



Controle de Atitude: Uma Abordagem Através das Redes Neurais

Thadeu Augusto Medina de Carvalho¹, Ronan Arraes Jardim Chagas²

¹ Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Mestrando do curso de Mecânica Espacial e Controle ETE-CMC.

²Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP, Brasil
Divisão de Sistemas Espaciais

thadeu.augusto@yahoo.com

Resumo. Neste trabalho buscamos desenvolver e aplicar técnicas baseadas em redes neurais artificiais (RNAs) como estratégia de controle para o problema de apontamento de um corpo rígido em uma dimensão. Sendo proposta uma RNA capaz de substituir um controlador proporcional derivativo clássico (PD) em malha fechada. Tal rede foi constituída por neurônios do tipo proporcionais e derivativos, e treinada através do algoritmo de backpropagation encontrado na biblioteca de machine learning, Flux.jl. Os resultados obtidos com o neuro-controlador proposto (PD-Neural) demonstraram sua capacidade de aprendizagem e generalização, sendo capaz de anular o erro de apontamento para diversas condições de estados iniciais, após 10000 treinamentos a partir de um vetor de estado inicial. Também foi realizada uma comparação do ponto de vista computacional entre o PD-Neural e PD clássico, permitindo discutir sua viabilidade e sua implementação em computadores de bordo de satélites.

Palavras-chave: Redes neurais; Controle inteligente; Controle de atitude; PD-Neural; Flux.jl.

1. Introdução

A abordagem da inteligência artificial (IA) dita de *baixo nível* é fundamentada em modelos biológicos do ponto de vista microscópico, dos quais fazem parte os modelos de rede neurais artificiais (RNAs) e algoritmos genéticos (AGs) [Munakata 1994]. Sendo a RNA, uma nova geração de técnicas de processamento de informação que fazem uso de certos princípios organizacionais similares aos existentes na estrutura do cérebro, que viabilizam a aprendizagem, recordação e generalização, a partir de padrões ou dados coletados e armazenados [Lin et al. 1996]. Essas técnicas oferecem a vantagem no que diz respeito à velocidade de processamento, robustez e tolerância à falhas. Em contrapartida, possuem a desvantagem de se comportarem em muitos casos como “caixa preta”, não havendo um consenso analítico bem definido para entender ou modificar o processo de tomada de decisão na construção da rede [Leondes 2000]. Sendo sua abordagem, muitas vezes, caracterizada por processos heurísticos.

No contexto do controle de sistemas dinâmicos, a estrutura e as conseqüentes ações de um controlador são determinadas em resposta ao comportamento observado entre a entrada e a saída da planta, com referência mínima a um modelo matemático que a descreve [Harris et al. 1994]. Sem perda de generalidade, tal característica pode ser extrapolada ao treinamento de RNAs, uma vez que esse processo consiste em uma série de repetições de pares entrada/saída com o objetivo de levar um estado do sistema até outro estado desejado, através do ajuste de pesos



dos parâmetros das funções que compõem a RNA. Portanto, existe uma forte conexão entre o treinamento das RNAs e a teoria de controle adaptativo. Sendo o campo do *controle inteligente* uma combinação das técnicas da IA, tais como as RNAs, juntamente com as técnicas da teoria de controle [White and Sofge 1992].

Embora o uso da teoria de controle moderno seja abrangente e de uso satisfatório na indústria e engenharia de controle em geral, suas aplicações em certos problemas práticos ainda envolvem algumas dificuldades, tanto no que diz respeito ao design do controlador quanto à obtenção da função de transferência da planta, principalmente em problemas mais complexos e com maior grau de não-linearidade. Dessa maneira, a implementação das RNAs em sistemas de controle tornam-se uma estratégia interessante.

O primeiro neuro-controlador foi desenvolvido para o problema de controle de um pêndulo invertido, utilizando uma rede composta por elementos do tipo *adaptive linear element* (ADALINE) [Widrow and Smith 1964]. Ainda em meados da década de 1960 e início da década de 1970, podemos encontrar as primeiras tentativas de introdução de técnicas de *controle inteligente* em atividades da engenharia aeroespacial ([Smith et al. 1964], [Mendel 1966] e [Gilbert and Saridis 1970]). Desde a década de 1990 até os dias atuais, há um crescente interesse de aplicação de RNAs para o controle de atitude de veículos espaciais, como demonstrado pelos trabalhos de: [Kumar et al. 1995], [Carrara 1997], [Sivaprakash and Shanmugam 2005], [Hu and Xiao 2012], [Ajorkar et al. 2014] e [Shakhovska et al. 2019]. Uma das principais motivações para o uso de IA embarcada à bordo de satélites, se encontra na complexidade de realização de certas manobras através dos métodos clássicos de controle. Por exemplo, manobras de descarte de carga útil ou captura de alvos não cooperativos, nas quais há mudanças abruptas dos parâmetros de atitude e massa, podendo extrapolar os requisitos operacionais da maioria dos sistemas de controle clássicos [Ma et al. 2018]. Ainda, com o crescente aumento do tráfego de satélites e detritos espaciais em órbitas baixas, se faz necessária a discussão de novos sistemas de guiagem, controle e navegação autônomos, que permitam evitar colisões através de manobras evasivas, ou ainda recuperar o controle de um satélite em situações críticas.

