n) \quad (15)$$

Considerando a RNA tradicional, o ciclo ativo obtido, y_k , é observado após a ativação de v_k , como y_k é a saída da função de ativação, portanto tem-se para o IANNC que $D_o = y_k$. De igual modo a derivada da função de ativação é dado por $\varphi'(D_o^*)$.

Em seguida calcula-se o erro entre a saída desejada, D_{ref} , e o valor obtido pela rede, D_o . A título de ampliar a legibilidade do algoritmo, D_{ref} será redefinido para D_d , o qual representa o ciclo ativo desejado, portanto o erro calculado pelo IANNC é encontrado pela equação (16).

$$D_{err}(n) = D_d(n) - D_o(n) \quad (16)$$

Também tem-se que o gradiente local que a priori foi definido pela equação (9), é atualizado para

$$\delta(n) = D_{err}(n)\varphi'(D_o^*)(n) \quad (17)$$

Pelo fato do IANNC ser do tipo monocamada com apenas um neurônio, ε se resume na energia instantânea do erro.

$$\varepsilon(n) = \frac{1}{2}(D_{err})^2(n) \quad (18)$$

Como o IANNC contém apenas três entradas, a atualização dos pesos será apresentada a partir da expansão da equação (11).

$$w_{V_e}(n+1) = w_{V_e}(n) + \eta\delta(n)V_e(n) \quad (19)$$

$$w_{V_s}(n+1) = w_{V_s}(n) + \eta\delta(n)V_s(n) \quad (20)$$

$$w_{D_{ref}}(n+1) = w_{D_{ref}}(n) + \eta\delta(n)D_{ref}(n) \quad (21)$$

No sentido de ampliar a percepção geral do algoritmo proposto para o IANNC desenvolveu-se o diagrama da Figura 4. Onde é evidenciado que a saída é atualizada apenas na medida que ε torna-se menor que o valor de *threshold*, o qual representa um limiar de tolerância para o aprendizado. Entende-se por RDG “Regra Delta Generalizada”.

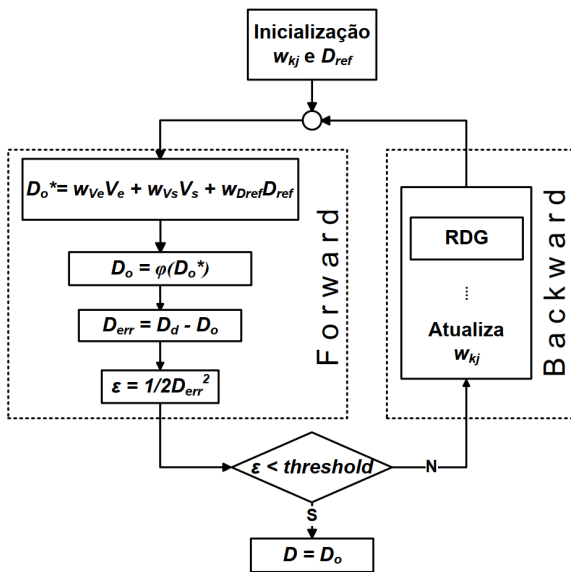


Figura 4. Diagrama de seqüência do IANNC.

4. SIMULAÇÃO E RESULTADOS

No sentido de avaliar o controlador IANNC proposto foram realizadas simulações com o software PSIM, configurado com *Time step* igual a $1 \mu s$ e *Total time* de $60 ms$. A ferramenta para desenvolvimento da rede neural no PSIM é conhecida como “C Block”, e para tal foi utilizada a linguagem de programação C. Os detalhes do controlador implementado podem ser observados na Figura 5. Já os parâmetros utilizados na simulação do conversor Boost encontram-se na Tabela 1.

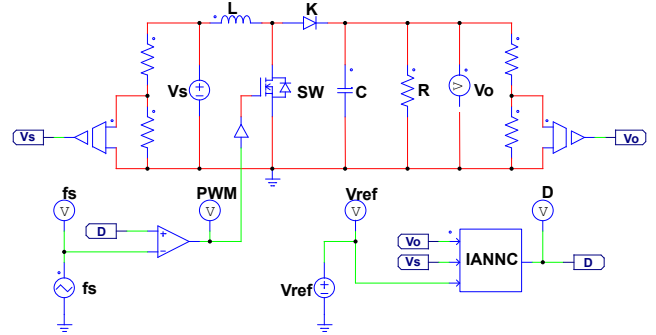


Figura 5. IANNC Boost implementado no simulador PSIM.

Considerando a Figura 6, com V_s configurado para 10 V, o IANNC ficou sujeito a alterações frequentes de V_{ref} , onde é possível avaliar a dinâmica do controlador a partir dos resultados apresentados pelos gráficos de V_o , I_o , D e também pela evolução da energia do erro. A estabilidade do algoritmo é comprovada, especialmente, a partir da curva da energia do erro, onde esta permanece sem alterações no decorrer das transições de V_{ref} , o que mostra a robustez do controlador.

