eISSN: 2477-7994

JMECS (Journal of Measurements, Electronics, Communications, and Systems)

Vol.10, No.2, Pg.58-66, December 2023

A Method of Anti-Windup PID Controller for a BLDC-Drive System

Aditya Adni Argaloka[1], Harwin Aptadarya[1], Fiendo Mahendra Arentaka[1], Fiky Yosef Suratman[1], Arief Suryadi Satyawan[1,2,3]

[1] *School of Electrical Engineering, Telkom University, Bandung, 40287, Indonesia*

[2] *Faculty of Engineering, Nurtanio University, Bandung, 40174, Indonesia*

[3]*Research Center for Telecommunication, National Research and Innovation (BRIN), Bandung, 40135 Indonesia*

[*]*adityaadnii@student.telkomuniversity.ac.id*

Manuscript received August 19, 2023; revised October 1, 2023; accepted November 2, 2023.

Abstract

This research aims to enhance control systems for Brushless DC (BLDC) motors by introducing Proportional-Integral-Derivative (PID) control as a straightforward yet reliable solution, known for its precision, quick responsiveness, and stability. Emphasizing its suitability for BLDC motor speed control, the study addresses PID controller windup challenges, highlighting anti-windup techniques crucial for maintaining system stability. The primary focus is on improving the performance of an anti-windup PID controller for BLDC motor speed control in electric vehicles. Through simulations and analyses, the research aims to minimize steady-state error and overshooting, contributing to overall operational efficiency. Practical implementation involves optimizing the PID anti-windup controller's gain using the MATLAB PID Tuner and implementing it in the Arduino IDE. Experimental tests, which cover constant and varying step function scenarios, are conducted on the designed hardware. Results show the PID anti-windup controller's superiority, exhibiting significantly lower overshoot values of 5.42% and 3.35% compared to 13.26% and 23.76%, respectively. Despite its higher control action, the traditional PID controller displays a more pronounced overshoot. This research is a significant step toward advancing control systems for electric vehicles and optimizing BLDC motor performance in practical applications.

Keywords: BLDC; MATLAB; PID anti-windup controller; Traditional PID controller.

DOI: The DOI 10.25124/jmecs.v10i2.7209

1. Introduction

As the scarcity of fossil fuels continues to grow, researchers are increasingly exploring electric-powered vehicles, particularly electric cars, as a more sustainable alternative [1].Considering the limited availability of fossil fuels, electric cars offer a practical alternative. Moreover, they have the potential to substitute conventional vehicles that heavily depend on petroleum, contributing to the conservation of natural resources. The primary advantage of electric vehicles is their environmental friendliness, as they emit no air pollutants and do not require oil-based fuels.[2].

The electric motor is the primary component that powers an electric car [3]. The selection of a motor is based on the particular needs of the vehicle. Direct current (DC) motors are commonly used in electric cars because they can be easily operated to rotate in both clockwise and counterclockwise directions by simply reversing their polarity. Furthermore, DC motors have high rotational speeds (RPM) and their speed can be easily controlled.

In conventional brushed DC motors, the field winding is located in the stator while the armature winding is in the rotor. [4]. However, this configuration has its drawbacks, such as elevated costs and maintenance requirements stemming from the presence of brushes. Issues like the accumulation of debris, dust, and wear on the commutator surface are associated with these brushes. Moreover, brushed motors encounter constraints in environments that could pose dangers due to the potential risk of arcing. To address these concerns, the mechanical commutator and brushes are replaced with electronic semiconductor switches, giving rise to the Brushless DC (BLDC) motor. This motor design incorporates a permanent magnet rotor and a wound field stator linked to a power electronic switching setup. BLDC motor systems offer numerous benefits, including improved efficiency, reduced maintenance, prolonged lifespan, minimized noise, simplified control, lightweight construction, and a compact design [5].

Recently, numerous advanced control methods have emerged aiming to optimize the design of control systems for Brushless Direct Current (BLDC) motors [6]. Yet, these approaches are often characterized by their inherent complexity, demanding substantial computational resources In contrast, Proportional-Integral-Derivative (PID) control offers a straightforward yet highly effective solution to a wide range of control challenges [7].

