Preprints, 4th IFAC Conference on
Advances in Proportional-Integral-Derivative Control
Almería, Spain | June 12-14, 2024

Neuron-based back-calculation anti-windup strategy: first results

Andrés J. Serrano-Balbontín Inés Tejado Blas M. Vinagre

Universidad de Extremadura, Escuela de Ingenierías Industriales, 06006 Badajoz, Spain (e-mail: {ajserranob,itejbal,bvinagre}@unex.es).

Abstract: Back-calculation is an anti-windup strategy that consists in adding a feedback path to act on the integrator of a proportional-integral-derivative (PID) controller when the actuator causes a saturation. Neuromorphic control (NC) uses neuron models to encode the control signal into pulses. This paper is a first attempt towards using basis of NC as an anti-windup method in a PID controller. More precisely, a neuron is utilized to encode the saturation error (namely, the difference between the controller and the actuator outputs) into pulses, causing that the back-calculation works in lapses according to error intensity. A classical control problem with saturation is given as illustrative example of application of the proposed strategy.

Keywords: Anti-windup, neuromorphic, back-calculation, neuron models.

1. INTRODUCTION

Neuromorphic control (NC) is characterized by a control structure in which the control signal is encoded into pulses by bioinspired circuits known as silicon neurons or simply neurons. It is said that this type of information transmission is robust and efficient (Mead, 1989).

This control strategy arises within a context of exploring the potential of analog very large-scale integration (aVLSI) circuits. Analog circuits behavior is closer to biological systems because they have an asynchronous (event-driven) nature, while digital circuits are dependent on a clock signal. Specifically, the first use of NC was as an application of pulse-type signals to deal with the movement of a DC motor at very low speeds, a range in which nonlinear friction is very relevant. When the reference is very low, conventional controllers can cause the stick-slip phenomenon. In addition, the motor can stop due to small disturbances. NC manages these problems by guaranteeing that in each pulse the motor receives enough energy to surpass static friction, in such a way that it is able to (re)start its movement (DeWeerth et al., 1991).

Nowadays, the term "neuromorphic" commonly encompasses a wide range of bioinspired hardware implementations. For example, recent literature has predominantly linked this term to neural network-based works (Schuman et al., 2017). In particular, in many works, "neuromorphic control" is used to refer to controllers based on artificial neural networks, which substantially differs from the interpretation of the term considered in this paper. Likewise, while NC may be similar to conventional modulation strategies, it strives to further imitate biological processes. Neuromorphic controlled actuators could be used together with neuromorphic sensors and processing algorithms, such as vision sensors (Indiveri, 1999) and sensorimotor integration (Krauhausen et al., 2021), to conform a fully pulse-driven robotic system.

One of the best-known potential sources of proportional-integral-derivative (PID) controller degradation in practice is the so-called windup phenomenon in the integral part, which occurs when the controller output saturates due to the limits of the actuator. This generally results in large overshoots and settling times. A general approach to overcome such a practical problem is to design the controller without considering the actuator nonlinearity. Then, the detrimental effects due to the integrator saturation are compensated by incorporating an additional functionality to the classical PID algorithm, which is conveniently designed for this purpose (Visioli, 2006). A variety of strategies to limit this effect can be found in the literature. Among them, it is worth mentioning the following (see e.g. (da Silva et al., 2018; Zaccarian and Teel, 2011; Tarbouriech and Turner, 2009; Visioli, 2006) and references therein): limiting or smoothing the set-point changes and/or selecting a slower controller, conditional integration (such as limiting the integral term to a predefined value or when the error is greater than a predefined threshold or the controller output saturates), recalculating the integral term when it saturates by feeding back the difference between the saturated and unsaturated control signals multiplied by a constant gain (this strategy is known as back-calculation), and automatic reset implementation by inserting the saturation function in the control scheme.