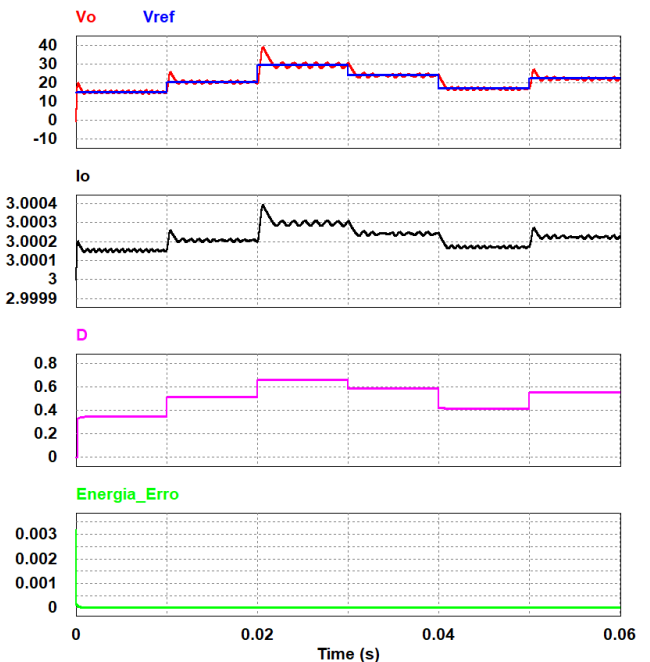


Figura 6. V_{ref} assume: 15.4 V, 20.8 V, 29.8 V, 24.4 V, 17.2 V e 22.6 V. V_o , I_o , D e a energia do erro atualizados na medida das variações de V_{ref} .

Tabela 1. Parâmetros de Simulação para o IANNC.

V_s	I_{OCR}	ΔV_o	D_{max}	L	C	R	F	η	$threshold$	α	Pesos Iniciais		
[V]	[A]	[mV]		[μ H]	[μ F]	[Ω]	[kHz]				w_{k0}	w_{k1}	w_{k2}
10-20	1.41	404	0.8	20	180	5.0	39.1	1.0e-2	1.0e-4	1.0	0.15	0.0	0.9

A comparação do IANNC com um controlador PI convencional pode ser averiguada na Figura 7. Neste exemplo é possível ver que, devido a capacidade dinâmica de aprendizado da rede neural, o controlador proposto não apresenta oscilações para D como averiguado para o controlador PI em D_PI (ciclo ativo do controlador PI). A resposta do IANNC aos frequentes degraus em V_{ref} mostra resultado superior se comparado com o controlador PI, especialmente no que tange a correção do ciclo ativo. Mesmo que V_o apresente *overshoot* superior, se comparado com V_o PI quando submetida a um degrau de subida em V_{ref} , ainda assim, do ponto de vista do controle aplicado ao ciclo ativo, o controlador PI mostrou-se inferior ao IANNC, pois arremete a chave de controle à perdas de potência superiores.

De acordo com a Figura 8, é possível avaliar a performance do IANNC frente a seguidos degraus em V_s . Constatase que o IANNC, como esperado, mostrou-se robusto ao manter V_o alinhado à V_{ref} , ainda que V_s estivesse submetido a alguma instabilidade, o que nesse caso variou segundo o intervalo [12 18] volts.

A Figura 9 apresenta o resultado da ação do IANNC no instante em que o sistema de processamento é energizado, ou seja, no momento que o microcontrolador foi ligado. Ao se considerar a curva disposta na Figura 6, especialmente abaixo de 10 ms, percebe-se que a estabilização de V_o ocorreu aproximadamente a partir de 2 ms. Ao traçar um perfil comparativo entre os resultados, pode-se afirmar que o modelo físico do IANNC respondeu satisfatoriamente frente ao modelo simulado, dado que V_o manteve-se estável a partir de aproximadamente 4 ms. A diferença entre o IANNC simulado e o protótipo deve-se as não idealidades intrínsecas do modelo real.

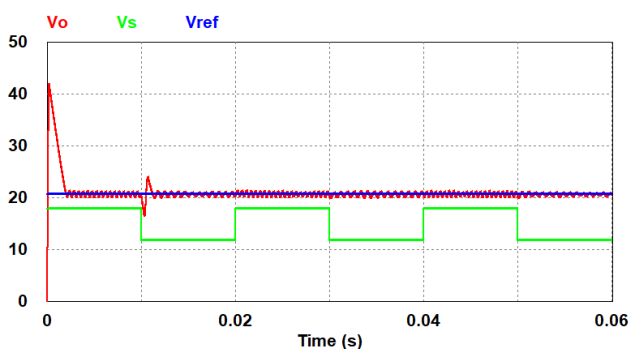


Figura 8. V_{ref} fixo em 20.8 V e V_s variando no intervalo [12 18] volts.

