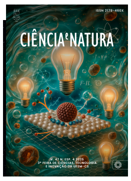
UFSM

CIÊNCIAeNATURA



ISSN 2179-460X
OPEN ACCESS

Ci. e Nat., Santa Maria, v. 47, n. spe. 4, e92172, 2025 • https://doi.org/10.5902/2179460X92172
Submitted: 24/05/24 • Approved: 01/07/24 • Published: 10/09/25

III Science, Technology, and Innovation Fair of UFSM-CS

Control System Design for a Monocopter Using Arduino Nano

Projeto de Controle de um Monocóptero Utilizando Arduino Nano

Eagro Henrique Brenner Muller [I], **Pedro Antônio Castro Regio dos Santos** [I], **André Francisco Caldeira** [I], **Charles Rech** [I] **Simone Ferigolo Venturini** [II], **Carmen Brum Rosa** [II]

[I]Universidade Federal de Santa Maria - campus Cachoeira do sul, RS, Brazil
[II]Universidade Federal de Santa Maria, RS, Brazil

ABSTRACT

This paper presents the development and application of an electromechanical control methodology utilizing a reduced-order model, implemented in a real physical system. The chosen plant for validation was a monocopter, controlled via an Arduino Nano microcontroller running a digital PID controller. The methodology employs a simplified mathematical model of the plant, which proved sufficient to design and tune an effective controller, even in the face of nonlinear dynamics and the inherent instability of the system. This approach facilitated prior simulations and controller adjustments, streamlining the implementation process. The development included an analysis of the sensors and actuators, and the use of digital filters such as the moving average filter to mitigate noise in measurement signals. The system demonstrated stability and disturbance rejection capabilities, validating the effectiveness of the proposed approach. The project reinforces the feasibility of applying classical feedback control techniques in mechatronic systems, offering a cost-effective and academically accessible solution with potential applications in autonomous control and robotics.

Keywords: PID, Control system, Reduced-order model, Arduino Nano, Controlled plant

RESUMO

Este artigo apresenta o desenvolvimento e a aplicação de uma metodologia de controle eletromecânico utilizando um modelo de ordem reduzida, implementada em um sistema físico real. A planta escolhida para validação foi um monocóptero, controlado por meio de um microcontrolador Arduino Nano executando um controlador PID digital. A metodologia emprega um modelo matemático simplificado da planta, que se mostrou suficiente para projetar e sintonizar um controlador eficaz, mesmo diante de dinâmicas não lineares e da instabilidade inerente ao sistema. Essa abordagem facilitou simulações prévias e ajustes do controlador, otimizando o processo de implementação. O desenvolvimento incluiu uma análise dos sensores e atuadores, e o uso de filtros digitais, como o filtro de média móvel, para mitigar ruídos nos sinais de medição. O sistema demonstrou estabilidade e capacidade de rejeição a distúrbios, validando a eficácia da abordagem proposta. O projeto reforça a viabilidade da aplicação de técnicas clássicas de controle por realimentação em sistemas mecatrônicos, oferecendo uma solução de baixo custo e academicamente acessível com potenciais aplicações em controle autônomo e robótica.

Palavras-chave: PID, Sistema de controle, Modelo reduzido, Arduino Nano, Planta de controle

1 INTRODUCTION

Digital control is an essential area in the field of engineering, particularly in electromechanical systems where automatic control processes are required to ensure precision and stability. Unlike analog control systems, where signals vary continuously over time, in digital control, information is processed discretely, i.e., at well-defined time intervals. With the introduction of microcontrollers, such as Arduino, which possess sufficient processing power to perform real-time calculations, digital control has become an efficient and accessible alternative for diverse applications, such as the automation of aerial vehicles and industrial systems.

The advancement of digital control technology and the development of low-cost platforms, such as Arduino, have enabled the creation of precise and accessible control systems for a variety of applications, institutions, and groups, thus leading to a democratization of this technology. Alongside the growing use of these physical electromechanical control resources, Proportional-Integral-Derivative (PID) control algorithms are widely utilized in control systems due to their simplicity, versatility, and effectiveness in adjusting process variables to reach a desired value or *setpoint*. The PID controller combines three control actions: the Proportional (P) term, which reacts to the current error between the measured value and the desired value; the Integral (I) term, which considers the history of accumulated error over time, helping to eliminate steady-state errors; and the Derivative (D) term, which anticipates changes by measuring the rate of change of the error, providing greater stability to the system. This combination allows the PID controller to respond quickly and accurately to changes, adapting to the dynamic conditions of the system. Implemented on microcontrollers like Arduino, the PID controller is ideal for applications such as controlling the speed, position, and stability of various control plants, offering an effective solution to maintain system performance in the presence of disturbances.

A ”control plant” is a mathematical representation of a physical system to be controlled, serving as the basis for designing and implementing control algorithms that ensure the desired performance. In control systems, the plant defines the relationship between the inputs (control commands) and the outputs (observed responses), providing a model that describes how the system reacts to external stimuli. This model, usually expressed in terms of differential equations or transfer functions, is characterized by its order, which corresponds to the number of derivatives or independent states involved in the plant’s dynamics. Plant orders can vary, ranging from first-order systems, which respond directly and relatively simply, to higher-order systems, which exhibit more complex responses and often require more sophisticated controllers. Understanding the plant’s order and dynamic behavior is essential for controller design, as it allows predicting how the system will respond to variations and adjusting control parameters according to its complexity and performance requirements.

To obtain an accurate representation of the control plant, it is common practice to employ system identification methodologies that analyze input data and the system’s dynamic response. This approach allows for the mathematical characterization of the system’s behavior without the need for detailed theoretical modeling, often involving approximation using a reduced-order model with delay. Tools such as MATLAB and Python are widely utilized in this context, offering capabilities for fitting models directly from experimental data and applying data processing techniques like filtering and parameter tuning.