With the rapid advancement of science and technology, there is an increasing demand for precision, quick responsiveness, and stability in control systems. To meet these requirements, the conventional PID (Proportional Integral Derivative) control method has become widely popular for its straightforward design and robust reliability. The fundamental principle of PID control entails generating control actions by incorporating proportional, integral, and derivative elements, forming a linear combination that effectively manages the desired target or system [8]. To enhance the speed control of a brushless DC motor, one can utilize the Proportional Integral Derivative (PID) controller. These enhancements are guided by specific time-domain criteria, which include settling time, undershoot, overshoot, recovery time, and steady-state error. Assessment of the motor's performance entails metrics such as root mean squared error, integral of absolute error, multiplied time integral of absolute error, and integral of squared error [9].

When a controller is constrained by limits that restrict its control signal within specific maximum and minimum boundaries, it may undergo saturation if a set point or disturbance pushes the control signal beyond these limits. Persistent saturation leads the controller to operate in an open-loop mode, causing the integral component to accumulate errors continuously. This error accumulation is commonly referred to as windup which introduces nonlinearity into the controller and has the potential to destabilize the closed-loop system. To mitigate the adverse effects of windup, various strategies, and techniques, collectively known as anti-windup techniques, are employed in Ref. [10].

When employing a conventional PID controller with pre-defined constraints on the control output, saturation may arise if the system encounters a shift in setpoint or disturbance, leading the control output signal to exceed the designated upper and lower bounds With the saturation phenomenon in place, integral anti-windup is necessary, which is the Back Calculation method. In this method, when the controller reaches saturation, the integral value is recalculated, resulting in a control value that does not exceed the saturation limit [11].

Previous research utilizing PID methods has shown that the system exhibited significant steady-state error and overshoot, indicating the need for further improvement in control strategies to achieve better performance [12]. The primary objective is to minimize steady-state error and reduce overshooting in the operational performance of the motor. This research paper aims to conduct simulations and analyses to enhance the performance of an anti-windup PID controller for the speed control of a BLDC (Brushless DC) motor in electric vehicles.

2. Material and Method

2.1 Drive System Models

Generally, there are three primary approaches to determine the system model of a Brushless Direct Current (BLDC) motor: white-box, black-box, and gray-box. The white-box, or analytical approach, involves defining a system model using mathematical equations to describe the physical properties of the plant. [13]. In contrast, the black-box, or experimental approach, relies on input-output data of the system when there is no accessible physical insight [12],[14]. On the other hand, the gray-box model combines elements of both black-box and white-box modeling [15]. This research opts to employ black-box modeling.

Fig. 1 illustrates the block diagram of the BLDC-drive system in this research. The system consists of a YALU BM1109's BLDC motor, a speed sensor, a motor driver controlled by a digital potentiometer and a relay, an Arduino Microcontroller, and a personal computer (PC). There is an essential specification of the BLDC motor branded Yalu BM1109 in Table 1. The 2560 series of Arduino Mega communicates with the PC through a universal serial bus (USB) and delivers control mechanisms to the BLDC motor by adjusting the appropriate resistive value of the digital potentiometer. The MCP41050's digital potentiometer varies and controls the BLDC's speed, while the relay performs three-speed modes: slow, normal, and fast and the speed sensor will read the motor rotational speed at that time, then the reading of this speed sensor will be used as a comparison against the desire rotational speed value, the speed sensor used are hall effect sensor type ky-003.

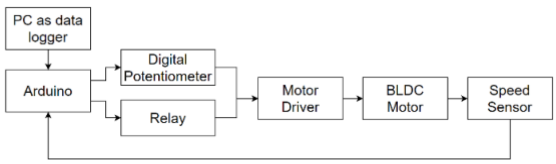


Fig. 1. Block diagram of BLDC-drive system.

Table 1. Yalu Bm1109 BLDC Specifications

Parameters	**Value**
Rated Voltage (V)	48
Rated Current (A)	30
Power (Watt)	2000
Max Speed (rpm)	5000
Rated Speed (rpm)	4000
Rated Torque (Nm)	3.5

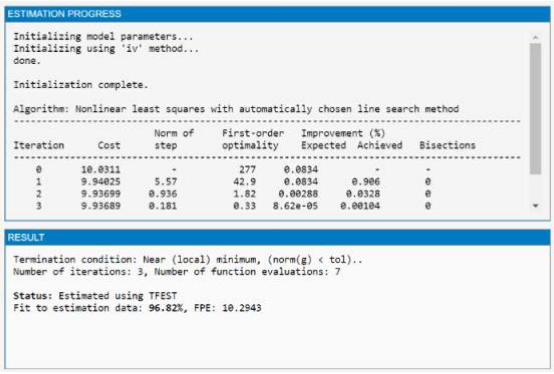


Fig. 2. Fit estimation Eq. (1).

