Attitude Determination and Control System Open-Source Simulator of a 1U CubeSat Nanosatellite

1^{st} Túlio Santos Resende
COPPE
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
tulio.resende@coppe.ufrj.br

2^{nd} Josué Lima da Silva
Department of Electroelectronics
CEFET-MG
Leopoldina, Brazil
josuelima@cefetmg.br

3^{rd} Alessandro Jacoud Peixoto
COPPE
Federal University of Rio de Janeiro (UFRJ)
Rio de Janeiro, Brazil
jacoud@poli.ufrj.br

4^{th} Murillo Ferreira dos Santos
Department of Electroelectronics
CEFET-MG
Leopoldina, Brazil
murillo.ferreira@cefetmg.br

5^{th} Ana Isabel Pereira
Research Centre in Digitalization and Intelligent Robotics (CeDRI), IPB
Bragança, Portugal
apereira@ipb.pt

6^{th} José Lima
Research Centre in Digitalization and Intelligent Robotics (CeDRI), IPB
Bragança, Portugal
jllima@ipb.pt

7^{th} Maurício Herche Fófano de Morais
Department of Electroelectronics
CEFET-MG
Leopoldina, Brazil
mauricio.fofano@gmail.com

2025 26th International Carpathian Control Conference (ICCC) | 979-8-3315-0127-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCC65605.2025.11022953

***Abstract*—This work developed an open-source educational simulator to evaluate control schemes for small CubeSat-class satellites of 1U size (1 unit). Attitude (orientation) estimation was achieved using the Three-Axis Attitude Determination (TRIAD) algorithm, commonly employed in competitions and real-world CubeSats. A simple Proportional (P) controller was designed to regulate the CubeSat's angular velocity. Additionally, for pointing control (line-of-sight orientation), a Proportional-Derivative (PD) controller was implemented. Numerical simulations were conducted to assess the performance of the CubeSat's Attitude Determination and Control Subsystem (ADCS).**

***Index Terms*—CubeSat, TRIAD Algorithm, Modified PD Controller, Attitude Determination, Open-Source Software**

I. Introduction

The main challenges in satellite development and launch are the high project costs and the issues related to space debris. Additionally, there is a shortage of skilled human resources capable of managing these technologies, mainly due to the complexity of large and medium-sized satellite projects compared to small satellites. To address these challenges, CubeSats were developed in 1999 through a partnership between California Polytechnic State University and Stanford University [1].

CubeSats are small satellites named for their standard 1U cube structure, which measures 10 cm^3 and has a weight limit of 1.33 kg [2]. However, their small size and low cost make CubeSats an ideal proposition for academic and educational environments, gaining popularity among universities.

CubeSats are organized into modules that house the subsystems essential for their missions. Typically, four subsystems are considered vital for satellite operation: the Energy Power Supply (EPS), the On-Board Computer (OBC), telemetry, tracking and command (TT&C or COMM), and the Attitude Determination and Control System (ADCS).

This work focuses on the ADCS, which determines and controls the satellite's orientation (or attitude) in space. Regarding attitude, it is defined as the angular displacement of a body or object relative to another coordinate system.

Several techniques are used for attitude determination, each with its characteristics, advantages, and disadvantages. CubeSats' most commonly employed methods include the Kalman Filter, the SDRE (State Dependent Riccati Equation) Filter, and the TRIAD (Tri-Axis Attitude Determination) algorithm.

Maintaining CubeSat stability is crucial for adequately functioning the telemetry subsystem, which acts as a communication bridge between the CubeSat and mission control. Proper orientation is required for the space segment to send

The authors are also grateful to CeDRI (UID/05757), SusTEC (LA/P/0007/2021), SmartProduce project, funding by COMPETE2030-FEDER-01204700. They also thank the FAPEMIG for the financial support funded by project number APQ-01202-23.

telemetry to the ground segment (or vice versa). Similarly, in more complex missions like remote sensing, the camera must be stabilized and pointed toward Earth to capture images. Proportional (P), integral (I), and derivative (D) controllers and their combinations are well-established control techniques in aerospace applications, offering simplicity and ease of implementation.