Although anti-windup techniques have been used in many studies, there are still some open problems, such as their application to nonlinear systems (Tarbouriech and Turner, 2009). Likewise, in principle, none of the strategies proposed in the literature can handle this phenomenon for all

⋆ This work is part of projects PID2022-141409OB-C22 and PID2019111278RB-C22, supported by MCIN/AEI/10.13039/501100011033/FEDER, EU. Andrés Serrano would like to thank the Ministerio de Ciencia, Innovación y Universidades its support through the scholarship no. FPU22/00885 of the FPU Program.

cases or applications, but, depending on the application, it is common to find adaptations of existing strategies.

Our previous works on NC consisted in exploring the benefits of combining neurons with fractional order controllers to deal with friction and increase controller robustness (Serrano-Balbontín et al., 2023a). An analog implementation of this combination was proposed in (Serrano et al., 2023). Additionally, the extension of NC by using fractional order neurons, which have a closer behavior to that of actual neurons, was proposed in (Serrano-Balbontín et al., 2023b). In this context, the research initiated with this paper explores the possibility of using basis of NC as an anti-windup method in a PID controller due to its advantages of encoding information into pulses like those in biological nervous systems. In particular, in this study a silicon neuron is placed in the back-calculation feedback path to observe the effect of using a pulse-like signal instead of continuous.

The remainder of this paper is organized as follows. Section 2 briefly describes the basics of the most common anti-windup strategies, namely classical back-calculation and conditional integration. Section 3 presents a neuron-based back-calculation strategy, including the description of the neuron model used to handle the windup of the integrator. The effectiveness of the mentioned strategy is demonstrated in Section 4 through a classical control problem in comparison with other existing anti-windup strategies. Section 5 covers concluding remarks of this study.

2. FUNDAMENTALS ON ANTI-WINDUP STRATEGIES

PID controllers dealing with limited actuators are often designed to work on the so-called small signal range, i.e., expecting the PID to work within the limits. Nonetheless, for certain reference values there exists a risk of integral windup. Instead of changing the design of the controller, the PID structure is preserved and some mechanisms acting on the integral part are added to deal with those cases. Two classical anti-windup strategies are recalled next: back-calculation and conditional integration.

2.1 Classical back-calculation

In classical back-calculation, an additional feedback path is generated by feeding the error e_s between the actual controller output u, and the output of the actuator v (henceforth referred to as saturation error: $e_s = v - u$) through a gain of value k_t, as shown in Fig. 1. This serves to recompute the integral term in the controller and do not have any effect when the actuator does not saturate (Åström and Hägglund, 2006).

2.2 Conditional integration

Conditional integration or integrator clamping consists in switching off the integration when a certain condition is verified (Visioli, 2006). The possible condition more similar to back-calculation is related to saturation error, in contrast with other conditions involving the control error (e). The most common approach is to switch off the integration when there exists actuator saturation ($e_s \neq 0$). Conditional integration structure is illustrated in Fig. 2, where the new feedback path consists in determining whether the error e_s equals to zero. If true, the PID controller works normally, but, if the condition is no longer satisfied, the integrator receives a zero as input.

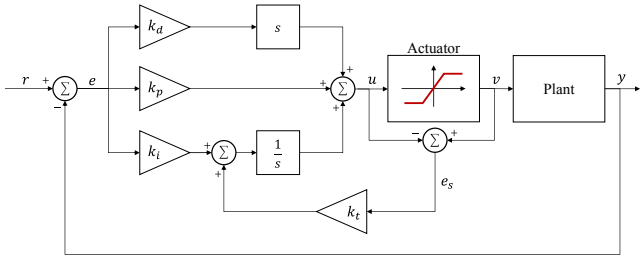

Fig. 1. Diagram of classical back-calculation structure.

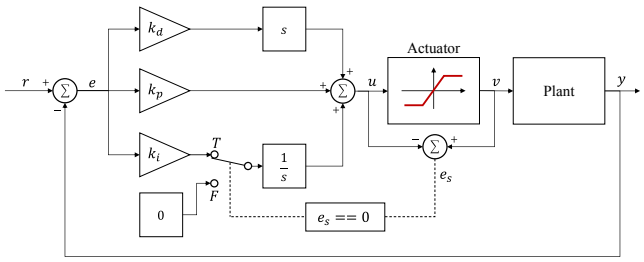

Fig. 2. Diagram of conditional integration structure based on saturation error.