Black-box modeling utilizes input and output data, employing MATLAB system identification techniques to derive a system model. In this context, the input signal ranging from 0 to 255 originates from the digital potentiometer, while the output signal represents the speed measured by a Hall effect sensor, with a sampling interval of 0.5 seconds. Utilizing this data, MATLAB System Identification was employed to derive system models. As depicted in Fig. 2, the fitting between input and output facilitated the extraction of the transfer function presented in Equation (1). The fit to estimation yielded a value of 92.99%, surpassing the threshold of 90%, thus validating the usability of the derived transfer function [14]. Eq. (1) is the transfer function model of the system in second-order form already tested in MATLAB/Simulink.

$$G(s) = \frac{3.594}{s^2+1.872\,s+0.9206} \tag{1}$$

2.2 PID Anti-Windup Control Design

PID Anti-Windup combines a PID control and an integral of Anti-Windup [10]. We use the parallel form of the PID controller in which the gain proportional (k_p), gain integrator (k_i), and gain derivative (k_d) components are separated. It is the more practical PID controller. This form is often written as Eq. (2).

$$u(t) = k_p e_y(t) + k_i \int_0^t e_y(t) + k_d \frac{d}{d_t} e_y(t). \tag{2}$$

Table 2. Value of k_p, k_i, and k_d from PID tuning MATLAB

Controller Parameters	**Tuned**
k_p	0.016857
k_i	0.0082878
k_d	0.0085712

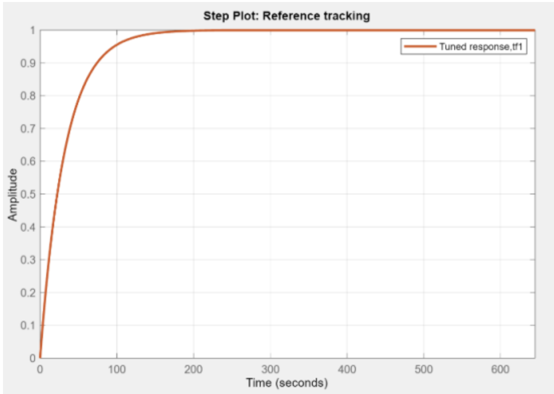


Fig. 3. Response system with PID controller.

From Eq. (2), $u(t)$ performs as control signals [10]. The value of k_p, k_i, and k_d was obtained by using PID Tuner in MATLAB and it can be seen in Table 2. $e_y(t)$ can be expressed as $e_y(t) = \bar{y}(t) - y(t)$ meaning the difference between the output $y(t)$ and a certain reference signal $\bar{y}(t)$. Fig. 3 shows the response system with the PID controller in k_p, k_i, and k_d values, in which a time response and robustness reach 64.5 seconds and 0.9, respectively.

When using a PID controller with predefined constraints on the control output, saturation may occur if the system experiences a setpoint change or disturbance that causes the controller's output signal to exceed these predefined upper and lower limits. If the actuator constraints are defined by two scalar values, with u_{min} less than u_{max}, the saturation function is established as follows[10] :

$$sat(u) = \begin{cases} u_{min}, & if\ u < u_{min} \\ u, & if\ u_{min} \le u \le u_{max} \\ u_{max}, & if\ u > u_{max} \end{cases} \tag{3}$$

So, we use Integral Anti-Windup to overcome the saturation as the Integral Anti-Windup control is designed with three steps, which are to stop the accumulation of integral when saturated, limit the minimum and maximum value of integral, and then reduce the input of integral when signal control the saturation [10]. This paper focuses on an early and fundamental anti-windup technique known as back-calculation [16].

In the back-calculation method, when the controller output reaches saturation, the integral is recalculated to provide a new value that corresponds to the output at the saturation limit. Rather than resetting the integrator immediately, it is beneficial to perform a dynamic reset with a time constant denoted as T_t.

The system performance is noticeably influenced by the tracking time constant in the context of back calculation. T_t dictates the speed at which the integral term undergoes a reset. Opting for smaller tracking time constants results in a quicker reset of the integrator, initially appearing advantageous. Therefore, a general guideline suggests that the tracking time constant should fall between the integrator time constant (T_i) and the derivative time constant (T_d). A commonly recommended rule of thumb is to set $T_t = \sqrt{T_i * T_d}$ (4), where T_i and T_d represent the integrator and derivative time constants of the PID controller [17]. To calculate the value of the integrator time constant (T_i) and the derivative time constant (T_d), we can use the control parameters in Table 2. Substituting Eq. (4) and Eq. (5) to Eq. (3) with the control parameters value from Table. 2 then we get the value of T_t is 1.01695.