The standard 1U unit was chosen for the CubeSat structure to simplify system dynamics, as the center of mass coincides with the geometric center of the body, resulting in symmetrical moments of inertia along the three principal axes. This symmetry allows for the design of identical controllers for each axis.

The model developed in this work is a reinterpretation of the CubeDesign Virtual 2020 competition model, where the focus was exclusively on developing the ADCS [3]. The model consists of five main blocks: control (dynamics, kinematics, and reaction wheel), mechanical modeling, and attitude determination.

Over the last few decades, numerous studies have been conducted to improve techniques and strategies for artificial satellites' attitude control and determination. For example, [4] employed a PD controller with an Extended Kalman Filter, using a magnetorquer as the control mechanism. In [5], a PID controller combined with an SDRE filter was used, with reaction wheels as actuators. [6] explored using B-dot and Fuzzy controllers (without an attitude determination algorithm), using a magnetorquer and reaction wheels as actuators. [7] implemented a PD controller with an SDRE filter, again using a magnetorquer as the control mechanism. A combination of solar and magnetometer sensors was used in all these cases.

In this work, an open-source CubeSat simulator[1] was developed based on the CubeDesign Virtual 2020 competition simulator [3]. The CubeSat modeling is divided into four parts: reaction wheel modeling, CubeSat dynamics and kinematics, and the TRIAD algorithm (attitude determination). Fig. 1 illustrates the developed computational model:

The TRIAD algorithm, a three-axis attitude determination method, was implemented using solar sensor and magnetometer measurements. While Euler angles can be directly obtained by integrating the CubeSat's angular velocity for small angles, the TRIAD algorithm was chosen for implementation in the simulator, as it is commonly used in real CubeSats [8].

From a control perspective, the control action is applied to three reaction wheels (one for each CubeSat axis), and the study focuses on the effects of P control for regulation and a modified PD control for CubeSat pointing.

Remark 1: **(Notation)** Hereafter, a time function $f(t)$ has a Laplace transform, sometimes denoted as $f(s)$ or $F(s)$.

[1]https://github.com/Tulio-Resende/Modelo_CubeSat_Open_Source

II. Considered Models

A. Reaction Wheel Model

In this work, it is assumed that the CubeSat's actuator consists of a reaction wheel on each axis, driven by a DC motor with a torque constant K_m and an armature current $I_{rr}(t)$, limited between -0.5 and $+0.5A$. The rotor dynamics, defined by the moment of inertia J_r and the viscous friction coefficient b, are given by.

$$J_r\dot{\omega}_r(t) + b\omega_r(t) = K_m I_{rr}(t)\,, \tag{1}$$

where $\omega_r(t)$ is the rotor's angular velocity.

The reaction wheel includes an internal PI controller with gains $K_{p\omega}$ and $K_{i\omega}$ to control $\omega_r(t)$. Table I lists the controller gains used in the simulator.

TABLE I
Reaction Wheel Parameters.

Parameter	Value
J_r	$3.00 \times 10^{-6} Kgm^2$
b	$5.16 \times 10^{-6} Nms/rad$
K_m	$2.64 \times 10^{-2} Nm/A$
K_{iw}	1.77×10^{-1}
K_{pw}	4.67×10^{-2}

The control loop for the reaction wheels operates internally within the CubeSat's attitude and pointing control loops. Since reaction wheel dynamics are much faster than the satellites, the external control loops assume the reaction wheel's transfer function as unit gain.

Let τ_{rr} and L_{rr} represent the torque and angular momentum generated by the reaction wheel. The transfer functions from I_{rr} to τ_{rr} and from I_{rr} to L_{rr} are simplified as.

$$\frac{\tau_{rr}(s)}{I_{rr}(s)} = -K_m, \tag{2}$$

$$\frac{L_{rr}(s)}{I_{rr}(s)} = \frac{K_m}{s}. \tag{3}$$

B. Dynamic Model

The conservation of angular momentum equation is utilized to model the satellite's rotational dynamics. The resulting torque is expressed as a combination of the external torque τ_{ext}, the reaction wheel torque τ_{rr}, and the gyroscopic torque $\omega \times (J\omega + L_{\text{rr}})$, as given by.

$$\dot{\omega} = J^{-1}\left[\tau_{\text{ext}} - \tau_{\text{rr}} - \omega \times (J\omega + L_{\text{rr}})\right]\,, \tag{4}$$

where ω represents the angular velocity of one of the axes (i.e., ω_x, ω_y, or ω_z), and J denotes the inertia tensor.