3. NEURON-BASED BACK-CALCULATION

In this section, a new back-calculation structure is proposed in order to reach a control loop able to handle integral windup using neuromorphic principles.

3.1 Neuron model

The output of a neuron is a succession of pulses of constant amplitude whose firing frequency varies according to the input amplitude. In Fig. 3, the relation between the input i and the output n of a neuron is illustrated. It shows how the firing frequency of spikes decreases as the amplitude of i decreases. Specifically, the firing frequency of neuron models known as integrate-and-fire neurons is determined by a process of integration until a threshold is reached, at which point the neuron resets and begins again (Indiveri et al., 2011). The most appropriate variant for control purposes is one able to produce pulse frequency modulation (PFM) (Tzafestas and Frangakis, 1980) as it is able to preserve average information transmission unlike other models that are thought to reproduce biophysical dynamics of actual neurons. It can be modeled as follows:

$$x(t) = \int_{t_{k-1}}^{t} i(t)dt, \text{ if } x(t) < K_{ti} \tag{1}$$

$$x(t_k^+) \leftarrow 0, \text{ if } x(t_k) = K_{ti} \tag{2}$$

where $x(t)$ is the value of the integral at time t, t_k the instant at which k-th pulse is fired, and K_{ti} the threshold. It is important to remark that the input to a neuron is

positive ($i(t) \geq 0$). The output of the neuron $n(t)$ is ruled by:

$$n(t) = \begin{cases} A & \text{if } 0 \leq t - t_{k-1} \leq t_h \\ 0 & \text{if } t_h < t - t_{k-1} < t_k \end{cases} \quad (3)$$

where A is the amplitude of the pulses, t_h the width of the pulses or duration in high state, and t_l the time between pulses or low state, as seen in Fig. 3. Equation (1) can be rewritten as

$$t_k - t_{k-1} = t_h + t_l = \frac{K_{ti}}{\bar{i}} \quad (4)$$

to show that the period of fired pulses varies according to the inverse of the average input amplitude, $\bar{i}$, in that interval with a proportional constant equal to the threshold. The input-to-output gain of the neuron G_N can be deducted from the average value of the output in one period and is given by:

$$G_N = \frac{At_h}{K_{ti}} \quad (5)$$

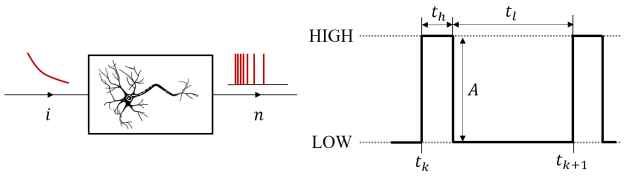

Fig. 3. Description of input to output relation of a neuron.

3.2 Neuron-based back-calculation structure

The proposed structure uses neurons in substitution of the gain k_t from the classical back-calculation structure (see Fig. 4). This scheme can be thought as an intermediate case between classical back-calculation and conditional integration, since the neuron switches between two states, and modulates a continuous signal. The effect of the neuron gain, G_N, should be similar to that of classical gain, k_t. However, a total of three parameters can be tuned in the proposed strategy, which offers more flexibility than both conditional integration and classical back-calculation anti-windup schemes. In general, two neurons will be needed to encode both positive and negative values of e_s, although only one neuron can be used when saturation occurs systematically by just one of the limits. In addition, we expect that the recalculation of integral signal during saturation occurs in certain instants at which the neuron is fired instead of continuous recalculation. It should be highlighted that the proper selection of the parameters could lead to minimal distortion of the PID controller action when the neuron is off.

4. ILLUSTRATIVE EXAMPLE OF APPLICATION

An example is formulated next to show the performance of the proposed neuron-based back-calculation strategy in comparison with other classical anti-windup strategies.