$$T_i = \frac{k_p}{k_i} \quad (5)$$

$$T_d = \frac{k_d}{k_p} \quad (6)$$

The PID Anti-Windup control design process is delineated in the flowchart illustrated in Fig. 4, comprising a sequential set of five steps. To commence, the initial step involves ascertaining both the input and output parameters. The input is represented by a digitized resistive value, ranging from 0 to 255, generated by the digital potentiometer. While the output is determined by the speed value obtained from the Hall effect sensor. Following this, the transfer function model, as articulated in Eq. (1), is derived through the utilization of MATLAB system identification techniques. This model encapsulates the dynamic relationship between the input and output variables. Subsequently, the third step involves the design and simulation of the PID Anti-Windup control in MATLAB, employing the aforementioned transfer function equation. The simulation process enables an evaluation of the system's response to the designed control Upon successful simulation, the fourth step entails scrutinizing the system response to ensure it meets predefined criteria. If the established criteria are satisfied, signifying the efficacy of the PID Anti-Windup control design in the simulated environment, the final step involves transitioning to the hardware implementation. The control system, now validated through simulation, is transferred to the physical hardware for real-world application, ensuring a seamless integration of the designed control mechanism into the practical setting. This approach

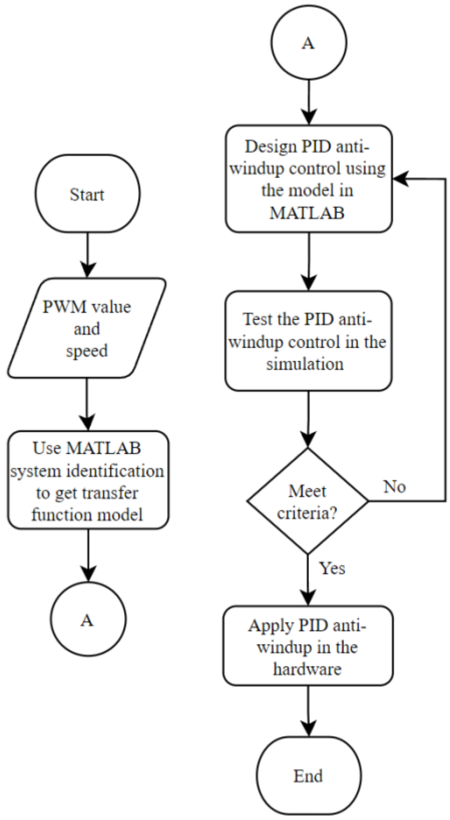


Fig. 4. Flowchart PID anti-windup control design process.

ensures a reliable PID Anti-Windup control design through simulation and hardware validation.

When saturation is absent, the signal remains at zero and does not affect normal operation. However, when the actuator saturates, the error signal deviates from zero, disrupting the regular feedback loop around the process, as the process input remains constant. Nonetheless, a feedback loop persists around the integrator, causing the integrator output to move towards a value that cancels out the integrator input. As we know the control signal $u(t)$ in PID controller also change because a saturation effect so the formula of control signal is shown in Eq. (7). Where the actuator undergoes saturation, the new control signal is denoted as $u^*(t)$.

$$u^*(t) = k_p e_y(t) + [(k_i du(t) k_a) \int_0^t e_y(t)\, d(t)] + k_d \frac{d}{d_t} e_y(t) \quad (7)$$

Where u^* is the new control signal, k_p is the proportional gain coefficient, k_i is the integral gain coefficient, k_d is the derivative gain coefficient, k_a is

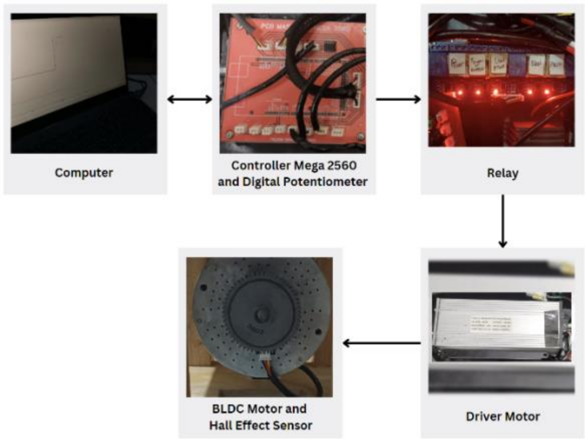


Fig. 5. Experimental setup

the anti-windup gain coefficient, and e_y is the error signal.

