

ANTI-WINDUP METHOD FOR FUZZY PD+I, PI AND PID CONTROLLERS APPLIED IN BRUSHLESS DC MOTOR SPEED CONTROL

J. R. B. A. Monteiro, W. C. A. Pereira, M. P. Santana, T. E. P. Almeida,
G. T. Paula and I. Santini

School of Engineering of São Carlos
University of São Paulo
São Carlos, SP 13566-590
Email: jrm@sc.usp.br

Abstract—This paper proposes a new anti-windup method for integrative portion of many controllers: PI, PID and fuzzy PD+I. The proposed method does not require any coefficient for the anti-windup method, as other known anti-windup methods does. All the deduction and development of the proposed anti-windup method for PI, PID and fuzzy PD+I are presented here, resulting in two general and simple anti-windup equivalent methods. Following, the application of the proposed anti-windup method in a brushless DC motor speed control, employing a fuzzy PD+I controller, is shown and compared to a traditional anti-windup method, which achieves better results than that traditional method.

Keywords—anti-windup integrative method, brushless DC motors, proportional integrative derivative control, fuzzy PD+I control

I. INTRODUCTION

Electrical machines are indispensable elements in contemporary world, from processes industry to home applications. Electrical motors have innumerable advantages over other kind of motors, not limited to, but including: low cost, high power density, simple construction and installation requirements, robustness, versatility, so it can be easily adapted to various types of loads, high efficiency, control simplicity, thus electrical motors are widely used in industrial applications [1].

Among the various types of electrical motors, direct-current motors are very attractive and widely used in variable speed applications. However, its brushes are a source of disadvantages, increasing machine operational cost due to maintenance, efficiency decrease due to losses in the mechanical commutator and brushes themselves, noise, etc. In this way, brushless DC motors come to address some of those issues. Perhaps, usually their cost for equivalent machines are higher, due to the fact that such machines are a set of a permanent magnet synchronous machine and a static electric converter [2]. If the cost of such a machine is critical in an application, another kinds of machine can be used, as induction motors and switched reluctance motors [3][4]

A. Fuzzy Controllers

Zadeh presents a new controller, named Fuzzy Controller, in 1965, which is based on fuzzy logic. That controller has high efficiency if applied to many kinds of non-linear systems, since other linear controllers, like PID controller,

have some problematic issues in plants which have some kind of non-linearity during its operation [4].

Fuzzy controllers are based on a set of knowledge represented by fuzzy set theory, where is characterized by three main aspects: it works using linguistic concepts, which avoid the necessity of a precise plant model; it is a non-linear controller, allowing the possibility of plant non-linearity compensation; it is robust to parametric variations of the plant.

Many fuzzy controllers structures are used in the motor speed control area, among them, one of the most used structure is fuzzy-PD [5][6].

A block diagram of a fuzzy-PD controller is shown in Fig. 1. There, the error e_k and its variation in time (d/dt) are the inputs for a fuzzy controller. That inputs are multiplied by constants (k_P and k_D), chosen properly by the designer, and are then applied to a set of fuzzy rules in order to the controller output be evaluated, which is then multiplied by the output constant (k_U) and used as the control action, as an input to the plant.

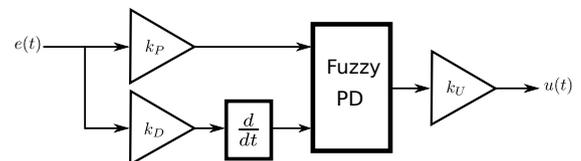


Fig. 1. Block diagram of fuzzy PD controller.

Although this kind of controller is very robust to parameter variations, it still presents an error in steady state operation. There are some controllers configurations to eliminate this error, using fuzzy controllers, which includes fuzzy incremental controllers, as in [7], and fuzzy PD controllers combined to an integrative controller, such as fuzzy PD+I controller, which is very common structure (Fig. 2). However, the incorporation of a parallel integrative action with a fuzzy PD controller leads to a very known problem related to the integrative action, i. e., the overshoot in plant output.

B. Linear controllers

During the past years, many theories involving optimal control and robust control have given important results, although some special architectures use the classical PID topology. This topology has simplicity, robustness and is very effective in many industrial processes. The continuous form of such control is given in (1).