4.1 Problem formulation

The plant to be controlled is modeled as a first order transfer function:

$$P(s) = \frac{1.5}{20s + 1} \quad (6)$$

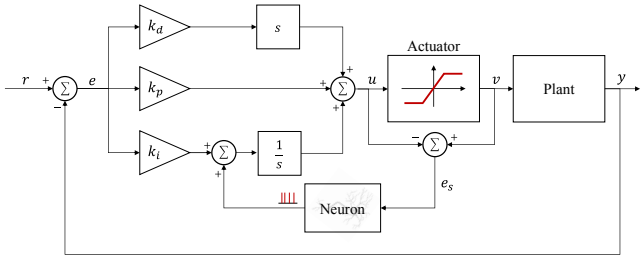

Fig. 4. Structure of back-calculation with neuromorphic feedback.

The actuator is subject to a saturation with the following limits:

$$[u_{min}, u_{max}] = [-0.8, +0.8] \quad (7)$$

A proportional-integral (PI) controller is designed to control the system for the ideal case, i.e., considering that the saturation limits are not reached in normal operation. It is designed using the root locus as

$$C(s) = k_p + \frac{k_i}{s} = 17.69 + \frac{6.32}{s} \quad (8)$$

4.2 Control scheme

Simulations are performed in Simulink. Fig. 5 shows the neuron-based back calculation structure designed for this application example (top) and the neuron subsystem block (bottom). Notice that each neuron has a rectifier at its input, and that the input to the second neuron is inverted in order to administrate a positive input. Likewise, the output of the second neuron is fed to the inverting input of an adder block to produce a negative pulse train.

4.3 Simulation results

Firstly, the response of the closed loop system without saturation to an input of 0.5 is compared with the case considering the saturation limits. The output of the system in both cases are plotted in Fig. 6. It is observed that the

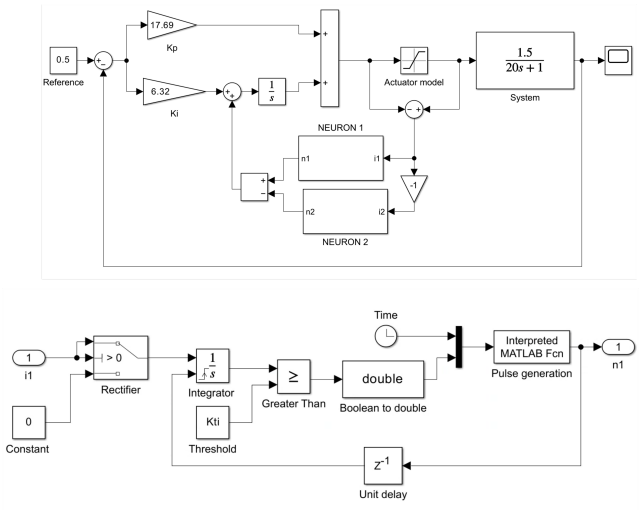

Fig. 5. Simulink model of: neuron-based back-calculation anti-windup control scheme (top), and neuron subsystem (bottom).

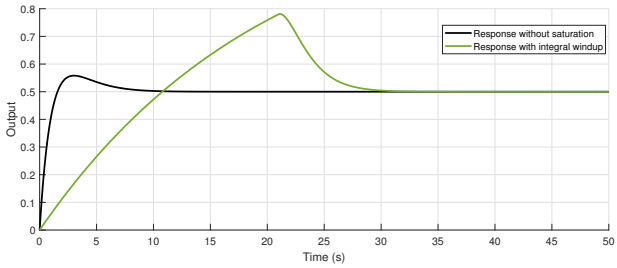

Fig. 6. System output without and with saturation in absence of anti-windup mechanism.

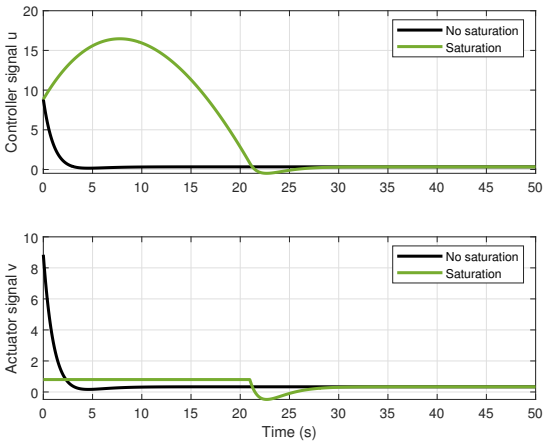

Fig. 7. Controller (top) and actuator (bottom) signals without and with saturation in absence of anti-windup mechanism.

overshoot is largely increased from 10% to 56%, as well as the settling time from 10 s to 30 s.