To mitigate the risk of overshoot during exercise, the back-calculation method can be utilized to incorporate an external anti-windup mechanism. This approach involves processing the input $du(t)$, which represents the variance between the saturated control signal $(sat(u))$ and the computed unsaturated control signal $(u(t))$. Subsequently, this difference is multiplied by the anti-windup gain coefficient, which is determined by the time constant $(\frac{1}{T_t})$ and the integral error. Following that, adds the amplified signal from the integral gain.

3. Result and Discussion

In the experiment illustrated in Fig. 5, The system utilizes an Arduino Mega 2560 as the microcontroller, with the Arduino code depicted in Fig. 6. The speed of the BLDC motor is finely adjusted using a digital potentiometer, which acts as the control input. Concurrently, a motor driver, integrated with a relay, manages the operational aspects of the motor. Furthermore, the system integrates a Hall effect sensor to precisely measure the motor's rotational speed in RPM, playing a crucial role in ensuring accurate feedback.

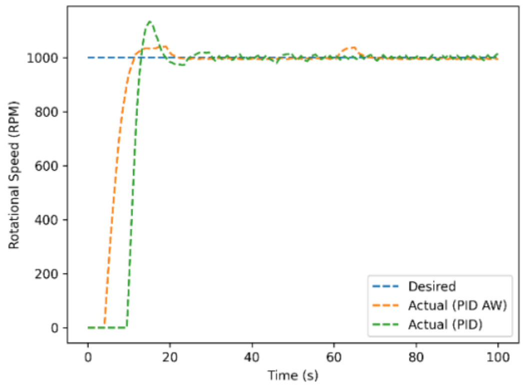
The aim of the experiment was to compare the effectiveness of a PID anti-windup controller to that of a traditional PID controller in a Brushless DC (BLDC) drive system. The hardware test was conducted without a load, incorporating two conditional tests: a constant step function test and a varying step function test.

3.1 Constant Step Function Test

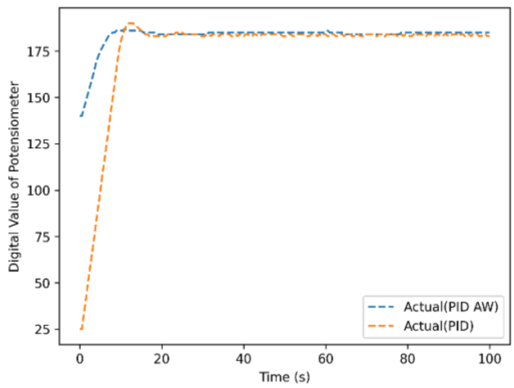
The response of the control system over a time duration of 100 seconds is illustrated in Fig. 7 (a). The graph illustrates that the BLDC motor precisely follows the given speed command, and all controllers have achieved the desired speed. The given time frame.

```
double up_limit = 255; //upper limit
double un_limit = 0;//under limit
double newpot;
double error = 0;
double P, I, D, prev;
P = error;
D = (error - prev) ;

double kp = 0.016857;  //k_p value
double ki = 0.0082878; //k_i value
double kd = 0.0085712; //k_d value
double ka = 0.98333;//anti windup gain

if (newpot > up_limit) { //anti-windup
  newpot = (kp * error) + (ki * I) + (ka * (up_limit -
newpot) * I) + (kd * D);
 }
else if (newpot < un_limit) { //anti-windup
  newpot = (kp * error) + (ki * I) + (ka * (un_limit -
newpot) * I) + (kd * D);
 }
else {
  newpot = (kp * error) + (ki * I) + (kd * D);
 }
I += error;
prev = error;
```

Fig. 6. Arduino code for PID anti-windup



(a)



(b)

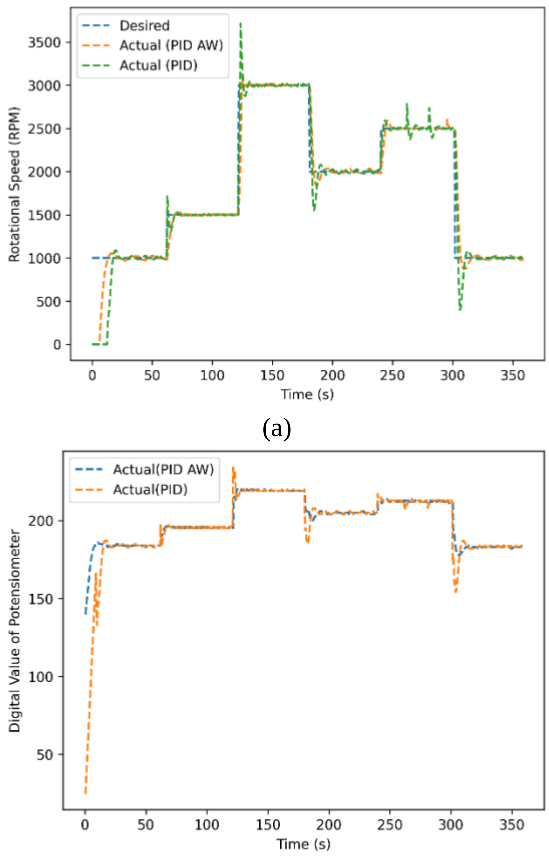
Fig. 7. Constant step test, (a) control response (b) control action