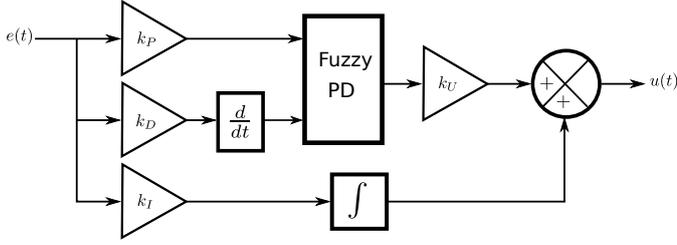


Fig. 2. Block diagram of fuzzy PD+I controller.

$$u(t) = k_P e(t) + k_I \int_0^t e(\zeta) d\zeta + k_D \frac{d}{dt} e(t) + u(0) \quad (1)$$

Where:

- $u(t)$: controller output control action;
- $e(t)$: error value, the difference between the reference value and measured value;
- k_P : proportional gain;
- k_I : integrative gain;
- k_D : derivative gain;
- $u(0)$: controller output initial value, normally this initial value is 0 ($u(0) = 0$).

The integrative term of (1) is essentially used to keep system steady state error equal to zero. However, the controller can lead to a saturation state due to excessive error accumulation in the integrative term, causing a high value of output plant overshoot [8]. Therefore, an anti-windup method is used in such situations. Literature are plenty of anti-windup methods in order to minimize output plant overshoot.

C. Anti-windup methods

In this way, some specialized commercial software have some of these methods implemented inherently, as the case of back-calculation, integrative clamping and feedforward methods, implemented in the software MATLAB [9]. Those methods require additional parameters in order to operate, i. e., designer must choose parameters that will be used by these methods, besides the parameters of the controller itself. Other anti-windup methods will also use extra parameters, as conditional integration and integral state prediction [10].

II. PROPOSED TOPOLOGY

The proposed topology regards to a fuzzy PD+I, PID and PI controllers, perhaps it is better understood if taken as a base a simple integrative anti-windup method.

A. Anti-windup method for integrative controller

The time continuous equation for integrative controllers is given by (2).

$$u(t) = k_I \int_0^t e(\tau) d\tau + u(0) \quad (2)$$

The discrete time equation for integrative controllers is given by (3). In order to introduce anti-windup method of that controller, the output of the equation is passed through a saturation function given by (4). The controller output value

not saturated is referred by u'_k whereas the controller output clamped value is referred by u_k .

$$u'_k = k'_I e_k + u_{k-1} \quad (3)$$

Where:

$$k'_I = k_I T_s$$

u'_k : non saturated output control action.

$$S_{at}(x, X_{MAX}, X_{MIN}) =$$

$$= \begin{cases} X_{MAX} & \text{if } x > X_{MAX}, \\ X_{MIN} & \text{if } x < X_{MIN}, \\ x & \text{if } x \geq X_{MIN} \text{ and } x \leq X_{MAX}. \end{cases} \quad (4)$$

Where:

- X_{MAX} : Maximum output value of x (eg. if x is the machine stator current, this is the maximum stator current allowed for that machine);
- X_{MIN} : Minimum output value of x (eg. if x is the machine stator current, this is the minimum stator current allowed for that machine, which can be equal to $-X_{MAX}$, for symmetrical situations).

$$u_k = S_{at}(u'_k, U_{MAX}, U_{MIN}) \quad (5)$$

The anti-windup action is simple, as the value of previous step, u_{k-1} , is passed through the saturation function, as well as the current value is (5), it will not accumulate values above the maximum (or minimum) values. The above equations can be graphically represented by diagram shown in Fig. 3.

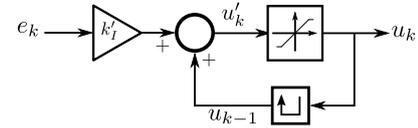


Fig. 3. Graphical representation of simple integrative controller anti-windup method.

B. Anti-windup method for proportional integrative controllers

The time continuous equation for a proportional integrative controller is given by (6).

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + u(0) \quad (6)$$

It is assumed that $e(0) = 0$, thus the proportional integrative controller equation for discrete time is shown in (7).

$$u_k = k_P (e_k - e_{k-1}) + k'_I e_k + u_{k-1} \quad (7)$$

One way to apply an anti-windup method is applying saturation function (4) in controller output value (u_k), as done in the integrative controller. However, this can lead to a little collateral effect: if the proportional action by itself leads to a value higher than (or lower than) the maximum

value (or minimum value), the proportional action will be also clamped, what will lead to errors in the next iterations.