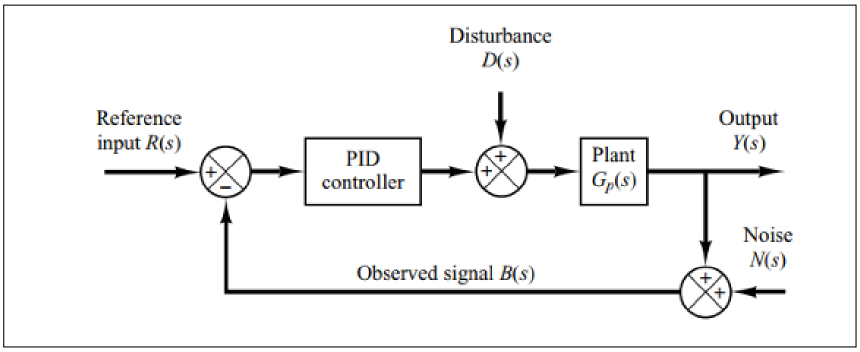
The development of precise models that capture the system dynamics is essential for the design of effective controllers. Within this framework, the general objective of this work is to apply these techniques to design and implement a discrete PID control system using the Arduino platform for a predefined monocopter plant. Specifically, this involves obtaining and validating a first-order plus delay model for this plant, systematically organizing data from manuals, sensors, actuators, and components, and performing instrumentation tests, including the design of a moving average filter for signal processing, thereby ensuring the necessary foundation for controller development.

2 LITERATURE REVIEW

The ability to minimize and correct errors in processes is crucial for engineering and science. Closed-loop control systems are indispensable tools for achieving this objective, being widely used both in industrial environments and in research. The difference between the reference signal and the output signal, known as the error, acts as an input to the controller in a closed-loop system. The controller, in turn, generates a control signal that aims to minimize this error, ensuring the desired stability and performance of the system (Ogata, 2010).

Figure 1 presents a typical block diagram of a PID control system. In this configuration, the PID controller uses the difference between the reference value and the plant output (the error) as input to generate a control signal. This signal is applied to the plant with the objective of minimizing the error and ensuring that the output follows the reference (Ogata, 2010).

Figure 1 – Control system represented by block diagrams



Source: Ogata (2010)

The output signal of the PID controller can be represented as a function of the error and the proportional, integral, and differential gains. The derivative control action has the advantage of anticipating the error; however, it has the disadvantage of amplifying noise (Ogata, 2010). The proportional-integral-derivative (PID) control action can be represented by the equation,

$$u(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^1 e(t)dt + K_p T_d \frac{de(t)}{dt}, \tag{1}$$

where K_p represents the proportional gain, $e(t)$ is the error of the measure, T_d represents the derivative time, and T_i represents the integral time, with $u(t)$ being the controller output (Ogata, 2010).

2.1 Discrete PID

The discrete approximation of the PID controller is achieved by approximating the integral in Equation 1 using a trapezoidal summation and approximating the derivative using the finite difference method (Caldeira & Rech, 2024). The discrete PID control action can be represented by the equation,

$$u(k) = k_c \left[e(k) + \frac{T_s}{2t_i} \sum_{h=0}^{k} [e(h) + e(h-1)] + \frac{t_d}{T_s} [e(k) - e(k-1)] \right], \tag{2}$$

and by delaying the controller output by one sample $(k-1)$, the following equation is obtained:

$$u(k-1) = k_c \left[e(k-1) + \frac{T_s}{2t_i} \sum_{h=0}^{k} [e(h-1) + e(h-2)] + \frac{t_d}{T_s} [e(k-1) - e(k-2)] \right]. \tag{3}$$

Subtracting Equation 3 from Equation 2, substituting the summation, and rearranging the terms yields the equation,

$$u(k) - u(k-1) = k_c \left[\left(1 + \frac{T_s}{2t_i} + \frac{t_d}{T_s}\right) e(k) - \left(1 - \frac{T_s}{2t_i} + \frac{2t_d}{T_s}\right) e(k-1) + \frac{t_d}{T_s} e(k-2) \right], \tag{4}$$

and from Equation 4, we can write the PID controller equation as (Caldeira & Rech, 2024):

$$u(k) = u(k-1) + q_0 \cdot e(k) + q_1 \cdot e(k-1) + q_2 \cdot e(k-2), \tag{5}$$

where:

$$q_0 = k_c \left(1 + \frac{T_s}{2t_i} + \frac{t_d}{T_s}\right), \tag{6}$$

$$q_1 = -k_c \left(1 - \frac{T_s}{2t_i} + \frac{2t_d}{T_s}\right) \tag{7}$$

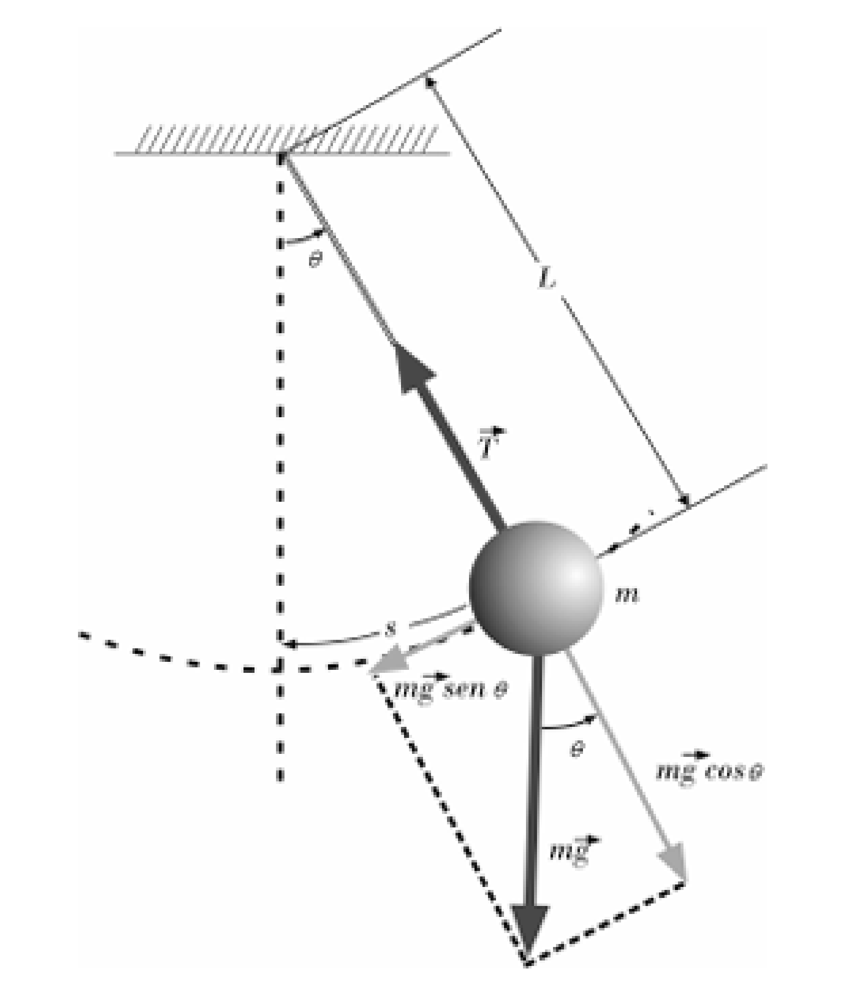
and

$$q_2 = k_c \frac{t_d}{T_s}. \tag{8}$$

2.2 Pendulum Behavior

The modeling of the simple pendulum, a fundamental physical system, has wide application in various fields, ranging from physics to control engineering. Detailed understanding of this system is essential for the development of more complex models and the analysis of oscillatory phenomena in dynamic systems. Figure 2 shows a diagram of a pendulum, where the mass, m, is displaced from its rest position; thus, the system will enter oscillatory motion (Arnold et al., 2011)

Figure 2 – Simple pendulum and acting forces



Source: Arnold et al. (2011)

Accounting for the forces acting on the pendulum in Figure 2, the following equation can be written (Arnold et al., 2011),

$$mL\frac{d^2\theta}{dt^2} + mg\sin\theta = 0, \tag{9}$$

where θ is the angular position (in radians), t is time (in seconds), g is the acceleration due to gravity (in m/s^2), and L is the length of the string (in meters). When considering the application of a tangential force $F(t)$ to the pendulum's trajectory, acting on the mass, the equation becomes:

$$mL\frac{d^2\theta}{dt^2} + mg\sin\theta = F(t)L. \tag{10}$$

Assuming small oscillation angles ($\sin\theta \approx \theta$) and that the arc length s can be approximated by the horizontal displacement x (i.e., $x \approx s = L\theta$), substituting these approximations into Equation 10 and rearranging yields:

$$\frac{d^2x}{dt^2} + \frac{g}{L}x = \frac{F(t)}{m}. \tag{11}$$

Equation 11 represents the pendulum's oscillation in the time domain. Applying the Laplace transform, the pendulum's oscillation in the frequency domain, including initial conditions $s(0)$ and $\frac{ds}{dt}(0)$, is represented by the equation,

$$S(s) = \frac{\frac{F(s)}{m} + ss(0) + \frac{ds}{dt}(0)}{s^2 + \frac{g}{L}}, \tag{12}$$

and the transfer function (relating input $F(s)$ to output $S(s)$, assuming zero initial conditions) is given by the equation,

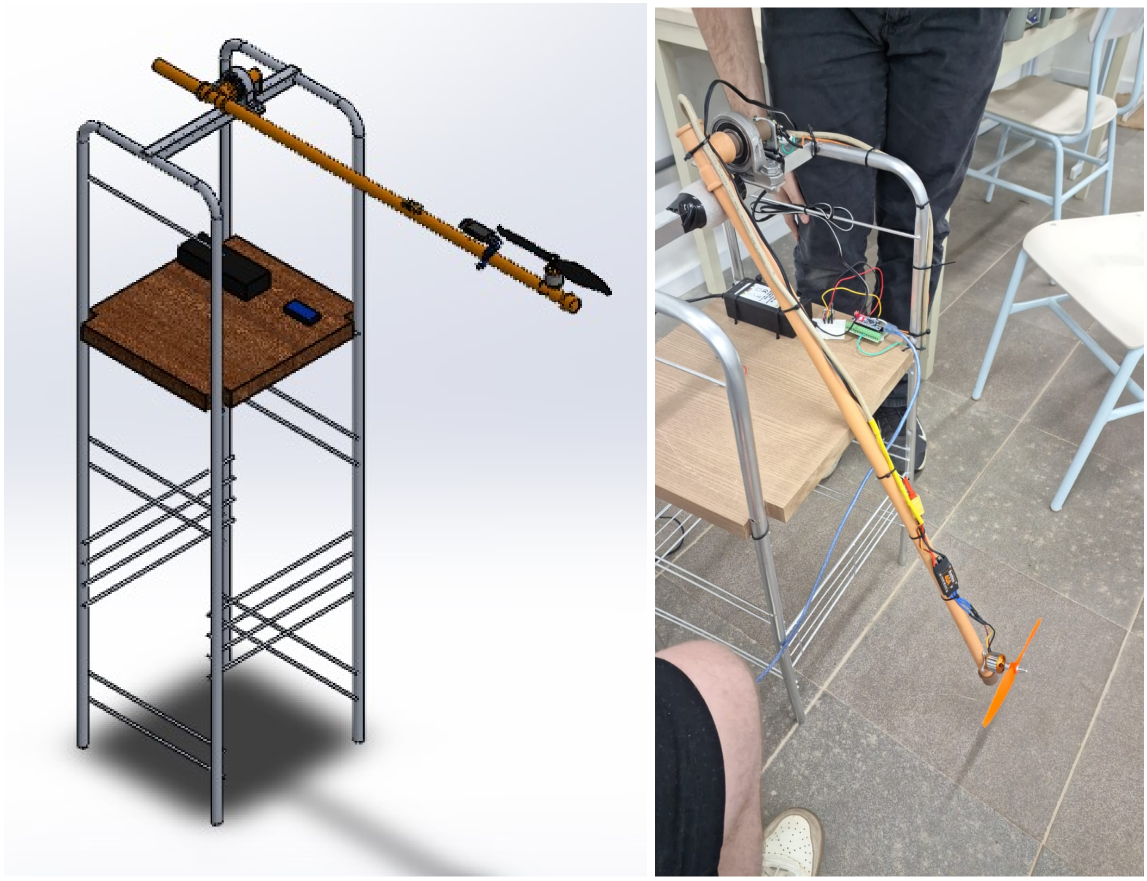
$$G(s) = \frac{1}{m\left(s^2 + \frac{g}{L}\right)}, \tag{13}$$

the denominator $s^2 + \frac{g}{L}$ indicates that the transfer function is second-order.