Table 3. Control Performance Under Constant Step Function

Performance	Traditional PID	PID anti-windup
Rise time (s)	13.00	11.50
Peak time (s)	15.00	16.50
Settling time (s)	19.00	17.00
Overshoot (%)	13.26	5.42

However, distinct performance variations were observed among the controllers during the tracking process, with the PID anti-windup controller notably demonstrating superior tracking speed compared to the traditional PID controller. Additionally, the PID anti-windup controller exhibits slight oscillations during its steady-state condition, accompanied by minimal overshoot. As a result, it causes a rapid increase in rise time, as validated by the data in Table 3, and demonstrates strong control action shown in Fig. 7 (b).

The BLDC motor is instructed to follow a rotational speed which is 1000, 1500, 3000, 2000, 2500, and 1000 revolutions per minute using a varying step function, representing changing load conditions.



(a)

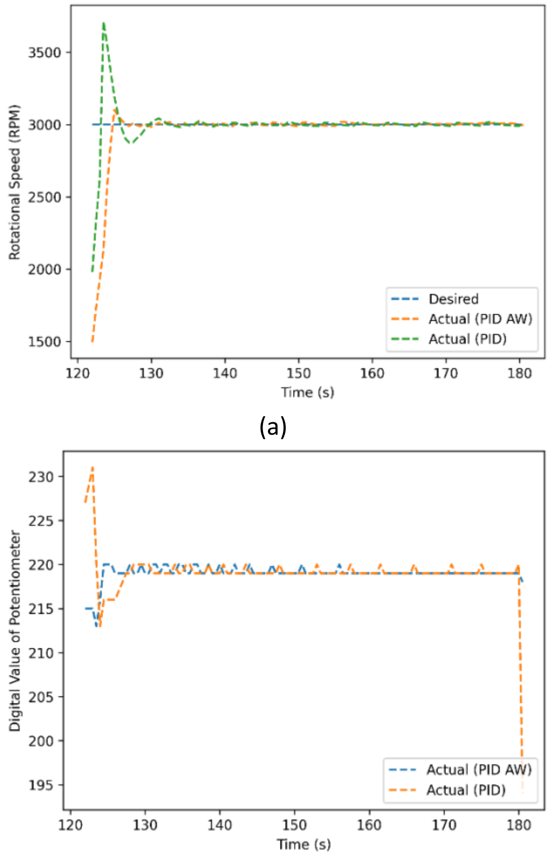
(b)

Fig. 8. Varying step test, (a) control response (b) control action

3.2 Varying Step Function Test

These outcomes are depicted in Fig. 8. Once again, the BLDC motor maintains the specified speed, with both controllers accurately tracking it throughout the experiment. To provide clarity, the control response and action are amplified within the 122-180-second timeframe, as shown in Fig. 9. These observations underscore the consistency of results with the constant step function in this scenario.

The PID anti-windup controller has a rise time of 1.20 seconds, compared to the traditional PID controller's 3.75 seconds. However, the former generates an overshoot of only 23.76%, which minimizes overshoots and leads to a quicker rise time and improved control action. Therefore, the PID anti-windup controller is superior to the traditional PID controller in suppressing overshoots.. This observation is corroborated by Fig. 9 and Table 4.