In Fig. 7, the controller output, u, (top) and the actuator signal, v, (bottom) are plotted in both cases: with and without saturation. It is observed that the actuator signal without saturation begins above 8 and suddenly decreases. Meanwhile, when the saturation is considered, the signal of the actuator is limited to 0.8 for almost 20 s, which causes the slow rising transient shown in previous figure. Additionally, it produces the controller to increase its integral part during this saturated state causing the overall to increase as well. This situation leads to an increased overshoot as it takes more time to reduce integral part once the reference has been surpassed.

Results introducing the traditional back-calculation feedback path are shown and commented next. In Fig. 8, the system output when varying k_t from 0 (without back-calculation) to 1 is shown. It is seen how it is possible to adjust both the overshoot and the settling time by tuning this parameter as follows: lower values of k_t causes the response to be closer to the case without back-calculation, i.e., higher overshoot and settling time. As the value of k_t is increased, it is possible to damp the response and eliminate the overshoot. Values of k_t higher than 1, in this example, can further slow down the response, leading to an increase in settling time.

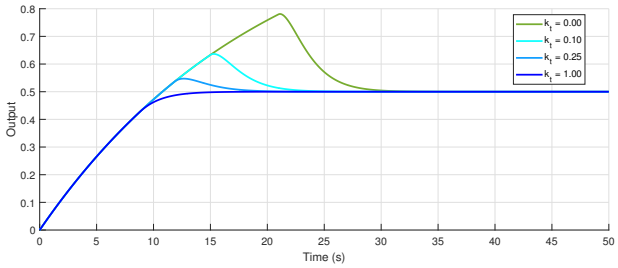

Fig. 8. Output curves for different k_t values using classical back-calculation anti-windup mechanism.

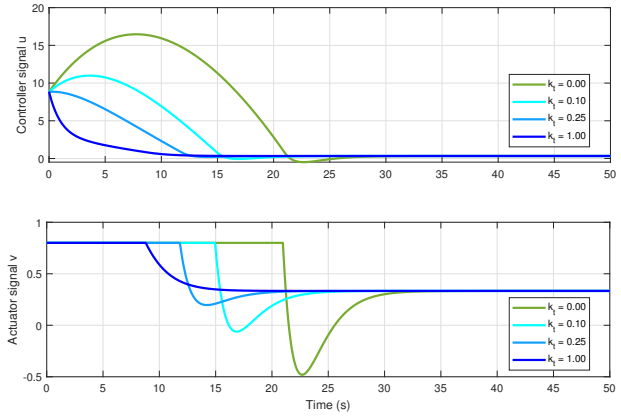

Fig. 9. PI controller output, u, (top) and actuator output, v (bottom) for different values of k_t using classical back-calculation anti-windup mechanism.

Controller (top) and actuator (bottom) outputs when considering traditional back-calculation are plotted in Fig. 9. It is observed that, although saturation persists in order to achieve the minimum rising time, the actuator output leaves saturation earlier as k_t is increased, reducing its settling time. The controller signal is more similar to saturation-less case as k_t is increased because integral part is reduced, which explains the reduction in overshoot.

In the following simulations, the results using the neuron-based back-calculation anti-windup strategy are shown and compared with the previous cases. Tuning of neuron parameters is analyzed. First, in Fig. 10, the output changes due to changes in the value of t_h are shown. The simulations were performed with $A = 20$, and $K_{ti} = At_h/0.25$ ($G_N = 0.25$). For narrow pulses $t_h \leq 0.01$ s, the output is very similar to the continuous case shown previously. In this case, the neuron acts as a good modulator that introduces little distortion. As t_h is increased, although preserving the neuron gain, the pulses cause the output to lose smoothness, in general. For $t_h = 0.15$ s, the overshoot is eliminated. The value $t_h = 0.2$ s causes and undershoot. Further increases, such as $t_h = 1$ s, produce the output to deviate from the saturation path earlier when compared to the previous cases, causing a backward movement that has not been observed in the traditional back-calculation structure.