2.3 CHARACTERISTICS OF THE PLANT TO BE CONTROLLED

The plant to be controlled is a monocopter, as shown in Figure 3. A brushless motor is attached to the end of a lever arm; this arm is mechanically connected and free to rotate around an axis located at the opposite end. This axis is connected to a potentiometer whose purpose is to measure the angle of the lever arm. The measurement range used considers 0% as the position where the arm rests on the plant's wooden base and 100% when the arm is fully horizontal.

Figure 3 – Monocopter plant in: (a) CAD Drawing (b) Real Model



(a) (b)

Source: Authors, 2025

The potentiometer functions as a sensor, sending the position to the Arduino Nano microcontroller, which, based on the control loop logic, determines the signal to be sent to the Electronic Speed Controller (ESC). The ESC receives the *Pulse Width Modulation* (PWM) signal (typically a logic-level signal with pulses ranging from 1 to 2 milliseconds) from the Arduino and is powered by a 12V source. It interprets the pulse width to control the speed and sends appropriate power signals to the actuator (the brushless motor). All components comprising the plant are listed in Table 1 and will be detailed subsequently.

Table 1 – List of plant components

Number	Component	Weight	Reference
1	Power Supply 12V-5A - MSP23800ic12.0-48W	260 g	(Intelbras, 2022)
2	Brushless ESC 40A 2-4S LiPo BEC 5V/3A	38 g	(Modelismo, 2024)*
3	Potentiometer LR l20 W148-1 1M	N/A	(Technology, 2024)
4	Motor A2212/6T 2200kv	53g	(4-Max, 2024)
5	Arduino Nano	N/A	(AllDatasheet, 2024)
6	BT 1406$_{Base}$	N/A	(Livre, 2024)*
7	Propeller EP8040	[...]	(AeroModel, 2024)*
8	Cables	N/A	(Blucabos, 2024)*
9	Pillow Block Bearing$_{AL}$ P004	N/A	(Vendas, 2024)*
10	Bearing R004	N/A	
11	Base$_{AL}$	N/A	
12	Base$_{Wood}$	N/A	
13	Arm$_{PVC}$	105 g	(Merlin, 2024)*

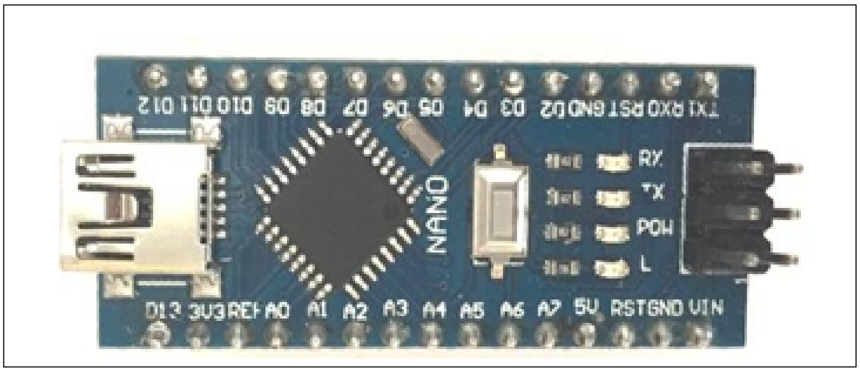
Note: * Commercial link only

Source: Authors, 2025

2.3.1 Arduino Nano

The Arduino Nano board is one of the smallest development boards in the Arduino family, applied in situations where significant processing power is not required or where a compact system is necessary. The Arduino Nano utilizes the Atmega328P microcontroller (Gheorghe & Stoica, 2021).

Figure 4 – Arduino Nano


D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 GND RST RX0 TX1
NANO
RX
TX
POW
L
D13 3V3 REF A0 A1 A2 A3 A4 A5 A6 A7 5V RST GND VIN

Source: Gheorghe & Stoica (2021)

The development board possesses 14 digital pins usable as input/output and 8 analog pins usable only as inputs. It has a total of 6 PWM pins distributed among the digital pins (Gheorghe & Stoica, 2021). This prototyping board has a 10-bit Analog-to-Digital Converter (ADC) resolution. It is possible to calculate the voltage resolution using the equation (Beckwith, 2007),

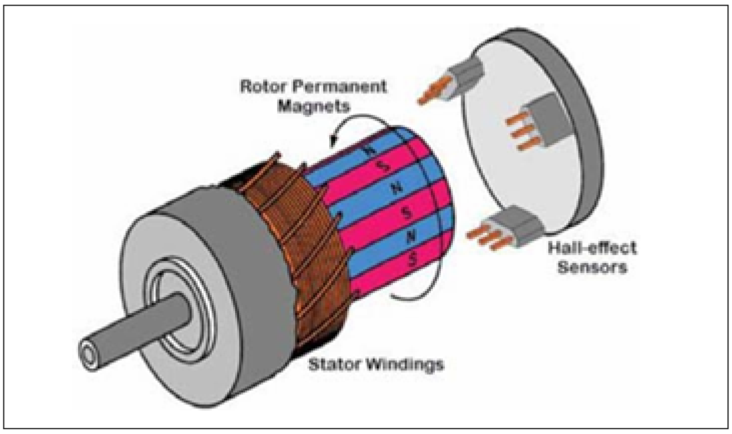
$$R = \frac{\text{Voltage Dynamic Range}}{2^n}, \tag{14}$$

where R is the system's voltage resolution and n is the number of bits. Knowing that the input voltage range (from the potentiometer) is 5 V and the number of bits is 10, applying Equation 14 yields a resolution of approximately 4.88 mV.