Dessa forma, o *controle inteligente* oferece atualmente um grande potencial de desenvolvimento para áreas estratégicas do desenvolvimento de satélites e missões espaciais que integram e se destacam nas atividades realizadas pelo Instituto Nacional de Pesquisas Espaciais (INPE). Portanto, o presente trabalho tem o objetivo de contribuir, ainda que de forma preliminar, com a investigação das possíveis aplicações de técnicas de IA em sistemas de controle de atitude de satélites. Sendo proposto neste artigo, uma estratégia de controle através de uma RNA denominada PD-Neural, capaz de ser treinada para fornecer valores de torques de controle em malha fechada. A sua finalidade neste trabalho é controlar o ângulo de apontamento e velocidade angular do movimento de um sólido em torno de um eixo.

2. Metodologia

Para o modelo dinâmico do problema abordado, consideramos o caso de rotação em torno do eixo z de um sólido regular, o qual possui em seu centro de gravidade algum sistema atuador, capaz de fornecer torques de controle T_c em torno do eixo z definido, permitindo alterar a atitude do sólido para uma dada orientação de referência como mostrado na Figura 1.

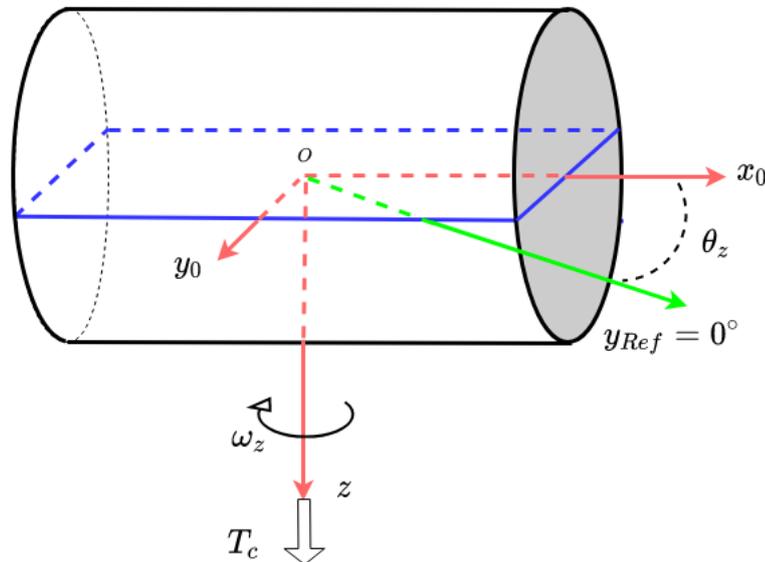


Figura 1. Diagrama esquemático do problema de apontamento de um sólido regular, com momento de inércia em torno do eixo z.

A equação de Euler para o movimento rotacional de um corpo rígido na ausência de torques externos é dada pela forma vetorial:

$$I\dot{\omega} = T_c - \omega \times I\omega . \quad (1)$$

Em que ω é o vetor velocidade angular do corpo rígido, I é o tensor de momento de inércia e o termo T_c é o vetor dos torques de controle agindo sobre o corpo. Já as equações cinemáticas podem ser definidas na forma diferencial através da representação de atitude por quatérnios.

$$\frac{d\mathbf{q}}{dt} = \frac{1}{2}\Omega\mathbf{q} . \quad (2)$$

Sendo Ω uma matriz 4×4 , com diagonal nula e composta pelas componentes do vetor de velocidade angular [Wertz 1978]. O vetor de quatérnios é definido como: $\mathbf{q} = [q_1 \ q_2 \ q_3 \ q_4]^T$. Sendo os termos q_1, q_2, q_3, q_4 os parâmetros simétricos de Euler, obtidos da parametrização da matriz de cossenos diretores, os quais especificam a orientação do sistema do corpo em relação ao sistema de referência, e definidos como:

$$\begin{cases} q_1 = e_x \text{sen}\left(\frac{\theta}{2}\right) \\ q_2 = e_y \text{sen}\left(\frac{\theta}{2}\right) \\ q_3 = e_z \text{sen}\left(\frac{\theta}{2}\right) \\ q_4 = \cos\left(\frac{\theta}{2}\right) \end{cases} , \quad (3)$$

em que e_x, e_y e e_z são as componentes do vetor unitário do sistema do corpo em relação ao sistema de referência. Ainda das propriedades da álgebra de quatérnios, temos:



$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1. \quad (4)$$

Para o caso estudado, consideramos o movimento apenas em torno do eixo z, então a matriz Ω da Equação 2 é dada por:

Através da integração numérica das equações do movimento (dinâmica e cinemática) para o caso discreto, podemos obter os estados θ , ω e $\dot{\omega}$ para cada instante de simulação do movimento do sólido. O método mais comum empregado no problema de controle de atitude de satélites é conhecido como controle proporcional derivativo (PD), com lei de controle definida por:

$$T_c = -K_1\omega - K_2\theta, \quad (5)$$

em que K_1 e K_2 são os ganhos do controlador, que podem ser obtidos a partir de diversos métodos, levando em consideração as características da função de transferência da planta e os requisitos de controle em regime permanente. Como já citado, nem sempre a função de transferência é bem conhecida, uma vez que depende dos parâmetros de massa do corpo. Além disso, o problema pode se tornar extremamente complicado para casos não-lineares. A proposta de um neuro-controlador parte do princípio de se obter os ganhos K_1 e K_2 através de processos que não dependam do conhecimento dos parâmetros da planta, sendo a rede treinada para “aprender” suas características dinâmicas e assim adaptar os valores de ganhos após seu treinamento.

Neste trabalho, foi simulado o modelo dinâmico apresentado de maneira que os valores de torques de controle T_c sejam fornecidos pela rede neural, a qual substituiu o controlador em malha fechada (Figura 2). O treinamento da rede se deu através de um algoritmo de *back-propagation*, que atualiza os parâmetros da rede a cada iteração, de forma que o índice de desempenho seja minimizado. O algoritmo foi implementado na linguagem Julia, através da biblioteca Flux.jl, específica para aplicações em *machine learning*.

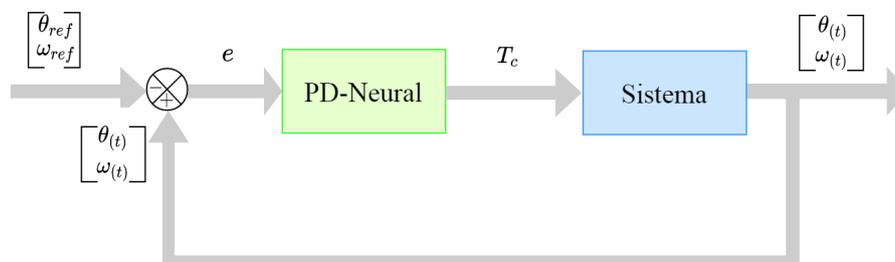


Figura 2. Diagrama de blocos para o sistema de controle proposto, com a presença do controlado PD-Neural.

Os modelos de redes neurais são compostos basicamente por três entidades fundamentais [Lin et al. 1996]: modelos de processamento dos neurônios (função de integração e ativação), modelos de conexões entre seus elementos (topologia de rede), e regras de aprendizagem. A rede PD-Neural proposta consiste em uma rede de três camadas de neurônios (*input layer*, *hidden layer*, *output layer*), como mostrado na Figura 3.