Second, Fig. 11 shows the effects on the controlled system of changing parameter A. As can be seen, for values close to $A = 20$, the neuron is able to reduce windup as in

the continuous case, only influenced by the value of G_N. Nonetheless, if the amplitude is largely increased up to 500, the structure eliminates the overshoot. Further increases cause undershoot, as in the case with $A = 1000$. It is observed that an increase of A has similar consequences as increasing t_h, but it is somehow more robust as it has to bee greatly changed in order to produce significant differences.

Lastly, three different values of the parameter K_{ti} are considered in Fig. 12, while A and t_h are chosen to provide a smooth response ($t_h = 0.01$ s and $A = 20$). From eq. (5), it is known that, as K_{ti} is varied, G_N is varied in the inverse proportion. Only slight differences can be found between the three plotted curves in comparison with the continuous case, meaning that the behavior of K_{ti} is closer to the behavior of $1/k_t$ than the other two parameters.

Once the influence of neuron parameters has been observed for the system output, the differences in the feedback signal error are shown next. Fig. 13 contains three different simulations: one case with the classical structure and two cases of the neuron-based strategy. Neuron parameters in common in these two cases are: $A = 20$ and $G_N = 0.25$. The width of the pulses is different: $t_h = 10$ ms in one case and $t_h = 500$ ms in the other. These two cases are expected to be representative because the variation in A has similar effects to the variation of t_h on output, as observed previously. The first obvious difference between the simulated cases is that the traditional structure produces a continuous curve, while the information is encoded into pulses in the neuromorphic structure. Secondly, when the pulse duration (or amplitude) is increased, less pulses are fired. In this case, only two pulses are fired. The second pulse occurs closely to the last instant when the continuous case leaves saturation. In other words, it can recompute the integral part immediately before the moment at which controller should leave the saturation state, which is reminiscent of the behavior of conditional integration anti-windup schemes.

Fig. 14 shows the controller signal that receives the actuator. It is observed that the discontinuities are only produced by the neuron when the limits are surpassed. On average, the performance is the same, but as t_h (or A) is increased, the observed ripple increases.

In the following figures, the results of using conditional integration are shown. Fig. 15 shows the output using conditional integration structure using the condition $e_s \neq 0$. It is observed that it produces no overshoot and behaves

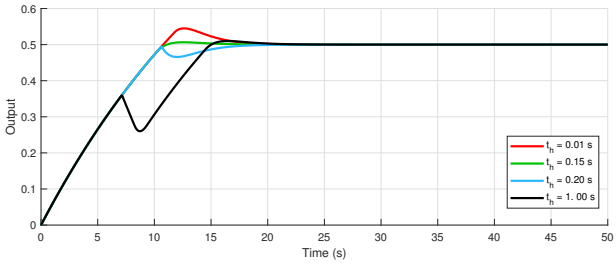

Fig. 10. Output curves for different t_h values. $A = 20$, $G_N = 0.25$ when using neuron-based back-calculation.

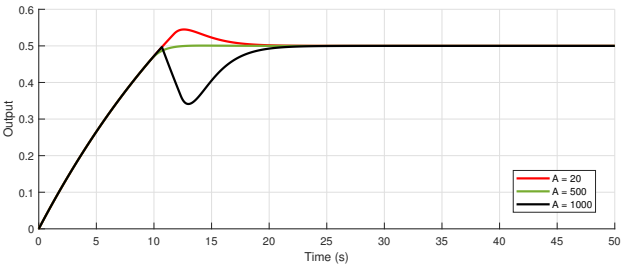

Fig. 11. Output curves for different A values. $t_h = 10$ ms, $G_N = 0.25$ when using neuron-based back-calculation.