Therefore, to correctly apply an anti-windup method, in the same flavor of integrative one, the saturation action applied in u_k must be restricted only to the integrative action. If the saturation function is applied in the controller output value (u_k), it will affect not only the integrative action, but also the proportional action, so the proportional action must be compensated. Defining the proportional action by a_k (8), the compensation due to the saturation of output is Δa_k , which is given by (9). The value of this difference of the previous iteration is then added to output value of controller in (10). The output value is eventually limited in (11).

$$a_k = k_p e_k \quad (8)$$

$$\Delta a_k = a_k - S_{at}(a_k, U_{MAX}, U_{MIN}) \quad (9)$$

$$u'_k = a_k - a_{k-1} + k'_I e_k + \Delta a_{k-1} + u_{k-1} \quad (10)$$

Where a_{k-1} is the proportional and derivative contributions of the previous step.

$$u_k = S_{at}(u'_k, U_{MAX}, U_{MIN}) \quad (11)$$

The proposed method is graphically represented in Fig. 4. Considering that it is possible to split the difference between a_{k-1} and Δa_{k-1} from the global adder block, the delays applied to both values can be applied in the difference of current values of a_k and Δa_k , so the method can be simplified to that shown in Fig. 5 and a variation of this topology can be viewed in Fig. 6.

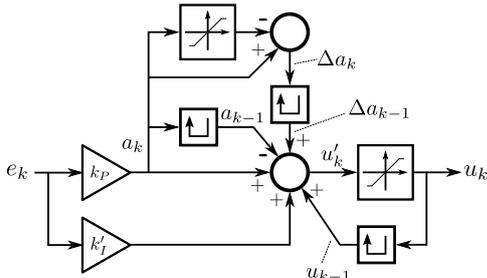


Fig. 4. Proposed anti-windup method for proportional integrative controllers.

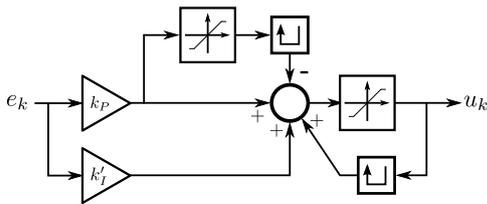


Fig. 5. Proposed anti-windup method for proportional integrative controllers. It is derived from the method of Fig. 4.

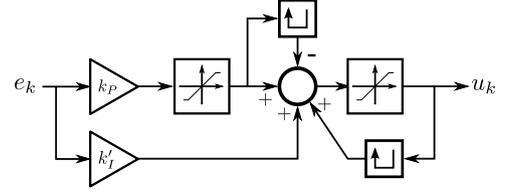


Fig. 6. A variation of the proposed anti-windup method for proportional integrative controllers.

C. Anti-windup method for proportional derivative integrative controllers

The time continuous equation for a proportional derivative integrative controller is given by (12) and its discrete form is given by (7).

$$u(t) = k_P e(t) + k_I \int_0^t e(\zeta) d\zeta + k_D \frac{d}{dt} e(t) + u(0) \quad (12)$$

$$u_k = k_P(e_k - e_{k-1}) + k'_D(e_k - 2e_{k-1} + e_{k-2}) + k'_I e_k + u_{k-1} \quad (13)$$

Where:

$$k'_D = \frac{k_D}{T_s}$$

$$u_k = k_P e_k + k'_D(e_k - e_{k-1}) - (k_P e_{k-1} + k'_D(e_{k-1} - e_{k-2})) + k_I T_s e_k + u_{k-1} \quad (14)$$

Considering the current proportional and derivative contributions as a_k , as in (15), the previous value of this component must be passed through the saturation function, as the case of PI controller, then PID equation (16) is evaluated and its output is passed through the saturation function again, as in (11).

$$a_k = k_P e_k + k'_D(e_k - e_{k-1}) \quad (15)$$

The PID controller discrete equation can be written as:

$$u'_k = a_k - a_{k-1} + k_I T_s e_k + u_{k-1} \quad (16)$$

The proposed anti-windup method for PID controller is shown in Fig. 7. Figure clearly show the a_k (proportional and derivative) contribution is saturated before being added to integrative contribution, so in the next step, it will not decrease more than its contribution.

D. Anti-windup method for Fuzzy PD+I controllers

The literature are plenty of Fuzzy controllers, however those used to emulate a PID controller are absent of an anti-windup method.