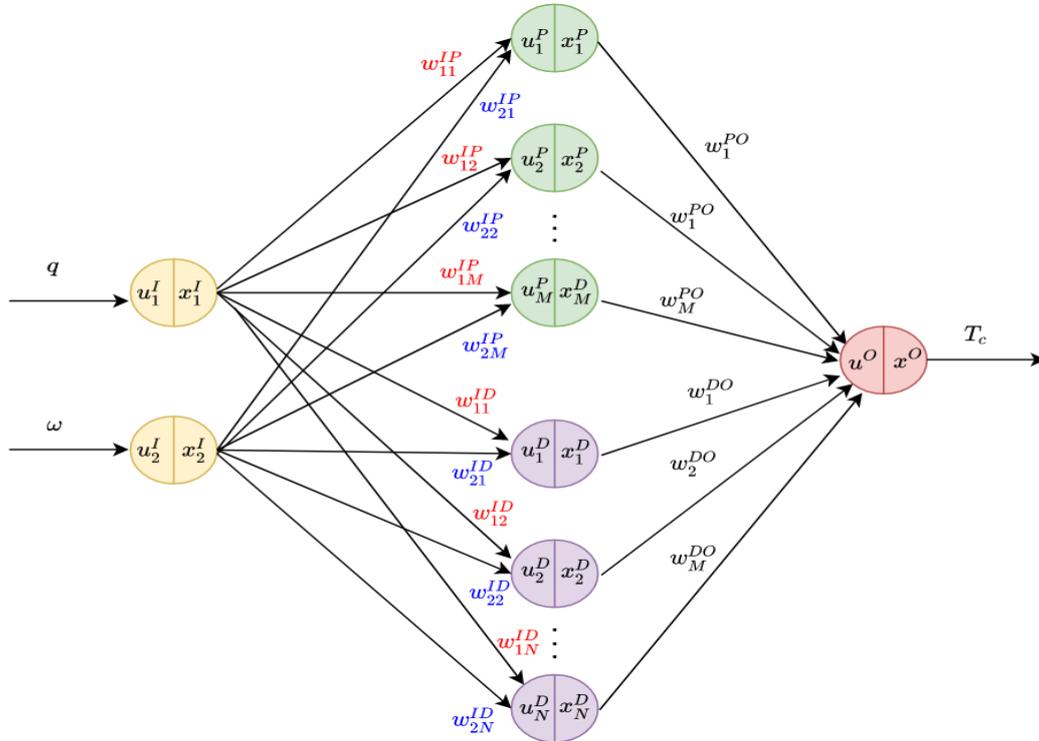


Figura 3. Estrutura da rede PD-Neural proposta.

O sinal de erro recebido pela rede PD-Neural é dado pelo próprio vetor de estado de *feedback* do sistema em malha fechada. Uma vez que consideramos o estado de referência como $[0 \ 0]^T$, a saída da rede PD-Neural atua no sentido de zerar o erro do ângulo de apontamento, definido entre o sistema do corpo e a referência, bem como a velocidade angular em torno do eixo z.

Os sinais de entrada da rede são definidos por:

$$u_1^I(k) = q_3(k) , \quad (6)$$

$$u_2^I(k) = \omega_z(k) . \quad (7)$$

A camada de entrada (*input layer*) é composta por dois neurônios do tipo P, representados pela seguinte função de ativação:

$$x_i^I(k) = \tanh(u_i^I(k)) , \quad (8)$$

para $i = 1, 2$ e k é o índice da amostragem temporal discreta da simulação. Em seguida, os sinais enviados pelos neurônios da camada de entrada são ponderados através das funções de integração, que definem os sinais de entrada dos neurônios P e D da camada interna (*hidden layer*), dadas respectivamente por:

$$u_m^P(k) = \sum_{i=1}^2 w_{im}^{IP} \cdot x_i^I(k) \quad (9)$$



$$u_n^D(k) = \sum_{i=1}^2 w_{in}^{ID} \cdot x_i^I(k), \quad (10)$$

com $m = 1, 2, 3, \dots, M$, sendo m a quantidade de neurônios P; e com $n = 1, 2, 3, \dots, N$, sendo n a quantidade de neurônios D, presentes na camada interna. Os pesos w_{in}^{IP} e w_{in}^{ID} são atualizados pelo algoritmo de *backpropagation* durante o processo de treinamento.

A função de ativação dos neurônios P da camada interna é definida por:

$$x_m^P(k) = \tanh(u_m^P(k)). \quad (11)$$

Já para os neurônios D, ela é definida por:

$$x_n^D(k) = \tanh\left(\frac{u_n^D(k) - u_n^D(k-1)}{\Delta t}\right). \quad (12)$$

Os sinais de saída da camada interna são novamente ponderados, agora pela função de integração da camada de saída (*output layer*), que contém apenas um neurônio do tipo P. As funções de integração e ativação para esta camada são dadas pelas expressões a seguir:

$$u^O(k) = \sum_{m=1}^M w_m^{PO} \cdot x_m^P(k) + \sum_{n=1}^N w_n^{DO} \cdot x_n^D(k), \quad (13)$$

$$x^O(k) = K_s [\tanh(u^O(k))]. \quad (14)$$

Portanto, a Equação 15 fornece torque de controle gerado pela rede PD-Neural. Sendo K_s a constante que limita a saída da rede em um intervalo dentro dos possíveis valores de saturação dos atuadores de controle de atitude (e.g. roda de reação). O índice de desempenho a ser minimizado pelo treinamento da rede, foi definido como:

$$J(k) = \sum_{k=0}^{t_f} \left[(1 - q_4(k))^2 + \tau T_c^2(k) \right], \quad (15)$$

em que $\tau = t_k/t_f$.

3. Resultados e Discussão

Os resultados apresentados nesta seção foram obtidos a partir do treinamento da rede, que consistiu na repetição da simulação por 10000 vezes, para o ângulo inicial de $\theta_0 = 1.1 \text{ rad}$, representado em quatérnio, e velocidade angular inicial de $\omega_0 = 0.0 \text{ rad}$. Além disso, os torques de controle foram limitados entre $[-0,075; 0,075] \text{ N.m}$ para simular a saturação do atuador. Sendo o momento de inércia em torno do eixo z do corpo definido como: $I_z = 530.0 \text{ kg.m}^2$.