Equation (4) represents the complete system model. However, specific considerations allow for simplification. First, CubeSats typically exhibit small angular velocities due to energy constraints in space, necessitating minimal power usage for rotation. Consequently, terms dependent on angular velocity are neglected, allowing the gyroscopic torque to be considered negligible, i.e., $\omega \times (J\omega + L_{\text{rr}}) = 0$.

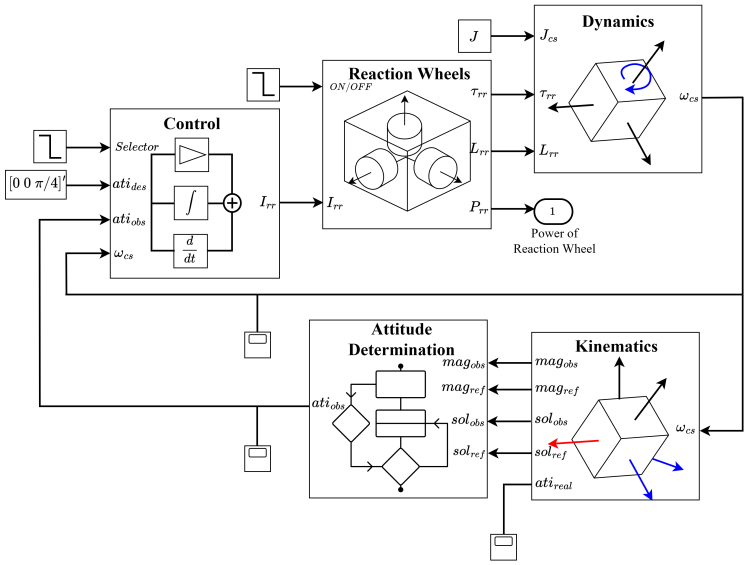

Fig. 1. Representation of the computational model.

Additionally, various external torques can occur in Earth's orbit, such as those from the Earth's magnetic field, gravity gradient torque, or aerodynamic torque [8]. However, this work considers the external torque negligible due to the relatively low intensity of these torques compared to the torque generated by the reaction wheels.

Thus, with $\tau_{\text{ext}} = 0$, the angular velocity of the system can be controlled solely by the reaction wheel torque τ_{rr}, leading to.

$$\dot{\omega} = J^{-1}[-\tau_{\text{rr}}]. \tag{5}$$

The dynamics block in Fig. 1 has three input parameters: the inertia tensor, the torque from the reaction wheel (τ_{rr}), and the angular momentum of the reaction wheel, with the latter two being sourced from the reaction wheels block. Additionally, the block has a single output: the angular velocity (ω). To derive the transfer function from τ_{rr} to ω, an integration is performed, resulting in.

$$\frac{\omega(s)}{\tau_{rr}(s)} = \frac{1}{Js}. \tag{6}$$

C. Kinematic Model

According to [9], attitude refers to the movement and orientation of a body relative to a body-centered reference frame, measured within another reference system that is assumed to be inertial, using a set of parameters. Various methods exist to represent attitude and its transformations, with the most commonly used being the rotation matrix, Euler angles, quaternions, and equivalent axis angles [10]. This work employs Euler angles in the "xyz" rotation sequence, resulting in the angles ϕ, θ, and ψ.

The following matrix gives the relationship between the angular velocity components along the three axes (ω_x, ω_y, and ω_z) and the derivatives of the three Euler angles.

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} + \begin{bmatrix}\cos(\psi)\cos(\theta)-1 & \sin(\psi) & 0\\ -\cos(\theta)\sin(\psi) & \cos(\psi)-1 & 0\\ \sin(\theta) & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}. \tag{7}$$

In (7), the sines and cosines of coupled angles introduce non-linearities. To apply a linear control technique, this work linearizes the trigonometric functions using a second-order Taylor series expansion, yielding.

$$\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} = \begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} + \begin{bmatrix}-\frac{\psi^2}{2}-\frac{\theta^2}{2} & \psi & 0\\ -\psi & -\frac{\psi^2}{2} & 0\\ \theta & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}. \tag{8}$$

Considering that satellites typically rotate at small angles, the second term in (8) can be neglected, leading to $\left[\omega_x, \omega_y, \omega_z\right]^T \approx \left[\dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T$.