2.3.2 Brushless Motor

Brushless motors achieve electrical commutation using a permanent magnet rotor and a stator with a sequence of coils. In this motor design, a permanent magnet rotates while the current-carrying conductors (the coils) remain stationary (Mach et al., 2018).

Figure 5 – Design of a brushless motor



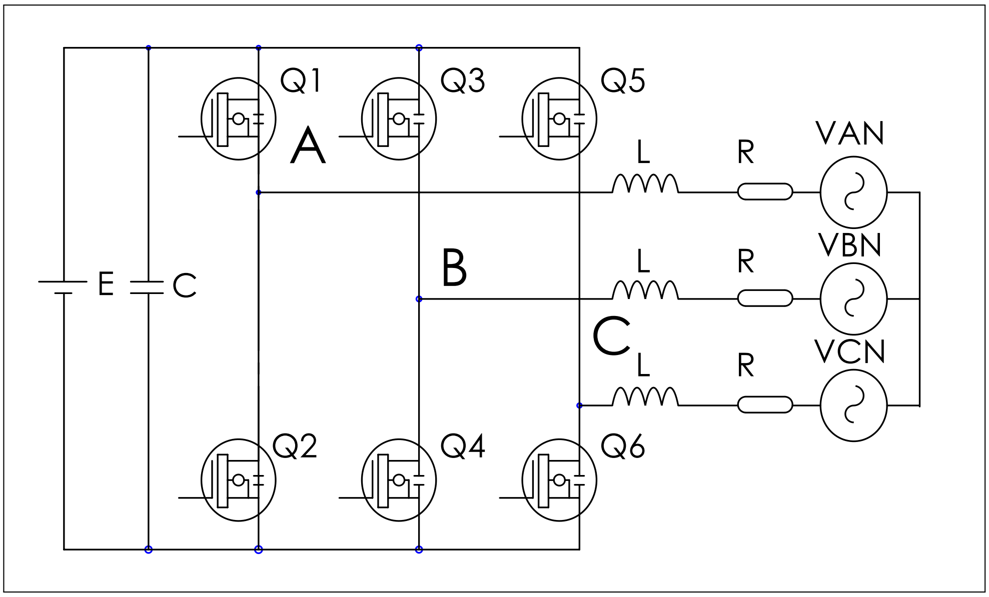
Source: Mach et al. (2018)

Brushless motors, known for their efficiency and high rotational speeds, are frequently powered by Lithium Polymer (LiPo) batteries. The operating voltage of these motors typically ranges between 10 and 14 volts. However, each cell of a LiPo battery provides only 3.7 volts (nominal). To achieve the required voltage, multiple cells are connected in series. The number of cells connected in series is indicated by the ”S” rating on the battery label, making it easy to identify the total nominal voltage (Mach et al., 2018).

2.3.3 Electronic Speed Controller (ESC)

As the name suggests, the Electronic Speed Controller (ESC) functions to vary the rotational speed of an electric motor. The ESC interprets the PWM signal sent by the microcontroller and generates the appropriate three-phase signals to drive the brushless motor (Mach et al., 2018).

Figure 6 – Electronic Speed Controller (ESC) Circuit Diagram



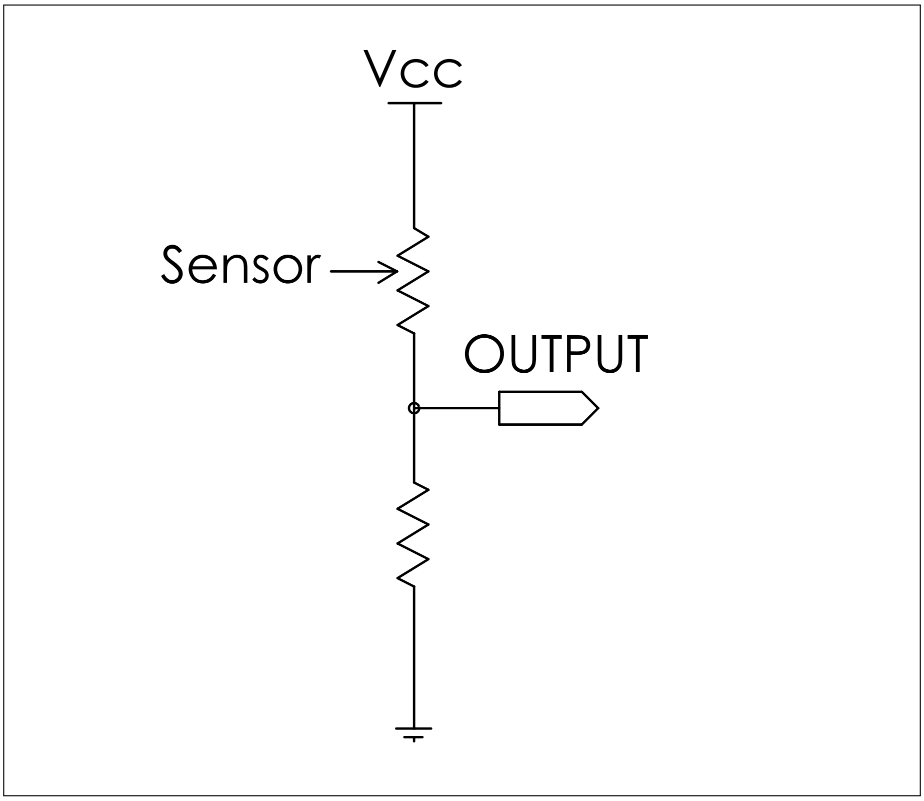
Source: Mach et al. (2018)

ESCs operate by modulating the power delivered to the motor phases. The input control signal typically uses pulse widths varying from 1 to 2 milliseconds to determine the desired speed level. The duty cycle relates to the proportion of time power is applied to the motor phases (Mach et al., 2018).