The proposed anti-windup for Fuzzy PD+I controllers is directly derived from that of PID anti-windup. In the place of a_k , in (16), the current Fuzzy PD output value is used (F_k). This output value is limited to the maximum and minimum values, so in this case, saturation function is not necessary for that output, as it is in Figs. 5, 6 and 7, if the value limits of

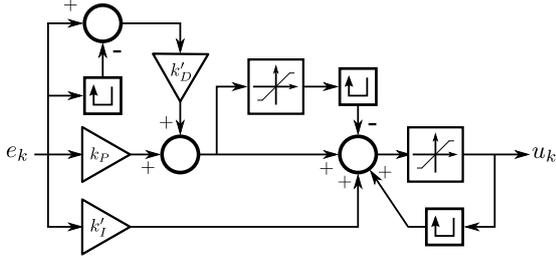


Fig. 7. Proposed anti-windup method for proportional integrative derivative controller.

fuzzy PD block (multiplied by k_u) are the same or below the limits of the output saturation function. Nevertheless, the saturation function is necessary in the output, which has the contribution of the integrative function. Thus the controller output u_k is the clamped value of u'_k given in (17), like (11). Thus, the resulting method for a fuzzy controller (in this case a fuzzy PD controller) is shown in Fig. 8, which is based on Fig. 5, and a variation can be based on Fig. 6, in the same way, given the same results. However, if the output value limits of fuzzy PD block (multiplied by k_u) are above the value limits of the output saturation function, it is necessary, like in PI and PID anti-windup block diagram controllers, to use a saturation function in the fuzzy PD output, as shown in Fig. 9.

$$u'_k = F_k - F_{k-1} + k_I T_s e_k + u_{k-1} \quad (17)$$

Where:

F_k : current output value of Fuzzy controller;
 F_{k-1} : last output value of Fuzzy controller;

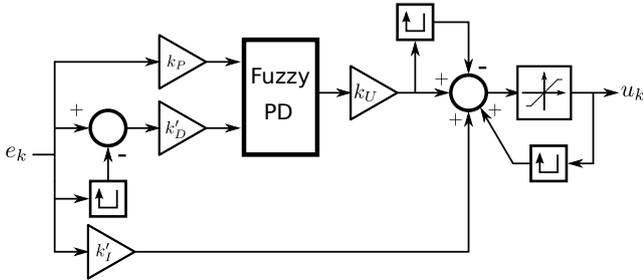


Fig. 8. Proposed anti-windup method for Fuzzy PD+I controller.

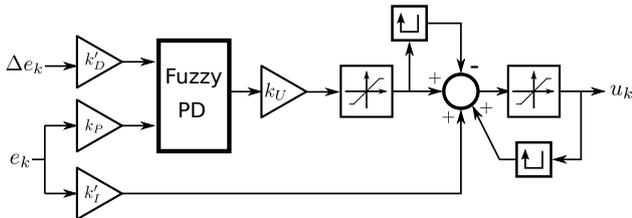


Fig. 9. Proposed anti-windup method for Fuzzy PD+I controller using additional saturation block for fuzzy PD block output, in the case of different output limits of fuzzy PD block and the output saturation block.

III. BRUSHLESS DC MOTOR ELECTRICAL DRIVE

The term brushless DC motor is the composition of an electrical machine, more precisely a surface-mount permanent magnet synchronous machine, with its electric converter, commonly a three phase machine with a three phase electric converter (a three phase inverter) [11][2]. Ideally, the electrical machine has a trapezoidal back-EMF waveform and with a 120° square wave stator current produces an almost ripple free electromagnetic torque, as in Fig. 10.

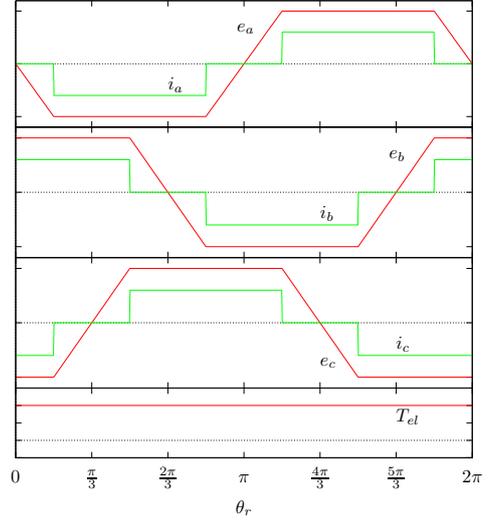


Fig. 10. Brushless DC motor ideal electromagnetic torque generation.