This approximation indicates that angular velocity can be expressed in terms of the derivatives of the Euler angles. Thus, Euler angles can be obtained by directly integrating the angular velocity derived from the dynamics model. It is crucial to specify the initial attitude, which in this work is considered with all three Euler angles set to zero.

III. Attitude Determination Block

The main idea of the TRIAD algorithm is to calculate the misalignment between the vectors of the reference coordinate system (director vectors u_1, u_2, and u_3) and the body-fixed

coordinate system of the satellite (described by the director vectors w_1, w_2, and w_3).

The director vectors of the reference system (u) are calculated from the reference values of the solar (r_1) and magnetic (r_2) sensors. These values are obtained *a priori* based on the satellite's position in orbit. The director vectors of the reference system are given by.

$$u_1 = r_1, \tag{9}$$
$$u_2 = u_3 \times r_1, \tag{10}$$
$$u_3 = \frac{r_1 \times r_2}{||r_1 \times r_2||}. \tag{11}$$

For simplification purposes, the orbital motion has been disregarded; therefore, the vectors r_1 and r_2 are fixed and are described in Table II.

TABLE II
REFERENCE SOLAR AND MAGNETIC SENSOR DATA.

Sensors	Value
Solar (r_1)	$[2 \quad -5 \quad 3]'$
Magnetometer (r_2)	$[5 \quad -1 \quad 1]'$

The director vectors of the body-fixed system (w) of the satellite are calculated from the measured values of the solar (b_1) and magnetic (b_2) sensors. To simulate the sensor values, the reference vectors (r_1 and r_2) were rotated according to the attitude of the CubeSat. The director vectors of the body-fixed system are given by.

$$w_1 = b_1, \tag{12}$$
$$w_2 = w_3 \times b_1, \tag{13}$$
$$w_3 = \frac{b_1 \times b_2}{||b_1 \times b_2||}. \tag{14}$$

If the reference and observed vectors indicate the same value, the systems are aligned, i.e., the three Euler angles describing the attitude are zero. On the other hand, if they are different, there is a difference in attitude between the two systems.

Once the director vectors of the coordinate systems u and w are calculated, the corresponding rotation matrices (C_{bw} and C_{ru}) and matrix that relates these systems, C_{br} can be obtained as follows.

$$C_{bw} = [w_1 \quad w_2 \quad w_3], \tag{15}$$
$$C_{ru} = [u_1 \quad u_2 \quad u_3], \tag{16}$$
$$C_{br} = C_{bw} C_{ru}^T. \tag{17}$$

Finally, the following equations are used to calculate the Euler angles.

$$\phi = \arctan\left(-\frac{C_{br(3,2)}}{C_{br(3,3)}}\right), \tag{18}$$
$$\theta = \arcsin(C_{br(3,1)}), \tag{19}$$
$$\psi = \arctan\left(-\frac{C_{br(2,1)}}{C_{br(1,1)}}\right). \tag{20}$$

The attitude determination block shown in Fig. 1 receives four parameters: b_1, r_1, b_2, and r_2. The output of this block represents the three Euler angles ϕ, θ, and ψ.

Note that the Euler angles could have been obtained by directly integrating the angular velocity of the CubeSat. However, this work contributes to implementing the TRIAD algorithm in the simulator, the scheme used in the real CubeSat.

IV. CONTROL BLOCK

This section outlines the control strategies for regulating the CubeSat's angular velocity and executing its pointing. A regulation control is employed to stop the CubeSat, assuming it is rotating. Once stabilized, the pointing control is activated.

A. Angular Velocity Regulation Control

A Proportional (P) controller with gain $K_{pe} > 0$ was selected for regulation due to the integrative term in the plant's transfer function, as illustrated in Fig. 2.