2.3.4 Potentiometer

The potentiometer measurement system is robust, simple, and reliable. The shaft of the component whose displacement is to be measured is mechanically coupled to the potentiometer, such that the rotation of the component alters the resistance, which in turn changes the output voltage of the circuit (Borges, 2008).

Figure 7 – Potentiometer Circuit Diagram



Source: Borges (2008)

This sensor configuration requires an external power source for operation. The sensor's output signal is a continuous voltage whose amplitude varies, classifying it as a continuous analog sensor (Borges, 2008).

3 METHODOLOGY

This chapter describes the methodological steps adopted for the development of the electromechanical control system applied to the monocopter, encompassing open-loop control, mathematical modeling of the plant, and closed-loop control.

3.1 Open-loop control

Initially, the plant's behavior was analyzed in open loop, without the application of feedback control. The monocopter was powered directly by a 12V-5A power supply, and the ESC was connected to a potentiometer to regulate the energy delivered to the motor. It was observed that when the monocopter arm reached the horizontal position, the mechanical arm could no longer be controlled, and the plant collapsed.

Afterwards, the potentiometer that controlled the power delivered to the ESC was replaced by a direct command executed via the Arduino Nano microcontroller interface. The input signal consisted of voltage values sent directly to the ESC using the microcontroller to generate PWM signals. During this stage, the basic dynamic characteristics of the monocopter were identified, such as the relationship between the PWM signal and the angular position response of the monocopter arm.

The collected data were analyzed to identify relevant behaviors, such as the motor's response time and the influence of external disturbances, allowing the determination of initial system parameters and possible control challenges.

3.2 Reduced-order mathematical model

Based on the results of the open-loop experiment, a reduced-order mathematical model was developed to represent the monocopter's dynamics using the reaction curve method. A Python script was created to record, in a ".csv" file, the values sent through the microcontroller's serial communication, namely: the timestamp of each acquisition in milliseconds, the resistance measured by the potentiometer (related to the angular position of the monocopter arm), and the PWM signal value defined by the user for the ESC. All values were normalized between 0 and 100% of the system's operational amplitude.

The model parameters were tuned through simulations and experimental validations using the data obtained in the previous stage. Additionally, the reduced-order modeling approach was chosen to balance the model's accuracy and the computational feasibility on the microcontroller, considering the Arduino Nano's hardware limitations.

The transfer function of a first-order system with time delay can be written as:

$$G(s) = \frac{\kappa}{\tau s + 1} \cdot e^{-\phi s}, \tag{15}$$

where κ is the system gain, τ is the time constant, and ϕ is the delay (transport time). The gain κ defines the system's response amplification, τ determines the speed at which the system reacts to a disturbance, and ϕ models the time lag between the input application and the output response.

3.3 Closed-loop control

With the mathematical model established, a digital PID controller was implemented directly on the Arduino Nano. The controller was designed to stabilize the monocopter by dynamically adjusting the motor speed to compensate for external disturbances and system variations.

The gains K_c, T_i, and T_d were initially determined using Table 2 and later refined through experimental testing. To handle sensor signal noise, an exponential moving average (EMA) digital filter was implemented. The EMA filter is defined by the following equation:

$$y(t) = \alpha \cdot x(t) + (1-\alpha) \cdot y(t-1), \tag{16}$$

where $y(t)$ represents the filtered value at time t, $x(t)$ is the raw value at time t, α is the smoothing factor ranging from $0 \leq \alpha \leq 1$, and $y(t-1)$ is the previous filtered value at time $t-1$. This filter provided a more stable and accurate reading of the plant, reducing the effects of unwanted noise without compromising the system's dynamic response.

Table 2 – Equations for standard PID controller coefficients

Methods	K_c	T_i	T_d
Ziegler-Nichols	$\frac{1.2 \cdot \tau}{K \cdot \theta}$	$2 \cdot \theta$	$\frac{\theta}{2}$
CHR	$\frac{0.6 \cdot \tau}{K \cdot \theta}$	τ	θ
Cohen-Coon	$\frac{1}{K}\left(4 \cdot \tau \cdot \theta + \frac{1}{4}\right)$	$\frac{\theta}{32 \cdot \theta + 8}$	$\frac{2 \cdot \theta}{11 + 2 \cdot \theta}$
3C	$\frac{1.370}{K \cdot \left(\frac{\theta}{\tau}\right)^{0.950}}$	$0.740 \cdot \tau \cdot \left(\frac{\theta}{\tau}\right)^{0.738}$	$0.365 \cdot \left(\frac{\theta}{\tau}\right)^{0.950}$