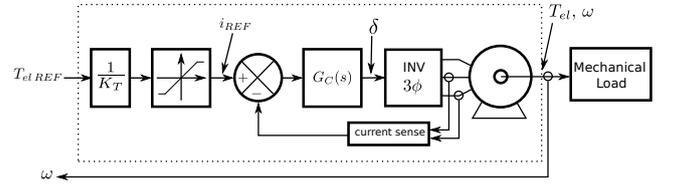


Fig. 11. The electrical drive, composed by a brushless DC motor and a current control loop, in order to control machine electromagnetic torque, and the mechanical load.

Fig. 12 shows the electromagnetic torque response of used electric drive, a permanent magnet synchronous machine, which its parameters are shown in Table I, fed by a three phase inverter, operating in six-step mode.

The machine model is shown in (18) to (21).

$$\begin{bmatrix} u_a \\ u_b \\ u_c \end{bmatrix} = \begin{bmatrix} L_s & M_s & M_s \\ M_s & L_s & M_s \\ M_s & M_s & L_s \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + R_s \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} + \begin{bmatrix} u_n \\ u_n \\ u_n \end{bmatrix} \quad (18)$$

Where:

e_a , e_b and e_c : induced voltage of stator phases a, b and c, respectively, due to rotor magnets movement, as in (19);

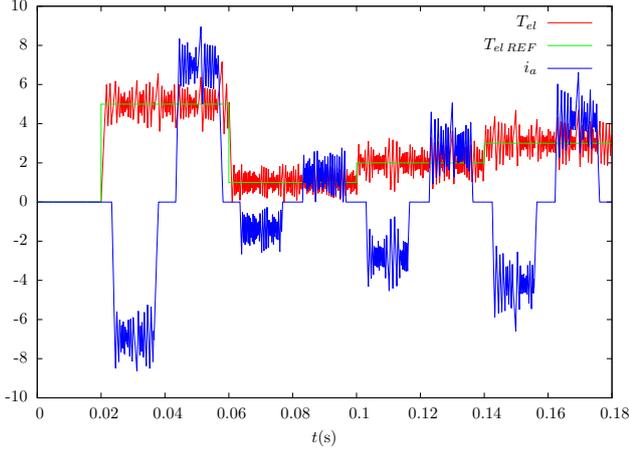


Fig. 12. Electromagnetic torque response of the electric drive shown in Fig. 11.

- i_a, i_b and i_c : stator phase currents a, b and c, respectively;
 L_s : stator phase self-inductance;
 M_s : stator phases mutual inductances;
 R_s : stator phase resistance;
 u_a, u_b and u_c : a, b and c stator phases applied voltages, respectively;
 u_n : stator neutral terminal voltage (this terminal is not normally connected).

$$\begin{bmatrix} e_a \\ e_b \\ e_c \end{bmatrix} = \frac{d}{dt} \begin{bmatrix} \Phi_{ra} \\ \Phi_{rb} \\ \Phi_{rc} \end{bmatrix} = \omega_r \begin{bmatrix} \Phi'_{ra} \\ \Phi'_{rb} \\ \Phi'_{rc} \end{bmatrix} \quad (19)$$

Where:

- Φ_{ra}, Φ_{rb} and Φ_{rc} : linked magnetic fluxes between rotor magnets and stator winding phases a, b and c, respectively;
 ω_r : electrical rotor speed.

$$T_{el} = n_{pp} (\Phi'_{ra} i_a + \Phi'_{rb} i_b + \Phi'_{rc} i_c) \quad (20)$$

Where:

- T_{el} : machine-generated electromagnetic torque;
 n_{pp} : number of machine's pole pairs;

From (19), it is possible to derive:

$$\begin{bmatrix} \Phi'_{ra} \\ \Phi'_{rb} \\ \Phi'_{rc} \end{bmatrix} = \frac{1}{\omega_r} \frac{d}{dt} \begin{bmatrix} \Phi_{ra} \\ \Phi_{rb} \\ \Phi_{rc} \end{bmatrix} \quad (21)$$

The dynamic mechanical load equation is shown in (22).

$$J \frac{d\omega_m}{dt} + B\omega_m + T_L = T_{el} \quad (22)$$

Where:

- B : equivalent frictional coefficient, composed by rotor shaft bearings and load frictional losses;
 J : combined inertia momentum of machine rotor and load;

T_L : load torque;

ω_m : rotor mechanical speed.