(a)

(b)

Fig. 9. Time window 122-180, (a) control response (b) control action

Table 4. Performance Control Performance Under Varying Step Function

Performance	Traditional PID	PID anti-windup
Rise time (s)	1.20	3.75
Peak time (s)	2.00	4.00
Settling time (s)	4.50	4.50
Overshoot (%)	23.76	3.35

4. Conclusions

The study compared the control performance of PID anti-windup with traditional PID control, drawing several key conclusions and recommendations. Firstly, it was found that the PID anti-windup controller outperformed the traditional PID controller in terms of overshoot. Tests involving both the constant step function and varying step function revealed that the PID anti-windup controller demonstrated a significantly lower overshoot, with values of 5.42% and 3.35% respectively, compared to 13.26% and 23.76% for the traditional PID controller. Additionally, it was observed that the traditional PID controller exhibited a higher control action, contributing to its greater overshoot compared to the PID anti-windup. As a recommendation, further exploration into BLDC motor speed control is suggested. This could involve investigating alternative approaches to enhance overall performance, aiming to build upon the observed advantages of the PID anti-windup controller and potentially uncovering new methods to optimize control systems in similar applications.

Acknowledgment

This research is a collaboration between BRIN and Telkom University. Huge credit is presented to Lembaga Pengelola Dana Pendidikan (LPDP), which has supported a fund for this research.

References

[1] B. Putri, S. Sutedjo, O. Qudsi, and L. Septya Mahendra, “Alat Penstabil Kecepatan Motor BLDC Menggunakan Kontrol PID,” *Emitor: Jurnal Teknik Elektro*, vol. 22, pp. 134–140, Jan. 2022, doi: 10.23917/emitor.v22i2.19384.

[2] M. Aziz, Y. Marcellino, I. A. Rizki, S. A. Ikhwanuddin, and J. W. Simatupang, “Studi Analisis Perkembangan Teknologi Dan Dukungan Pemerintah Indonesia Terkait Mobil Listrik,” *TESLA: Jurnal Teknik Elektro*, vol. 22, no. 1, pp. 45–55, Mar. 2020, doi: 10.24912/tesla.v22i1.7898.

[3] B. P. Nursandy, Y. S. Pramesti, and A. Sugianto, “Rancang Bangun Engine Sebagai Penggerak Prototipe Mobil Listrik,” *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, vol. 7, no. 2, pp. 881–887, Jul. 2023, doi: 10.29407/inotek.v7i2.3512.

[4] R. Badarudin, D. Hariyanto, M. Ali, and S. Meunmany, “Electric Machines Virtual Laboratory: Testing of DC Motor,” *J Phys Conf Ser*, vol. 2111, no. 1, p. 12041, Nov. 2021, doi: 10.1088/1742-6596/2111/1/012041.

[5] M. Deepak *et al.*, “A Review of BLDC Motor: State of Art, Advanced Control Techniques, and Applications,” *IEEE Access*, vol. 10, p. 1, Jan. 2022, doi: 10.1109/ACCESS.2022.3175011.

[6] H. Hu, T. Wang, S. Zhao, and C. Wang, “Speed Control Of Brushless Direct Current Motor Using A Genetic Algorithm–Optimized Fuzzy Proportional Integral Differential Controller,” *Advances in Mechanical Engineering*, vol. 11, p. 168781401989019, Jan. 2019, doi: 10.1177/1687814019890199.

[7] A. Ma’arif, R. ISTIARNO, and S. SUNARDI, “Kontrol Proporsional Integral Derivatif (PID) pada Kecepatan Sudut Motor DC dengan Pemodelan Identifikasi Sistem dan Tuning,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 9, p. 374, Apr. 2021, doi: 10.26760/elkomika.v9i2.374.

[8] Z. Yin *et al.*, “Torque-Induced-Overshoot Reduction Inspired Compensator for PMSM using Motor-Physics Embedded Gaussian Process Regression,” 2023. doi: 10.36227/techrxiv.170327187.79736266/v1.

[9] M. I. Azeez, A. M. M. Abdelhaleem, S. Elnaggar, K. A. F. Moustafa, and K. R. Atia, “Optimization Of PID Trajectory Tracking Controller For A 3-DOF Robotic Manipulator Using Enhanced Artificial Bee Colony Algorithm,” *Sci Rep*, vol. 13, no. 1, p. 11164, 2023, doi: 10.1038/s41598-023-37895-3.

[10] N. P. Lawrence, G. E. Stewart, P. D. Loewen, M. G. Forbes, J. U. Backstrom, and R. B. Gopaluni, “Optimal PID and Antiwindup Control Design as a Reinforcement Learning Problem,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 236–241, 2020, doi: https://doi.org/10.1016/j.ifacol.2020.12.129.

[11] A. R. Kumar and P. J. Ramadge, “DiffLoop: Tuning PID Controllers by Differentiating Through the Feedback Loop,” in *2021 55th*

Annual Conference on Information Sciences and Systems (CISS), 2021, pp. 1–6. doi: 10.1109/CISS50987.2021.9400299.