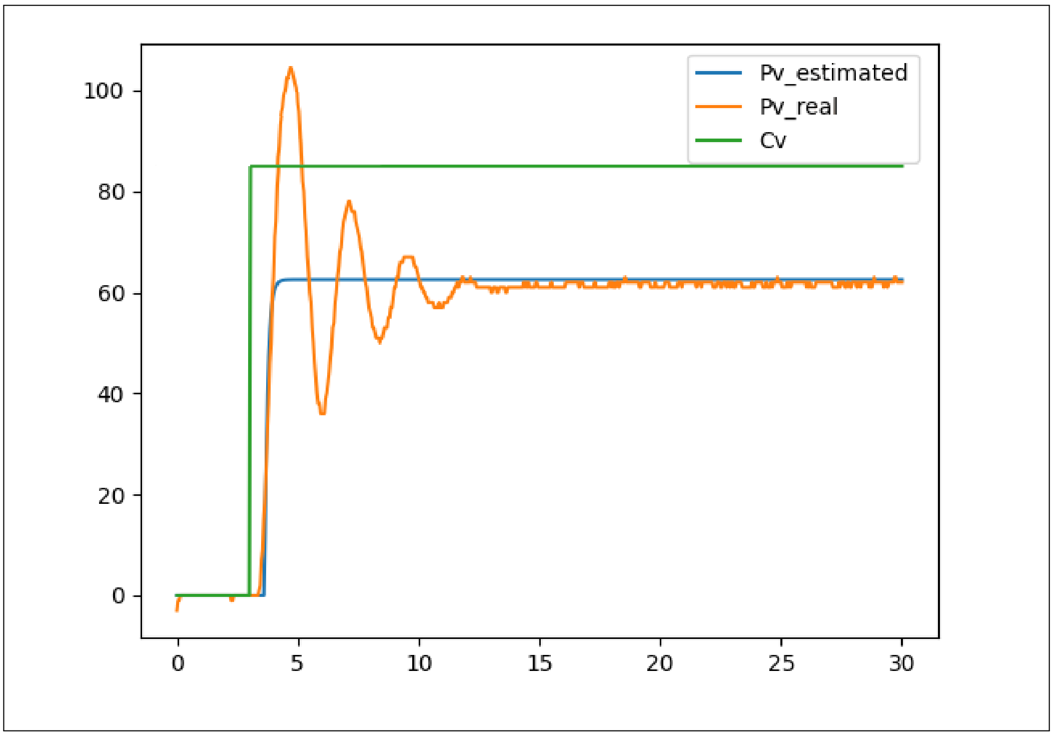
Source: Adapted from Garcia (2021)

4 RESULTS

In this section, the results obtained during the implementation of the control system for the monocopter are presented. Firstly, the reduced-order model obtained to describe the system dynamics was determined.

Figure 8 shows the response curve obtained for the plant when subjected to a unit step input. The input signal used in the system, shown in green in the figure, is a unit step. The system response, presented in orange, reflects how the system stabilizes after the step disturbance. The transfer function that approximates the system dynamics is represented in blue, illustrating the estimated system behavior.

Figure 8 – Identification of the plant's response curve to a unit step



Source: Authors, 2025

The transfer function obtained for the system was:

$$G(s) = \frac{0.7358}{0.1236s + 1} \cdot e^{-0.5744s}, \tag{17}$$

with the parameters considered for obtaining the gains being $\kappa = 0.7358$, $\tau = 0.1236$, and $\phi = 0.5744$, respectively.

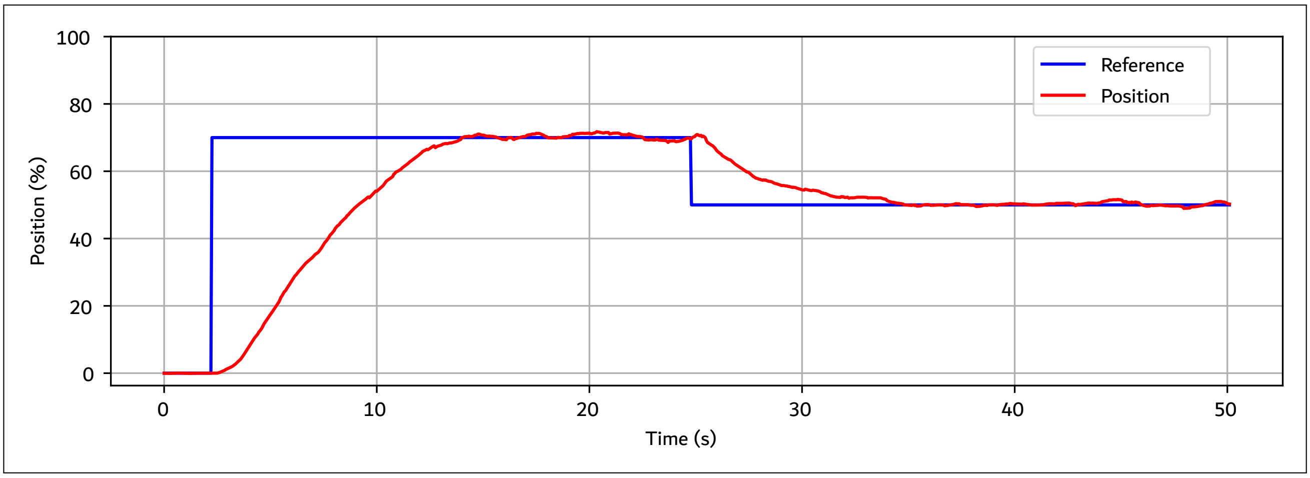
The gains K_c, T_i, and T_d, initially obtained from Table 2 using the CHR method, resulted in: $K_p \approx 0.175$, $T_i = 0.1236$, $T_d = 0.2872$. The system response was evaluated for different values of the PID controller parameters.

For the moving average filter, the value of the smoothing factor, α, that best suited the system was 0.08. This value was fixed before initiating the variation of the gain parameters for the PID controller.

The variation of the parameters K_c, T_i, and T_d was performed until obtaining a control model capable of stabilizing the plant even when the lever arm was in a position close to horizontal. The final values obtained for the controller were: $K_p = 0.05$, $T_i = 0.2$, $T_d = 0.8$. The controller's proportional gain (K_p) had to be significantly decreased to prevent the occurrence of *overshoot*. In some cases, the *overshoot* moved the plant's arm to positions above the horizontal, causing disturbances that resulted in the loss of system control.

For all gain tests, the closed-loop system performance was evaluated considering criteria such as response time, stability, and rejection of external disturbances. Figure 9 presents the response obtained for the plant's position (red line) and the defined *setpoint* (blue line) for the final combination of system parameters K_c, T_i, and T_d. The results demonstrated that the controller is capable of maintaining the monocopter's stability under different operating conditions.

Figure 9 – Closed-loop plant response with the final controller settings used



Source: Authors, 2025

5 CONCLUSION

The objective of this work was to design and implement a discrete PID control system for the monocopter using the Arduino platform. Through a well-defined methodology, the proposed goals were successfully achieved, including the development of a reduced-order model of the plant, its validation, and the organization of data from the sensors and actuators used in the plant.