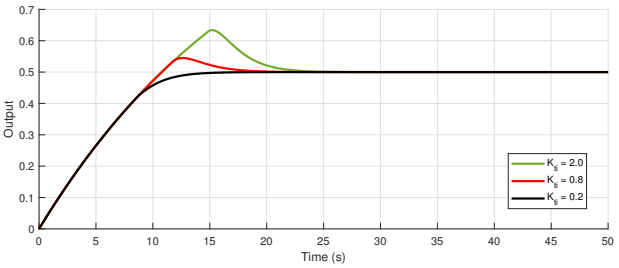

Fig. 12. Output curves for different K_{ti} values. $A = 20$, $t_h = 10$ ms when using neuron-based back-calculation.

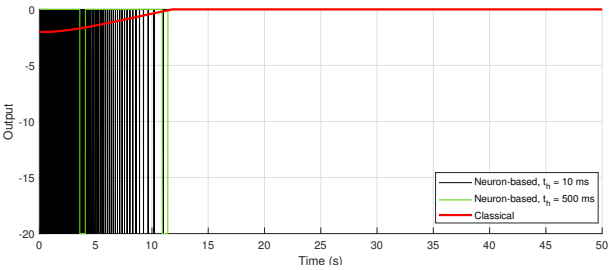

Fig. 13. Back-calculation feedback signal (e_s) in classical and two cases of neuron-based structures ($k_t = G_N = 0.25$).

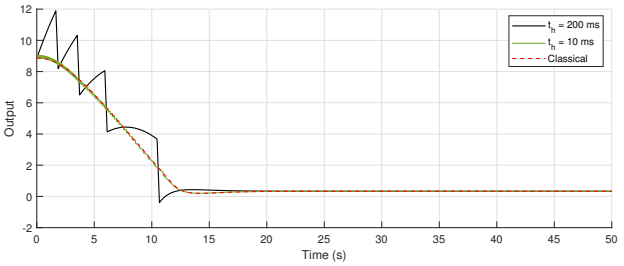

Fig. 14. Controller output in classical and two cases of neuron-based back-calculation.

similarly to the case of traditional and neuron-based back-calculation structure when $k_t = 1$ and $G_n = 1$, respectively. Neither the overshoot nor the speed can be adjusted which can be seen as an inconvenient to certain applications. Finally, the controller output is plotted in Fig. 16. This signal differs from both traditional and neuron-based back-calculation because the integral part is not activated during the first 10 s and, therefore, only the

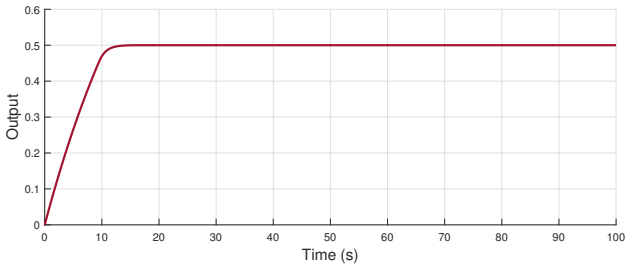

Fig. 15. Output with the conditional integration strategy.

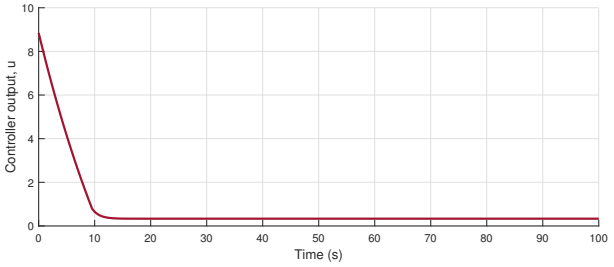

Fig. 16. Controller output with conditional integration strategy.

proportional part is being computed. As soon as $e_s = 0$, the integral part is activated and produces a smooth transient response with zero steady state error.

5. CONCLUSION

This paper has explored the possibility of using neuromorphic principles as anti-windup strategies. It has been observed that the introduction of neurons in the anti-windup loop, with the proper selection of their parameters, produce similar results to the back-calculation algorithm. In addition, if amplitude or width of the pulses of the neuron are increased, the number of instants at which the controller signal has to be recomputed becomes minimal. Nonetheless, it has been seen that excessive values of these parameters can cause an undesired backward motion in the output.

The scheme proposed here was compared with two classical anti-windup strategies, namely classical back-calculation and conditional integration. The neuron-based back-calculation can be interpreted as an anti-windup mechanism that has elements in common with both strategies, and provides more flexibility.