The machine parameters as well as mechanical load parameters are in Table I, load torque (T_L) is not show because it is different in each simulation.

TABLE I
BLDC motor and mechanical load parameters used in simulations.

Motor	Load
$R_s = 2.3\Omega$	$J = 4.2 \cdot 10^{-3} \text{kg m}^2$
$(L_s - M_s) = 12.5 \text{mH}$	$B = 3.032 \cdot 10^{-3} \text{kg m}^2/\text{s}$
$n_{pp} = 3$	
$\Phi_m = 0.12 \text{Wb}$	

IV. FUZZY SPEED CONTROLLER

The speed control topology is shown in Fig. 13, where the fuzzy controller is used in the speed control loop. The output value of this controller is the torque reference for electrical drive, shown in the previous section. As the output value of fuzzy controller is limited, thus the electromagnetic torque reference to electric drive is also limited, which means a limit in stator current applied to the machine, or better, the maximum allowed stator current for the machine. This current value is about 5A, giving 3.6Nm of maximum electromagnetic torque.

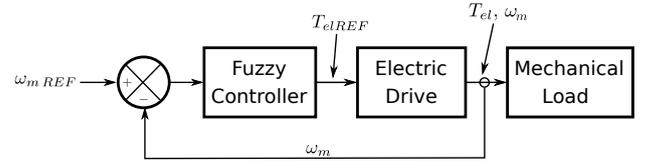


Fig. 13. Block diagram of BLDC motor speed control using fuzzy controller in speed control loop.

The fuzzy controller referred in Fig. 13 is a fuzzy PD+I controller, as shown in Fig. 2. The fuzzy PD membership functions are shown in Fig. 14 and rules table is shown in Table II.

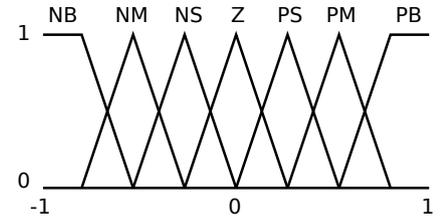


Fig. 14. Used fuzzy membership functions.

V. RESULTS

A very simple anti-windup method is the conditional integration method, where the integrative portion of controller is activated only when the absolute value of the error is bellow of a determined value [10]. Some results considering different values of speed errors ($\Delta\omega_m$) are shown in Fig. 15, which means that the integrative portion of controller is activated only when the absolute value of error is bellow

TABLE II
Fuzzy table for used controller.

e(t)	de(t)/dt						
	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	Z
NM	NB	NB	NB	NM	NS	Z	PS
NS	NB	NB	NM	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PM	PB	PB
PM	NS	Z	PS	PM	PB	PB	PB
PB	Z	PS	PM	PB	PB	PB	PB

5, 2, and 1rd/s. In that figure, a very good result is achieved using 2rd/s, however if the error limit of 1rd/s is used, the rotor speed error does not fall below 1rd/s only with fuzzy PD controller action, so the integrative action of the controller will not be ever activated. The proposed anti-windup (PAWU) method gives an intermediate response. The used values for k_P , k_D , k_u and k_I for fuzzy controller are 0.1, 0.01, 4.5 and 250, and they were not optimized, as the focus of this work is the comparison between different anti-windup methods.

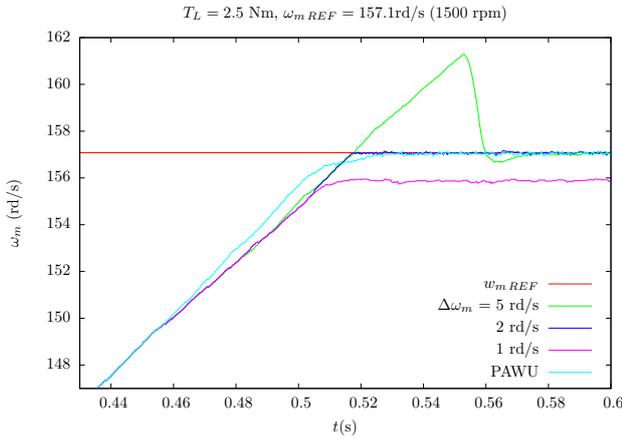


Fig. 15. Rotor speed versus time for load torque of $T_L = 2.5\text{Nm}$ and $\omega_{m_REF} = 157.1\text{rd/s}$ (1500rpm), using activation of integrative controller when the absolute speed error is below some values ($\Delta\omega_m$) and using proposed anti wind up method (PAWU).