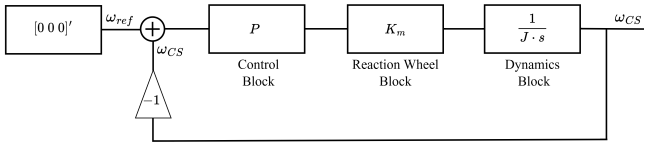

Fig. 2. Representation of the regulation control loop for a P controller.

The closed-loop system function is expressed as.

$$\frac{\boldsymbol{\omega}(s)}{\boldsymbol{\omega}_{ref}(s)} = \frac{1}{\frac{J}{K_{pe}K_m}s + 1}. \tag{21}$$

where $\boldsymbol{\omega}$ denotes the angular velocity along one of the axes (ω_x, ω_y, or ω_z).

The closed-loop time constant (τ) is given by.

$$\tau = \frac{J}{K_{pe}K_m}. \tag{22}$$

The regulation control, which breaks the CubeSat's motion, is activated 10 seconds after the simulation begins. With a controller switching time of $130s$, the regulation control was designed with a $50s$ for the settling time, ensuring smooth rotational movement without exceeding the current limits of the reaction wheel. The proportional gain was calculated as $K_{pe} = 8.3 \times 10^{-3}$, with separate controllers implemented for each principal axis. Table III summarizes the parameters used.

TABLE III
REGULATION CONTROL PARAMETERS.

Parameter	Value
J	$2.2 \times 10^{-3} Kgm^2$
τ	$10s$
T_s	$50s$
ω_{ref}	0 (x), 0 (y), 0 (z) rad/s
$\omega(0)$	0 (x), 60×10^{-3} (y), 0 (z) rad/s

B. Pointing Control

Pointing control directs the payload module to the desired target, a critical function in real satellite projects. A modified PD controller was selected, with the derivative term connected to the feedback signal to prevent spikes in the control signal. To avoid excessive oscillations and sudden responses that could lead to reaction wheel saturation, a maximum overshoot M_p of 20% and a settling time T_s of 50 s were specified. The design parameters are presented in Table IV.

TABLE IV
POINTING CONTROL PARAMETERS.

Parameter	Value
α_{ref}	0 (roll), $\pi/4$ (pitch), 0 (yaw) rad
M_p	20%
T_s	$50s$
ω_n	$1.75 \times 10^{-1} Hz$
ξ	4.6×10^{-1}

The gains $K_{pa} = 2.56 \times 10^{-3}$ and $K_{da} = 1.3 \times 10^{-2}$ were chosen to meet the desired settling time and overshoot criteria. Fig. 3 shows the derivative action implemented with a "dirty" derivative using a first-order filter.

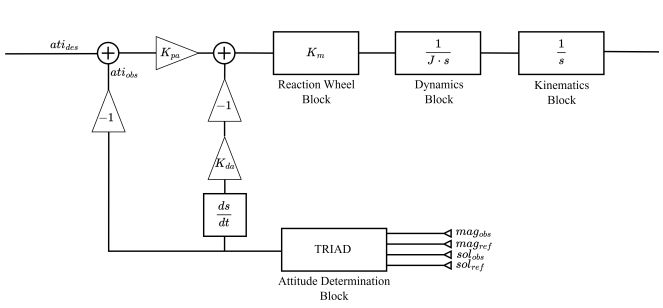

Fig. 3. Representation of the pointing control loop for a modified PD controller.

Let α represent one of the Euler angles (ϕ, θ, or ψ) and α_{ref} the corresponding desired value. The closed-loop transfer function from α_{ref} to α is.

$$\frac{\alpha(s)}{\alpha_{ref}(s)} = \frac{K_{pa}K_m}{Js^2 + K_m K_{da} s + K_{pa}K_m}. \tag{23}$$

The modified PD controller prevents control signal spikes and eliminates the unwanted zero present in traditional PD control.

V. EXPERIMENTAL RESULTS

The results are presented in two stages: the first involves simulating and analyzing the regulation controller's primary data, while the second focuses on the pointing controller's simulation and analysis.