The implementation of the discrete PID control system was successful, with satisfactory gain values for the controller: $K_p = 0{,}05$, $T_i = 0{,}2$, and $T_d = 0{,}8$. These parameters were adjusted to ensure system stability and prevent issues such as overshoot, which, in certain cases, negatively affected the monocopter's performance by introducing disturbances that resulted in a loss of control.

During the project development, several significant challenges were encountered. One of the main difficulties was programming certain components of the plant, such as the ESC (Electronic Speed Controller), which required particular attention in its implementation. Another critical challenge was adjusting values for discretizing the arm position range and the PWM signal sent to the motor, which required a meticulous approach to generate an accurate and functional model.

Despite these challenges, the system successfully met the project requirements, effectively stabilizing the monocopter even under disturbances. The implemented digital controllers were designed to reject external disturbances and ensure system stability, meeting the control and performance expectations.

This work makes a significant contribution to the field of digital control, particularly in the context of mechatronic systems. The study and implementation of discrete PID control not only improve the understanding of theoretical aspects but also hold great practical application potential, especially in autonomous systems and robotics. The success of this project opens doors for future research and improvements, focusing on new control methods and further system optimization, which could contribute to the advancement of similar systems in various engineering fields.

REFERENCES

4-Max (2024). A2212-13t 1000kv brushless motor. Acesso em: 06 nov. 2024.

AeroModel, S. (2024). Gws 8040 hd electric propeller. Acesso em: 06 nov. 2024.

AllDatasheet (2024). Arduinonano datasheet. Acesso em: 06 nov. 2024.

Arnold, F. J., Arthur, R., Bravo-Roger, L. L., Gonçalves, M. S., & Oliveira, M. J. G. d. (2011). Estudo do amortecimento do pêndulo simples: uma proposta para aplicação em laboratório de ensino. *Revista Brasileira de Ensino de Física*, 33:4311–4311.

Beckwith, T. (2007). *Mechanical Measurements*. Pearson Education.

Blucabos (2024). Blucabos - produtos e soluções. Acesso em: 06 nov. 2024.

Borges, J. F. A. (2008). Desenvolvimento de um medidor de ângulo de ataque para aeronaves de pequeno porte.

Caldeira, A. F. & Rech, C. (2024). Ufsm00388 (conteúdo da disciplina). Notas de aula.

Garcia, C. (2021). *Controle de processos industriais: estratégias convencionais*, volume 1. Editora Blucher.

Gheorghe, A. & Stoica, C. (2021). Wireless weather station using arduino mega and arduino nano. *The Scientific Bulletin of Electrical Engineering Faculty*, 21(1):35–38.

Intelbras (2022). Ef 1205 - datasheet. Acesso em: 06 nov. 2024.

Livre, M. (2024). Placa nano v3 com suporte borne cabo usb - compatível arduino. Acesso em: 06 nov. 2024.

Mach, V., Kovář, S., Valouch, J., & Adámek, M. (2018). Brushless dc motor control on arduino platform. *Przeglad elektrotechniczny*.

Merlin, L. (2024). Cano pvc marrom soldável 3m 1/2 20mm tigre. Acesso em: 06 nov. 2024.

Modelismo, B. (2024). Esc 40a brushless bec 3a lipo 2-4s completo com conectores. Acesso em: 06 nov. 2024.

Ogata, K. (2010). *Engenharia de Controle Moderno*. Pearson Prentice Hall, São Paulo. Acesso em: 19 out. 2024.

Technology, H. O. (2024). Wh148 potentiometer datasheet. Acesso em: 06 nov. 2024.

Vendas, E. (2024). Mancal com rolamentos p004 - man1006. Acesso em: 06 nov. 2024.

Author contributions

1 – Eagro Henrique Brenner Muller
Mechanical Engineering Academic, Universidade Federal de Santa Maria – Cachoeira do Sul
https://orcid.org/0009-0009-3248-6694 • eagro.muller@acad.ufsm.br
Contribution: Conceptualization; Methodology; Writing – Original Draft Preparation, Formal Analysis and Investigation

2 – Pedro Antônio Castro Regio dos Santos
Mechanical Engineering Academic, Universidade Federal de Santa Maria – Cachoeira do Sul
https://orcid.org/0009-0006-7768-0261 • pedro.regio@acad.ufsm.br
Contribution: Conceptualization; Methodology; Writing – Original Draft Preparation, Formal Analysis and Investigation

3 – André Francisco Caldeira
PhD in Electrical Engineering Professor of the Electrical Engineering Course, Universidade Federal de Santa Maria – Cachoeira do Sul
https://orcid.org/0000-0002-4939-2709 • andre.caldeira@ufsm.br
Contribution: Conceptualization, Methodology, Original Draft Preparation, Formal Analysis, Investigation, Review & Editing

4 – Charles Rech
PhD in Mechanical Engineering, Professor of the Mechanical Engineering Course, Universidade Federal de Santa Maria – Cachoeira do Sul
https://orcid.org/0000-0001-8523-6300 • charles.rech@ufsm.br
Contribution: Conceptualization, Methodology, Original Draft Preparation, Formal Analysis, Investigation, Review & Editing

5 – Simone Ferigolo Venturini
PhD Studies in Progress in Production Engineering, Universidade Federal de Santa Maria
https://orcid.org/0000-0002-9439-0008 • simone.venturini@ufsm.br
Contribution: Review & Editing

6 – Carmen Brum Rosa
Professor in the Graduate Program in Production Engineering (PPGEP), Universidade Federal de Santa Maria
https://orcid.org/0000-0002-0173-081X • carmen.b.rosa@ufsm.br
Contribution: Review & Editing

How to cite this article

Muller, E. H. B., & dos Santos, P. A. C. R., & Caldeira, A. F.., & Rech, C. (2025). Control System Design for a Monocopter Using Arduino Nano. *Ciência e Natura*, Santa Maria, v. 47, n. spe. 4, e92172. DOI: https://doi.org/10.5902/2179460X92172.