Considering the curve denoted by “1rd/s” in Fig. 15, it consists only by the fuzzy PD output, as the output error does not fall below the value to activate the controller integrative action. So, it is possible to note that integrative action of the proposed method is gradually activated after around 0.45s, and integrative actions for other curves are activated according to programmed value of $\Delta\omega_m$.

Another anti-windup method is the back calculation method, available in MATLAB software [9]. There, the designer must choose the value of a constant coefficient k_b . This coefficient, like the error value of example shown in Fig. 15, must be chosen by control designer and can be optimal for a given situation, not for all possible situations. The method works as follows: the non-saturated output of the controller ($u'(t)$) is subtracted from the saturated output ($u(t)$) and multiplied by k_b . This value is then used to subtract the the input of the main integrator as in (23) [9].

$$u'(t) = k_P e(t) \int (k_I e(t) - k_b(u'(t) - u(t))) dt \quad (23)$$

$$u(t) = S_{at}(u'(t), X_{MAX}, X_{MIN}) \quad (24)$$

As an example, two step responses are shown with different situations: in Fig. 16 and in Fig. 17, which must be analyzed together. In Fig. 16, the proposed anti-windup method is plotted against back calculation with k_b varying from $0.8k_I$ to $2.0k_I$ and with a load torque of $T_L=2.5\text{Nm}$; and in Fig. 17, the same rotor reference speed, but with no load torque (T_L). In the former, The response of back calculation method seems to be superior if compared with the proposed method, but in the latter case, those step responses present overshoot, and their responses are slightly inferior to the proposed method. In both cases, the response of proposed anti-windup method is almost the same, i. e., showing the same behavior with in different situations of operation. In the other hand, using back calculation method, the speed responses vary depending on external parameters, which means that the choice of k_b by the designer is an optimal issue that depends on the operating conditions, considering that in some applications the load parameters vary.

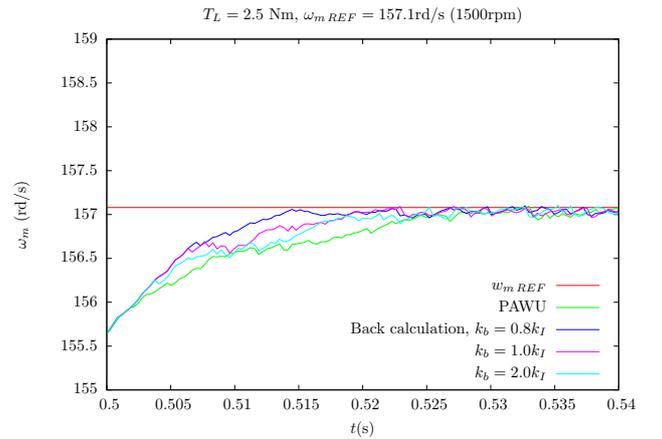


Fig. 16. Proposed (PAWU) and back calculation anti-windup methods for a step response of 157.1rd/s (1500rpm) with a load torque of 2.5Nm (T_L).

A. Considerations about the use of other kinds of motors

The used brushless DC motor electric time constant (τ_e) has a value much lower than the mechanical system time constant (τ_m), composed by the load and mechanical rotating parts of the motor. In this way, the pole of the mechanical system is dominant, far above the value of electrical system pole, so the electrical drive and the mechanical load shown in Fig. 11 can be approximated to a first order system [12].

Other kinds of motors can be used in spite of BLDC motor, shown in Fig. 11, since its electric converter must be changed properly and the condition that $\tau_e \ll \tau_m$ must be satisfied, in order to the proposed anti-windup method works as well as shown here.