In future work, the use of neurons to encode the controller signal together with the back-calculation strategy will be explored. It could serve to obtain a controller that can simultaneously handle friction-limited systems and saturation-constrained actuators using neuromorphic principles. Potential benefits of using neurons include noise robustness and power efficiency.

REFERENCES

Åström, K. and Hägglund, T. (2006). *Advanced PID Control*. ISA - The Instrumentation, Systems and Automation Society.

da Silva, L.R., Flesch, R.C.C., and Normey-Rico, J.E. (2018). Analysis of anti-windup techniques in PID control of processes with measurement noise. *IFAC-PapersOnLine*, 51(4), 948–953.

DeWeerth, S.P., Nielsen, L., Mead, C.A., and Åström, K.J. (1991). A simple neuron servo. *IEEE Transactions on Neural Networks*, 2(2), 248–251.

Indiveri, G. (1999). Neuromorphic analog vlsi sensor for visual tracking: circuits and application examples. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(11), 1337–1347.

Indiveri, G., Linares-Barranco, B., Hamilton, T.J., Schaik, A.v., Etienne-Cummings, R., Delbruck, T., Liu, S.C., Dudek, P., Häfliger, P., Renaud, S., Schemmel, J., Cauwenberghs, G., Arthur, J., Hynna, K., Folowosele, F., Saighi, S., Serrano-Gotarredona, T., Wijekoon, J., Wang, Y., and Boahen, K. (2011). Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5, 73. doi: 10.3389/fnins.2011.00073.

Krauhausen, I., Koutsouras, D.A., Melianas, A., Keene, S.T., Lieberth, K., Ledanseur, H., Sheelamanthula, R., Giovannitti, A., Torricelli, F., McCulloch, I., Blom, P.W.M., Salleo, A., van de Burgt, Y., and Gkoupidenis, P. (2021). Organic neuromorphic electronics for sensorimotor integration and learning in robotics. *Science Advances*, 7(50), eabl5068. doi:10.1126/sciadv.abl5068.

Mead, C.A. (1989). *Analog VLSI Implementation of Neural Systems*. Springer US.

Schuman, C.D., Potok, T.E., Patton, R.M., Birdwell, J.D., Dean, M.E., Rose, G.S., and Plank, J.S. (2017). A survey of neuromorphic computing and neural networks in hardware. *ArXiv*, abs/1705.06963.

Serrano, A.J., Vinagre, B.M., and Tejado, I. (2023). Fractional neuromorphic controller for a servomotor: Analog implementation. *IFAC-PapersOnLine*, 56(2), 4301–4306. doi:10.1016/j.ifacol.2023.10.1799. 22nd IFAC World Congress.

Serrano-Balbontín, A.J., Tejado, I., Mancha-Sánchez, E., and Vinagre, B.M. (2023a). Introducing fractional order dynamics in neuromorphic control: application to a velocity servomotor. In *2023 European Control Conference (ECC)*, 1–6. doi:10.23919/ECC57647.2023.10178322.

Serrano-Balbontín, A.J., Tejado, I., and Vinagre, B.M. (2023b). Fractional integrate-and-fire neuron: Analog realization and application to neuromorphic control. In *2023 International Conference on Fractional Differentiation and Its Applications (ICFDA)*, 1–6. doi: 10.1109/ICFDA58234.2023.10153307.

Tarbouriech, S. and Turner, M. (2009). Anti-windup design: an overview of some recent advances and open problems. *IET Control Theory & Applications*, 3(1), 1–19.

Tzafestas, S. and Frangakis, G. (1980). Design and implementation of pulse frequency modulation control systems. *Transactions of the Institute of Measurement and Control*, 2(2), 65–78. doi:10.1177/014233128000200202.

Visioli, A. (2006). *Practical PID Control*, chapter Anti-windup Strategies, 35–60. Springer-Verlag London.

Zaccarian, L. and Teel, A.R. (2011). *Modern Anti-windup Synthesis: Control Augmentation for Actuator Saturation*. Princeton University Press.