Após o treinamento da rede, executamos outros casos com novos valores de entrada, diferentes dos usados no treinamento. A rede proposta foi capaz de estabilizar a rotação para diversos casos. Apresentamos aqui alguns casos e seus resultados para o erro de apontamento absoluto (Figura 4 e Figura 8), torque de controle (Figura 5 e Figura 9), componente imaginária do quatérnio (Figura 6 e Figura 10) e velocidade angular em torno do eixo z (Figura 7 e Figura 11).



O treinamento da rede durou cerca de 10 horas. Entretanto, o tempo de execução após o treinamento para os casos apresentados são mostrados na Tabela 1, junto com o tempo de execução da simulação com o PD clássico, para os mesmos valores de estado inicial. Os resultados para o erro de apontamento do PD para ambos os casos estudados são mostrados nas Figura 12 e Figura 13. Os resultados foram obtidos a partir de simulações em um computador com processador *Intel® Core™ i7-4770 CPU @ 3.40GHz x 8* e com *OS: openSUSE LEAP 15.1 Linux*.

Tabela 1. Comparação de desempenho computacional entre os controladores.

Simulação	Controlador	Tempo de execução (s)	Condições iniciais $[rad \ rad/s]^T$
Caso 1	PD-Neural	1,090	$[-3, 0 \ -0, 072]^T$
Caso 2	PD-Neural	1,004	$[5, 7 \ -0, 014]^T$
Caso 3	PD	0,844	$[-3, 0 \ -0, 072]^T$
Caso 4	PD	0,828	$[5, 7 \ -0, 014]^T$

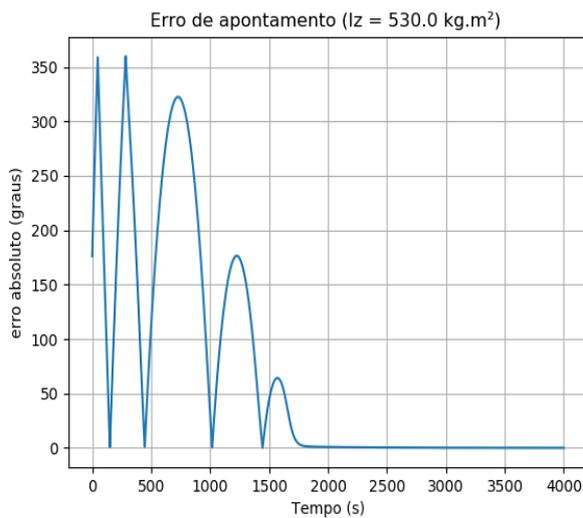


Figura 4. (caso 1).

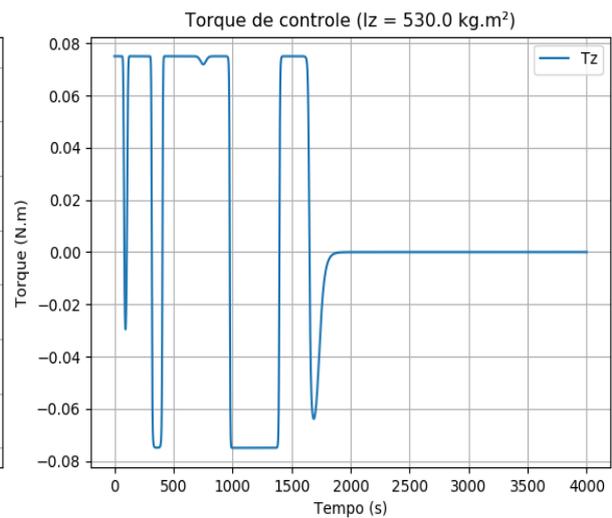


Figura 5. (caso 1).

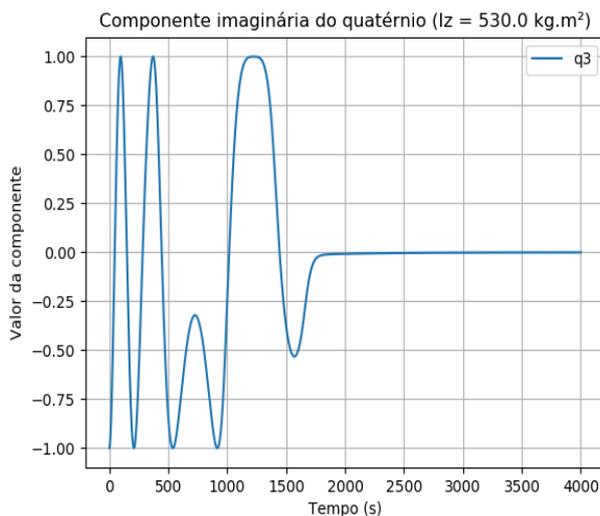


Figura 6. (caso 1).