O cenário observado na Figura 10, o qual o protótipo do IANNC foi submetido, assemelha-se ao observado na Figura 7. A configuração de V_{ref} para o controlador foi ajustada internamente de maneira a fazer V_o assumir valores iguais a 20 V, 22 V e 15 V. Como esperado, o controlador aplicou a atualização ao ciclo ativo do sinal

PWM de controle do Boost de modo que V_o se aproxima cabalmente de V_{ref} .

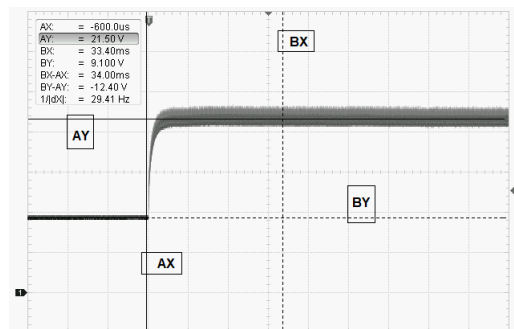


Figura 9. Atuação do IANNC no momento que o microcontrolador é energizado.

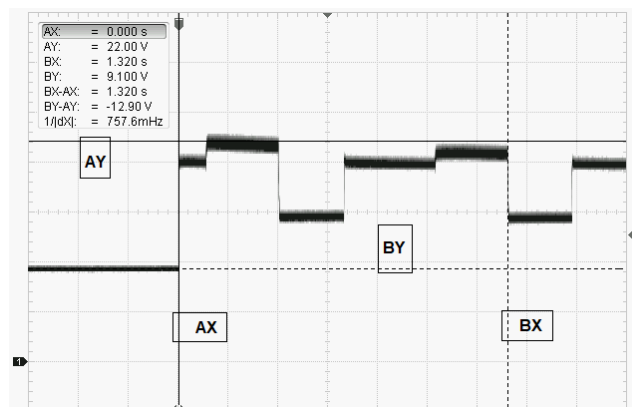


Figura 10. Controle do IANNC sobre V_o frente a variações constantes em V_{ref} .

As características do protótipo que se dedica destaque são: *i.* driver composto pelo circuito integrado IR2104; *ii.* como chave de potência o transistor IPA60R190E6; *iii.* já o processamento do sistema de controle ficou por responsabilidade do microcontrolador PIC18F4520, onde este trabalha com uma frequência de 40 MHz definida por um PLL (Phase Locked Loop) interno operando a partir de um cristal de 10 MHz externo, o que possibilita um ciclo de máquina na ordem de 100 nanossegundos.

5. CONCLUSÃO E TRABALHOS FUTUROS

Diante dos resultados apresentados pelo IANNC, observa-se que a partir do ciclo ativo do sinal PWM, o controlador conseguiu ajustar a tensão de saída do conversor Boost dentro do limite especificado pela tensão de referência. Portanto, avalia-se que o IANNC apresentou respostas satisfatórias, ficando evidente a sua eficiência ao rastrear a tensão de saída para a tensão desejada, bem como sua eficácia diante das oscilações e no tempo de estabilização da tensão de saída.

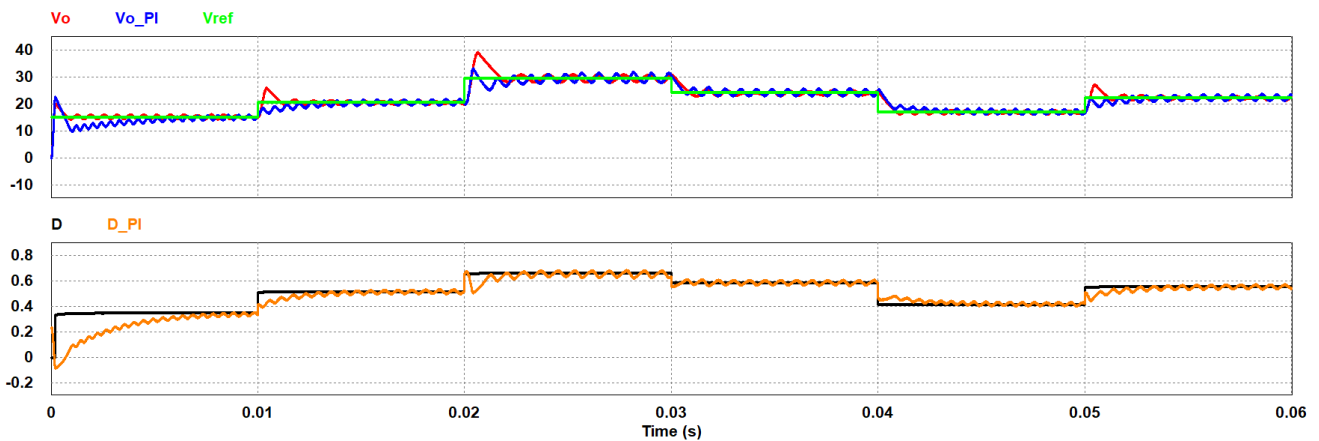


Figura 7. Comparação entre o IANNC e um controlador PI frente a variações constantes em V_{ref} . V_{ref} assume: 15.4 V, 20.8 V, 29.8 V, 24.4 V, 17.2 V e 22.6 V.