A. Analysis of Regulation Control

The regulation control initiates 10 seconds into the simulation and aims to halt the CubeSat's movement from an initial constant angular velocity. Fig. 4 shows a constant angular velocity of 60×10^{-3} rad/s during the first 10 seconds, with a steady-state velocity of zero, consistent with the controller parameters in Table III. The settling time was approximately $50s$.

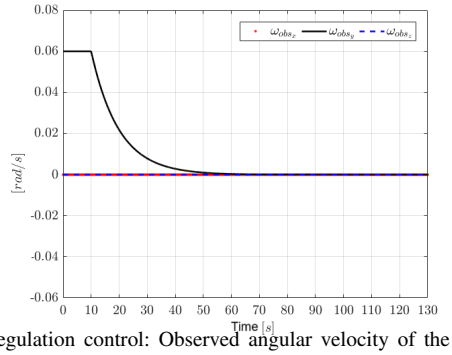

Fig. 4. Regulation control: Observed angular velocity of the CubeSat over time. ω_x in black, ω_y in red, and ω_z in blue.

Fig. 5 illustrates the system's torque relationships. Due to a relatively low K_m, the torque intensity is minimal, around $-2 \times 10^{-5} Nm$. The torque mainly results from the reaction wheel's torque, with negligible gyroscopic torque, validating the simplifications.

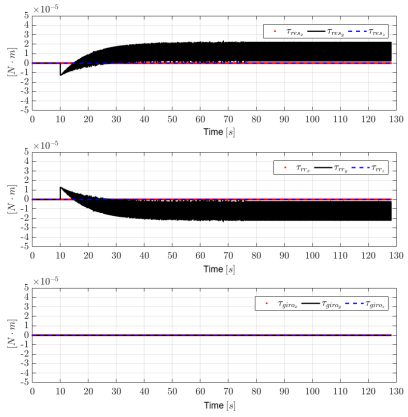

Fig. 5. Regulation control: Resultant torque (top), reaction wheel torque (middle), and gyroscopic torque (bottom). $x - axis$ in red, $y - axis$ in black, and $z - axis$ in blue.

Fig. 6 displays the angular displacement transient generated by the initial angular velocity in the y-axis. It can be observed that the angle θ follows the variation of angular velocity, with complete stabilization occurring near the settling time of 50 s. Furthermore, the strong similarity between the actual attitude and the one estimated by the TRIAD algorithm highlights the

proper functioning of the implemented approach. The actual and TRIAD algorithm-estimated attitudes align closely with the attitude error near zero.

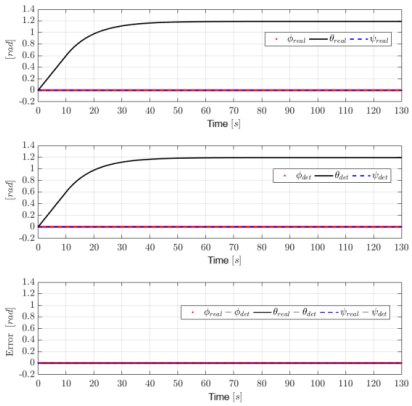

Fig. 6. Regulation control: Actual attitude (top), estimated attitude by the TRIAD algorithm (middle), and error between both (bottom). $x-axis$ in red, $y-axis$ in black, and $z-axis$ in blue.

B. Analysis of Pointing Control

Pointing control begins $130s$ into the simulation. Fig. 7 compares actual and estimated attitudes, showing that the TRIAD algorithm's estimates closely match the exact attitude, with negligible differences.

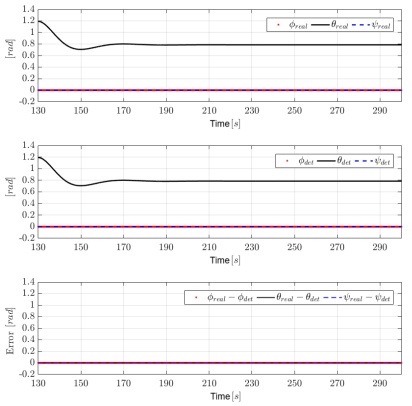

Fig. 7. Pointing control: Actual attitude (top), estimated attitude by the TRIAD algorithm (middle), and error between both (bottom). $x-axis$ in red, $y-axis$ in black, and $z-axis$ in blue.