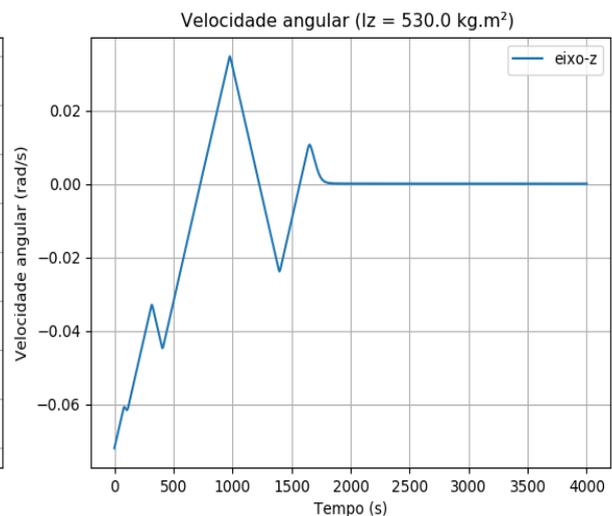


Figura 7. (caso 1).

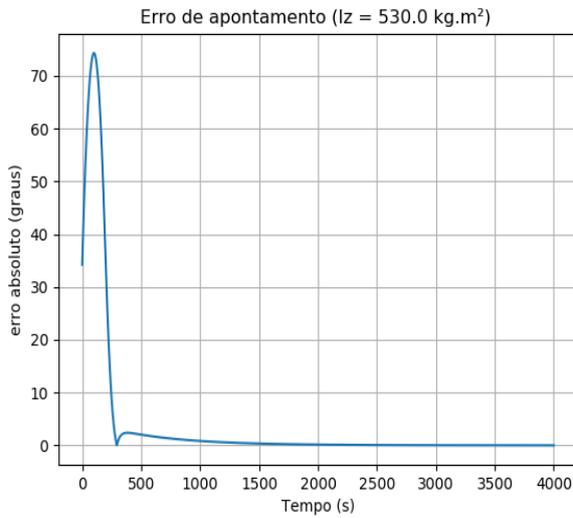


Figura 8. (caso 2).

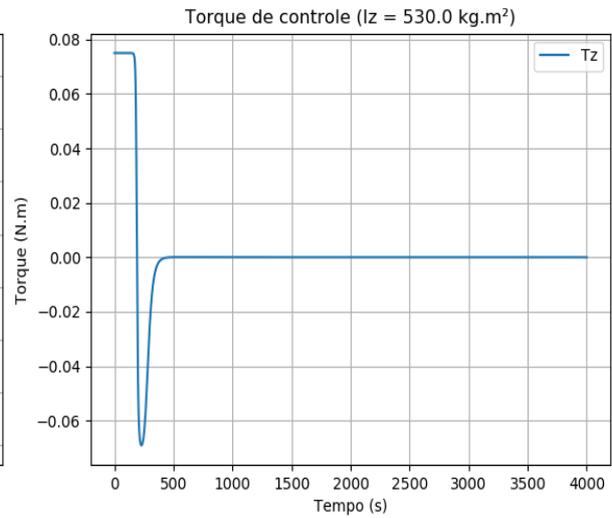


Figura 9. (caso 2).

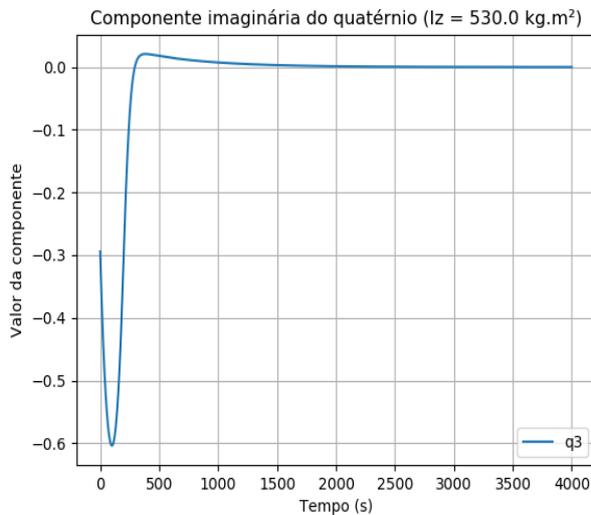


Figura 10. (caso 2).

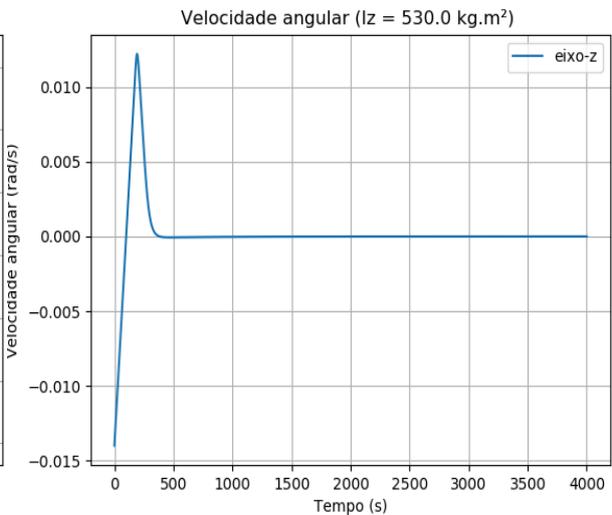


Figura 11. (caso 2).