[12] S. Triwijaya, Y. Prasetyo, and T. Wati, “Kontrol Kecepatan Motor BLDC dengan PID - Firefly,” *Jurnal IPTEK*, vol. 25, no. 1, pp. 51–58, 2021, doi: 10.31284/j.iptek.2021.v25i1.963.

[13] M. Haziq Bin and M. Nor, “Black-Box Modelling For Brushless Dc Motor And Control Strategy For Quadcopter Applications,” *School of Aerospace Engineering*, 2021.

[14] S. Praptodiyono, H. Maghfiroh, J. S. Saputro, and A. Ramelan, “Integral State Feedback Control Using Linear Quadratic Gaussian in DC-drive System,” *Jurnal Elektronika dan Telekomunikasi*, vol. 21, no. 2, p. 79, Dec. 2021, doi: 10.14203/jet.v21.79-84.

[15] F. Mangkusasmito, D. Y. Tadeus, and A. Subari, “Implementasi Identifikasi Sistem Metode Black Box Pada Motor Dc Menggunakan Correlation Analysis Dan Model Arx,” *Gema Teknologi*, vol. 20, no. 4, pp. 134–139, Apr. 2020, doi: 10.14710/gt.v20i4.29381.

[16] M. O. Okelola, D. Aborisade, and P. Adewuyi, “Performance and Configuration Analysis of Tracking Time Anti-Windup PID Controllers,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, pp. 20–29, Dec. 2020, doi: 10.26555/jiteki.v6i2.18867.

[17] H. Markaroglu, M. Guzelkaya, I. Eksin, and E. Yesil, “Tracking Time Adjustment In Back Calculation Anti-Windup Scheme,” in *20th European Conference on Modelling and Simulation: Modelling Methodologies and Simulation Key Technologies in Academia and Industry, ECMS 2006*, European Council for Modelling and Simulation, 2006, pp. 613–618. doi: 10.7148/2006-0613.

Author information

Aditya Adni Argaloka, was born in Cirebon, Indonesia in 2002, is presently pursuing a Bachelor's Degree in Electrical Engineering at the School of Electrical Engineering, Telkom University. He is actively engaged as a practicum assistant in the Electronic Laboratory and exhibits a strong passion for research in the field of control systems.

Harwin Aptadarya was born in Kediri, Indonesia in 2002. He was a practicum assistant and head of the practicum division at uPCI Laboratory from 2022 to 2023. He is interested in embedded systems and autonomous electric vehicles which are currently developing rapidly. He received his Bachelor’s Degree in Electrical Engineering at Telkom University, Bandung, Indonesia in 2024.

Fiendo Mahendra Arentaka was born on October 8, 2002, in Pati. He is an Electrical Engineering student at Telkom University and currently resides in Cilacap. During his tenure as the former R&D Department Chairman at KMTE, his leadership skills flourished. Presently, his focus is on groundbreaking research with BRIN, contributing to the development of autonomous electric vehicles. His journey reflects a commitment to innovation and a sustainable technological future.

Fiky Y. Suratman (S'09 M'14) was born in Jakarta, Indonesia, in 1976. He received Bachelor Degree from Engineering Physics and Master Degree from School of Electrical and Informatics (STEI), Institut Teknologi Bandung, Bandung, Indonesia in 1998 and 2006, respectively. He was awarded scholarship from German Academic Exchange Service (DAAD) to continue his study at Technische Universitaet Darmstadt (TU-Darmstadt), Germany, until he recieved his Dr.-Ing. Degree (PhD) in 2014. Early placement in industry (Astra Microtronics Technology) from 1998 to 2001, was then followed by lectureship in Universitas Komputer Indonesia (UNIKOM). In 2007, he joined the Faculty of Electrical Engineering, Telkom University as a faculty member. He is a member of IEEE and he currently acts as the chairman of Signal Processing Society (SPS) Indonesia Chapter. His research interest lies on Statistical Signal Processing, Radar Signal Processing, and Machine Learning.

Arief Suryadi Satyawan was born in Bandung, Indonesia, in 1973. He has developed his research career since 1996 at the Research Center for Electronics and Telecommunication, Indonesian Institute of Science, which later merged with the National Research and Innovation Agency in 2021. He received his Bachelor's degree in electrical engineering from Universitas Jenderal Achmad Yani Bandung, Indonesia, in 2000. In 2007, he obtained his Master's in electrical engineering from Institut Teknologi Bandung, Indonesia. He got his Doctor of Engineering in computer and communication engineering from Waseda University, Japan, in 2019. His current research is the development of autonomous vehicles.

Additional Information