Considera-se que a estratégia adotada para o controle da tensão de saída, onde o conversor opera em modo de tensão, pode ser facilmente adaptada para que o Boost opere em modo de corrente, de sorte que o modo de rastreamento seria em função de uma corrente de referência desejada.

Outro aspecto que pode ser considerado é a aplicação do IANNC a outros controladores, tais como os conversores com saída tipo fonte de corrente, e.g. Buck e Ćuk. O fato do aprendizado ser instantâneo torna o método de fácil aplicabilidade a outras topologias de conversores.

Também pode-se avaliar o IANNC voltado para a obtenção de estímulos de entrada versus saída de um conversor CC-CC, com o objetivo de adquirir os dados para a composição de conjuntos de treinamento e validação de tais conversores frente a outras topologias de redes neurais com aprendizado supervisionado.

Por fim, embora o aprendizado instantâneo proporcione uma facilidade de implementação do IANNC em qualquer conversor tipo Boost, a utilização de amostragem das tensões entrada e saída é uma desvantagem em relação ao controle clássico, no entanto pretende-se aplicação futura deste estudo em conversores CA-CC com PFC.

REFERÊNCIAS

- Arruda, L.F.M., Lessa, M.F.V., and Greff, D.S. (2020). Aplicação de redes neurais artificiais para rastreamento de ponto de máxima potência em painel solar aplicado a um conversor Ćuk. *Anais da Sociedade Brasileira de Automática*, 2(1).
- Barbi, I. (2014). Modelagem de conversores cc-cc. *Florianópolis: [Sn]*, 81–120.
- Barbi, I. and Martins, D.C. (2000). *Conversores CC-CC Básicos não Isolados*. CEP 88.040-970, Florianópolis - SC, 4 edition.
- Clevert, D.A., Untertiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.
- Dong, W., Li, S., Fu, X., Li, Z., Fairbank, M., and Gao, Y. (2021). Control of a buck dc/dc converter using approximate dynamic programming and artificial neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(4), 1760–1768.
- Greff, D.S. and Barbi, I. (2006). A single-stage high-frequency isolated three-phase ac/dc converter. In *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, 2648–2653. doi:10.1109/IECON.2006.347607.
- Haykin, S. (2001). *Redes Neurais - Princípios e prática*. Bookman, Porto Alegre, 2 edition.
- Kurokawa, F., Ueno, K., Maruta, H., and Osuga, H. (2011). A new control method for dc-dc converter by neural network predictor with repetitive training. In *2011 10th International Conference on Machine Learning and Applications and Workshops*, volume 2, 292–297. IEEE.
- Maddaleno, F. (1994). Computer simulation of a neural network controlled buck converter. In *Proceedings of 1994 IEEE Workshop on Computers in Power Electronics*, 205–208. IEEE.
- Maruta, H., Motomura, M., Ueno, K., and Kurokawa, F. (2012). Reference modification control dc-dc converter with neural network predictor. In *2012 IEEE 13th Workshop on Control and Modeling for Power Electronics (COMPEL)*, 1–4. IEEE.
- Mohan, N. (2011). *Power electronics: a first course*. Wiley.
- Nizami, T.K. and Mahanta, C. (2014). Adaptive backstepping control for dc-dc buck converters using chebyshev neural network. In *2014 annual IEEE india conference (INDICON)*, 1–5. IEEE.
- Oviedo, J., Boelen, T., and van Overschee, P. (2006). Robust advanced pid control (rapid): Pid tuning based on engineering specifications. *IEEE Control Systems Magazine*, 26(1), 15–19. doi:10.1109/MCS.2006.1580148.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Saadatmand, S., Shamsi, P., and Ferdowsi, M. (2020). The voltage regulation of a buck converter using a neural network predictive controller. In *2020 IEEE Texas Power and Energy Conference (TPEC)*, 1–6. IEEE.
- Utomo, W.M., Bakar, A., Ahmad, M., Taufik, T., and Heriansyah, R. (2011). Online learning neural network control of buck-boost converter. In *2011 IEEE Eighth International Conference on Information Technology: New Generations*, 485–489. IEEE.