Observamos através das Figuras (4-11) que a estabilização da atitude do corpo ocorreu com erro em regime praticamente nulo. Já na comparação do PD-Neural com o PD clássico, vemos que seu tempo de execução é cerca de 20 % mais lento. No entanto, existem possibilidades práticas de otimizações do algoritmo de treinamento que podem ser implementadas para deixar sua execução mais rápida. Portanto, vemos que, em termos de carga computacional, a implementação de um controlador através de redes neurais em computadores de bordo pode ser viável. A comparação dos resultados da Figura 12 e Figura 13 em relação aos resultados do erro de apontamento para o PD-Neural, mostrou um melhor desempenho do controlador PD clássico. Entretanto, a estratégia de controle clássica requer a sintonia dos ganhos da lei de controle a partir de certos requisitos de operação, possuindo maior limitação para se adaptar à novas condições não previstas durante o voo.

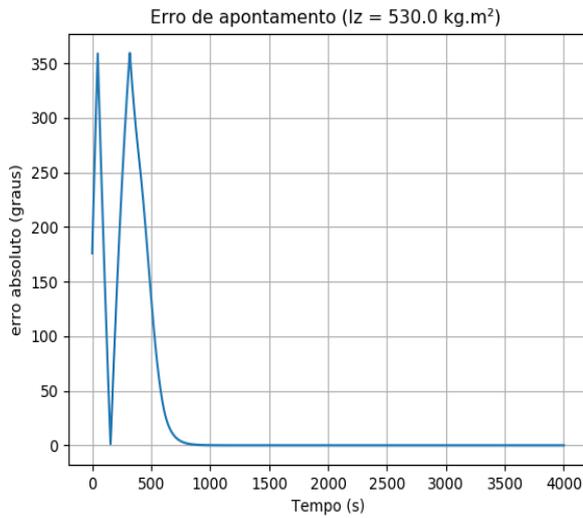


Figura 12. (caso 3).

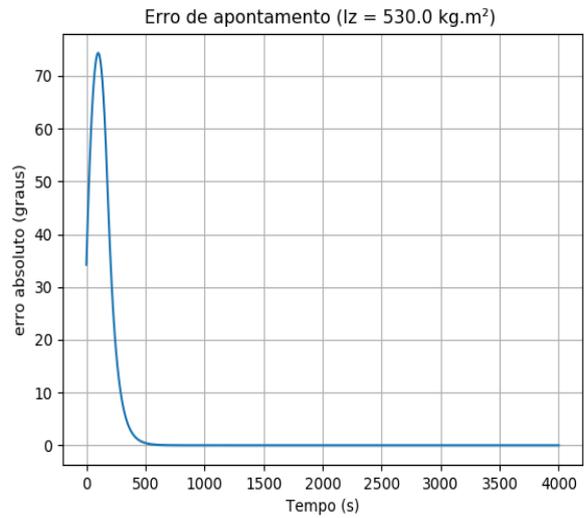


Figura 13. (caso 4).

A capacidade de adaptação do PD-Neural pode ser verificada na comparação entre os resultados da Figura 14 e Figura 15, que mostram os resultados do erro de apontamento para os dois controladores, para a simulação com os mesmos valores de estado inicial ($[3.0 \ -0.024]^T$) e com um novo valor do momento de inércia, agora para: $I_z = 10,0 \text{ kg.m}^2$. Podemos ver que o PD-Neural (Figura 14) foi capaz de controlar o apontamento, mesmo com a mudança de momento de inércia para um valor o qual a RNA não foi treinada. Já o desempenho do PD clássico (Figura 15) apresentou um erro em regime considerável, uma vez que os valores dos ganhos de sua lei de controle não foram determinados para essa nova configuração do momento de inércia.

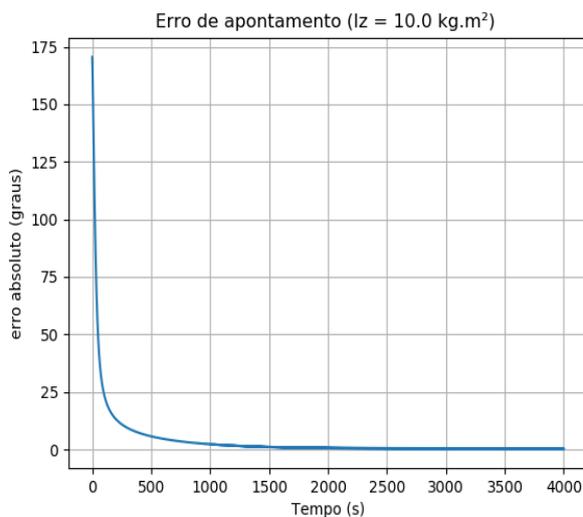


Figura 14. PD-Neural.