The controller is designed to meet the settling time and achieve the reference value ($\pi/4 \approx 0.8$), with a maximum overshoot limited to 20% of the reference ($\approx 0.16 rad/s$). The oscillation around $t = 150s$ aligns with the design parameters, confirming its validity.

Fig. 8 illustrates the angular velocity for pointing the satellite using a PD controller, showing higher intensity at the beginning to stabilize the velocity near the projected settling time. The PD controller fine-tunes the system's response, ensuring a smooth transition while minimizing oscillations. The desired attitude, constrained to the y-axis (pitch), is reflected in the angular velocity along the same axis, highlighting the system's stabilization behavior.

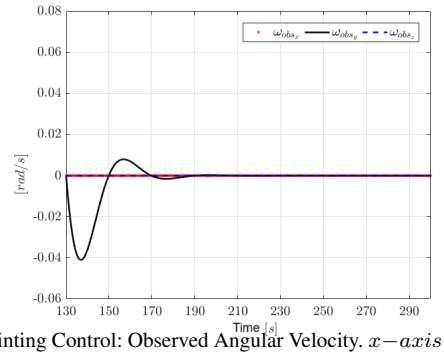

Fig. 8. Pointing Control: Observed Angular Velocity. $x-axis$ in red, $y-axis$ in black, and $z-axis$ in blue

VI. Conclusion and Future Work

An open-source CubeSat simulator was developed based on the CubeDesign Virtual 2020 competition simulator, focusing on implementing the TRIAD attitude determination algorithm. This algorithm, integrating solar sensor and magnetometer data, achieved high precision with negligible error. Two controllers were designed: a P-type for angular velocity regulation and a modified PD-type for orientation. The applied simplifications proved effective, with minimal performance impact. The system met key requirements, including settling time and maximum overshoot. Future work includes evaluating computational performance and developing advanced techniques with optimized power consumption, considering space environment constraints.

Acknowledgment

The authors thank UFRJ and CEFET-MG for their support and collaboration.

References

[1] S. po CubeSAT, "Cubesat design specification," *CubeSat standard" CubeSat Design Specification"]. California, USA, ver*, vol. 13, no. 20.02, 2014.

[2] T. Villela, C. A. Costa, A. M. Brandão, F. T. Bueno, and R. Leonardi, "Towards the thousandth cubesat: A statistical overview," *International Journal of Aerospace Engineering*, vol. 2019, no. 1, p. 5063145, 2019.

[3] INPE, "/cubedesign virtual 2020," 2022, https://www.citedrive.com/overleaf [Acessed in: 05/01/2025].

[4] W. d. V. C. Viegas, D. A. d. Santos, R. Waschburger, and J. Waldmann, "Controle em três eixos para aquisição de atitude por satélite universitário partindo de condições iniciais desfavoráveis," *Sba: Controle & Automação Sociedade Brasileira de Automatica*, vol. 23, pp. 231–246, 2012.

[5] B. D. R. de Mesquita, "Análise de estimação e controle de atitude em modo de operação nominal do conasat por filtro sdre e controle pid," *Master Degree (INPE)*, 2017.

[6] J. Li, M. Post, T. Wright, and R. Lee, "Design of attitude control systems for cubesat-class nanosatellite," *Journal of Control Science and Engineering*, vol. 2013, no. 1, p. 657182, 2013.

[7] M. Abdelrahman and S.-Y. Park, "Integrated attitude determination and control system via magnetic measurements and actuation," *Acta Astronautica*, vol. 69, no. 3-4, pp. 168–185, 2011.

[8] N. S. Takahashi, "Metodologia de desenvolvimento de um determinador de atitude portátil de baixo custo para interfaces homem-máquina," *Universidade Estadual de Londrina*, 2011.

[9] V. Carrara, "Cinemática e dinâmica de satélites artificiais," *São José dos Campos: INPE*, 2012.

[10] G. Arantes Jr, "Estudo comparativo de técnicas de controle de atitude em três eixos para satelites artificias," *Master of Science Thesis INPE-12970-TDI/1018 (in portuguese), INPE, S. Jose dos Campos, Brazil*, 2005.