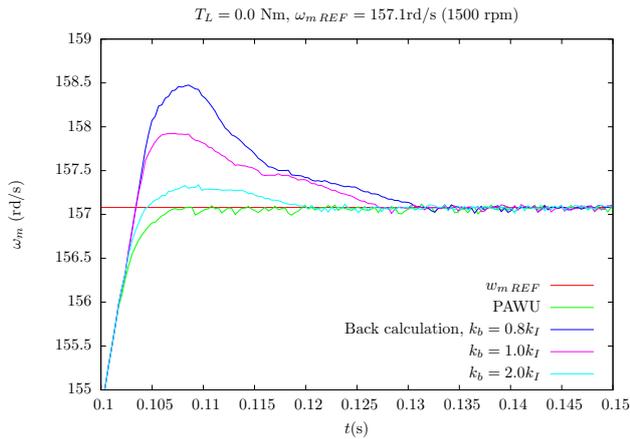


Fig. 17. Proposed (PAWU) and back calculation anti-windup methods for a step response of 157.1rd/s (1500rpm) with a no load torque ($T_L=0.0\text{Nm}$).

VI. CONCLUSIONS

The presented anti-windup method is simple, with a very low computational cost, and effective. It allows the integrative portion of the controller, be it fuzzy, PID or PI controllers, be activated only when the proportional or fuzzy action of the controller is not under saturation, without erroneous accumulation due to controller action saturation, as if compared to other methods, as Fig. 15. Also it does not need an extra coefficient or parameter, so the designer does not have the need to evaluate another parameter for the controller, neither needs to simulate different operational conditions to pick up the best value, as is the case of integrative activation shown in that figure.

Although back calculation method presents very good responses, as shown in Figs. 16 and 17, it needs an extra coefficient (k_b), and is computationally more heavy than the proposed anti-windup method. Also, opposed to the presented anti-windup method, back calculation dynamically modifies the integrative coefficient k_I , or the integrative effect, of the controller, what can be sometimes an undesirable behavior.

Additionally, the use of this method in speed controllers employing other kinds of motors, for example, induction motors or switched reluctance motors is also feasible, as discussed above, on the condition that electric time constant of the electrical drive is much lower than the mechanical system time constant, composed by machine mechanical parts and mechanical load.

ACKNOWLEDGMENT

The authors would like to acknowledge FAPESP, CAPES and CNPq, for the financial support given to this research.

REFERENCES

- [1] A. Fitzgerald, C. Kingsley, and S. Umans, *Electric Machinery*, ser. Electrical Engineering Series. McGraw-Hill Companies, Incorporated, 2002.
- [2] T. J. E. Miller, *Brushless Permanent-Magnet and Reluctance Motor Drives*. Oxford: Clarendon Press, 1993.
- [3] A. Vagati, A. Fratta, G. Franceschini, and P. Rosso, "AC motors for high-performance drivers: A design-based comparison," *IEEE Transactions on Industry Applications*, vol. 32, no. 5, 1996.
- [4] Z. Kovacic, *Fuzzy controller design: Theory and Applications*. Boca Raton, FL: CRC Press / Taylor & Francis, 2006.
- [5] T. C. Siong, B. Ismail, M. F. Mohammed, M. F. N. Tajuddin, S. Rafidah, A. Rahim, and Z. M. Isa, "Study of Fuzzy and PI controller for Permanent-Magnet Brushless DC motor drive," in *2010 4th International Power Engineering and Optimization Conference (PEOCO)*. IEEE, Jun. 2010, pp. 517–521.
- [6] T. C. Siong, B. Ismail, S. F. Siraj, M. F. Mohammed, and M. F. N. Tajuddin, "Implementation of Fuzzy Logic controller for permanent magnet brushless DC motor drives," in *2010 IEEE International Conference on Power and Energy*. IEEE, Nov. 2010, pp. 462–467.
- [7] R. Arulmozhiyal and R. Kandiban, "Design of fuzzy pid controller for brushless dc motor," in *Computer Communication and Informatics (ICCCI), 2012 International Conference on*, 2012, pp. 1–7.
- [8] P. L. Castrucci, A. Bittar, and R. M. Sales, *Controle Automático*. São Paulo: LTC, 2011.
- [9] K. Åström and T. Hägglund, *Advanced PID Control*. Research Triangle Park, August 2005.
- [10] X. Ian Li, J.-G. Park, and H.-B. Shin, "Comparison and evaluation of anti-windup PI controllers," *Journal of Power Electronics*, vol. 11, no. 1, pp. 45–50, 2011.
- [11] S. A. Nasar, I. Boldea, and L. E. Unnewehr, *Permanent magnet, reluctance, and self synchronous motors*. Boca Raton: CRC Press, 1993.
- [12] K. Ogata, *Modern Control Engineering*, 5th ed. Prentice Hall, September 2009.