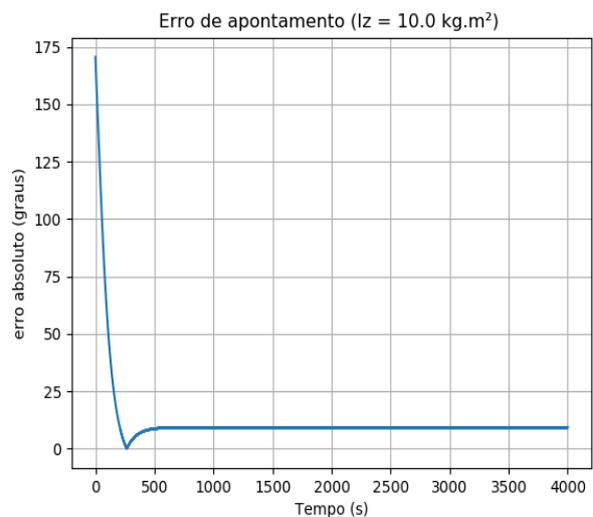


Figura 15. PD clássico.

4. Conclusão

O conjunto de resultados apresentados no presente trabalho têm caráter preliminar e consistiu em uma prova de conceito da possibilidade de estabilizar o apontamento de um corpo em uma



dimensão através de torques de controle fornecidos por RNA simples e treinada para um caso arbitrário. Ainda, verificamos a capacidade de adaptação da RNA proposta e seu potencial uso em futuros problemas de controle de atitude de satélites.

Agradecimentos: À CAPES pelo financiamento de bolsa de mestrado concedida ao autor principal.

Referências

- Ajorkar, A., Fazlyab, A., Fani Saberi, F., and Kabganian, M. (2014). Design of an adaptive-neural network attitude controller of a satellite using reaction wheels. *Journal of Applied and Computational Mechanics*, 1(2):67–73.
- Carrara, V. (1997). Redes neurais aplicadas ao controle de atitude de satélites com geometria variável. *São José dos Campos, INPE*.
- Gilbert, H. and Saridis, G. (1970). Self-organizing solution of the stochastic fuel regulator problem. *IEEE Trans. Systems Science and Cybernetics*.
- Harris, C., Moore, C., and Brown, M. (1994). *INTELLIGENT CONTROL*, volume 6. World Scientific.
- Hu, Q. and Xiao, B. (2012). Intelligent proportional-derivative control for flexible spacecraft attitude stabilization with unknown input saturation. *Aerospace Science and Technology*, 23(1):63–74.
- Kumar, R. R., Seywald, H., Deshpande, S. M., and Rahman, Z. (1995). Artificial neural networks in space station optimal attitude control. *Acta Astronautica*, 35(2-3):107–117.
- Leondes, C. T. (2000). *Knowledge-Based Systems, Four-Volume Set: Techniques and Applications*. Elsevier.
- Lin, C.-T., Lee, C. G., Lin, C.-T., and Lin, C. (1996). *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Prentice hall PTR Upper Saddle River NJ.
- Ma, Z., Wang, Y., Yang, Y., Wang, Z., Tang, L., and Ackland, S. (2018). Reinforcement learning-based satellite attitude stabilization method for non-cooperative target capturing. *Sensors*, 18(12):4331.
- Mendel, J. (1966). Applications of artificial intelligence to spacecraft control. Technical report.
- Munakata, T. (1994). Commercial and industrial ai. *Communications of the ACM*, 37(3):23–26.
- Shakhovska, N., Montenegro, S., Kryvenchuk, Y., and Zakharchuk, M. (2019). The neurocontroller for satellite rotation. *International Journal of Intelligent Systems and Applications*, 11(3):1.
- Sivaprakash, N. and Shanmugam, J. (2005). Neural network based three axis satellite attitude control using only magnetic torquers. In *24th Digital Avionics Systems Conference*, volume 2, pages 6–pp. IEEE.
- Smith, F. B., Lee, J. F., Butz, A. R., and Prom, G. J. (1964). Trainable flight control system investigation. Technical report, HONEYWELL INC ST PAUL MINN.
- Wertz, J. R. (1978). Spacecraft attitude determination and control, vol. 73, d. *Astrophysics and Space Science Library*.
- White, D. A. and Sofge, D. A. (1992). *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*. Van Nostrand Reinhold Company.
- Widrow, B. and Smith, F. W. (1964). Pattern-recognizing control systems.