International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:15, No:1, 2021

A Quadcopter Stability Analysis: A Case Study Using Simulation

C. S. Bianca Sabrina, N. Egidio Raimundo, L. Alexandre Baratella, C. H. João Paulo

***Abstract*—**This paper aims to present a study, with the theoretical concepts and applications of the Quadcopter, using the MATLAB simulator. In order to use this tool, the study of the stability of the drone through a Proportional - Integral - Derivative (PID) controller will be presented. After the stability study, some tests are done on the simulator and its results will be presented. From the mathematical model, it is possible to find the Newton-Euler angles, so that it is possible to stabilize the quadcopter in a certain position in the air, starting from the ground. In order to understand the impact of the controllers gain values on the stabilization of the Euler-Newton angles, three conditions will be tested with different controller gain values.

***Keywords*—**Controllers, drones, quadcopter, stability.

I. Introduction

SINCE the beginning of modern aviation, the constant search for advances in aerodynamic technologies, control and stability of air vehicles has allowed the improvement in flight performance of vehicles heavier than air, expanding their operating limits and opening up possibilities for new applications for existing tools. The recent development in high capacity batteries, small electric motors, microprocessor processing capacity, and miniaturization of electronic and micro electromechanical sensors has immensely reduced the cost of these technologies, allowing the market to expand unmanned aerial vehicles (UAVs) [1].

Long associated with military use (around the 1980s), new civilian applications for these aircraft are being developed recently. In applications as remote sensing for large areas the use of UAVs presents commercial advantages over alternative technologies, greater versatility and image quality than satellites and a lower cost and less risk to users than using manned aircraft [2].

The drone is an unmanned and remotely controlled aerial vehicle that can perform numerous tasks. Used both in wars and to deliver pizza, this equipment is increasingly present in different parts of the world.

The drone has become a very important vehicle that is growing more and more these days. Its great development has opened up many opportunities for future generations. Nowadays the drone has several features, for example, in agriculture it can serve to demarcate from planting [2], in an event take photos and videos, on Amazon [3] it is used for parcel delivery, and several other things. To perform these numerous tasks, the drone first has to be well stabilized. In order to stabilize the drones, there are several methods created by researchers. It is worth noting that an aerial vehicle is not well stabilized, there will be several errors during the flight, with these errors it will not be possible to perform its activities correctly, possibly even damaging the device.

The objective of this work is to tune the flight control and stability system for the aircraft in question, as an alternative to the standard tuned controller existing in the system in the initial conditions. The controlled variables are the pitch, roll and yaw angles, the most important angles in describing the system behavior.

The design of a control system traditionally involves two steps: the identification and modeling of the system to be controlled and the design of the appropriate controller. The identification consists of mathematical modeling using measured input and output data in system tests, and model validation by comparing the dynamic response of a simulation to the response of the physical system. A modern data-based control strategy is also addressed, which does not require the identification of a system model. However, in this work only the behavior of the controls will be evaluated according to the variation of the controller's gains, in order to understand the influence of each parameter on the behavior of the variables to be controlled.

As a computational tool used to perform the necessary operations and to facilitate all stages of this work, the MATLAB software, version 2019a was used to implement all the numerical methods performed, and the entire analysis of results.

In order to make this work possible, it was separated into some chapters for better organization. Chapter II describes about some types of drones and their main characteristics. Chapter III describes the materials and methods used. Chapter IV describes the simulations and comparisons made. Finally, Chapter V describes the conclusion and future work.

II. Drones

This chapter describes the concepts and definitions of this paper.

A. Single Rotor Drones

Single rotor drones have as their main feature the presence of only one propeller, thus managing to keep the drone stabilized. This model is also present in some reduced helicopters, which are common in the case of hover flights [4]. This type of drone is considered more efficient than those with

C. S. Bianca Sabrina, N. Egidio Raimundo, L. Alexandre Baratella, and C.H. João Paulo are with the Department of Industrial Automation, National Institute of Telecommunications (INATEL), CEP 37.540-000, Santa Rita do Sapucaí, MG, Brazil (e-mail: biancasilva@gea.inatel.br, egidio.neto@inatel.br, baratella@inatel.br, joao.paulo@inatel.br).

multiple rotors, as they can generally fly higher, generate thrust more effectively and have greater flight range. In addition, this type of aircraft can use a gas engine, which is more cost-effective. Fig. 1 shows an example of single-rotor drone [4].

Fig. 1 Single-rotor drone [4]

B. Multi-Rotor Drones

The multirotor is a drone that has more than one rotor to maintain its position and is more common in drones of the rotary type. The multirotor drones are easier to pilot and present lower cost when compared to other types of drones. Due to this features the multirotor is one of the most common types of drones for sale on the market for both, amateurs and professionals. These drones are able to remain in a stable position in the air for a regular period, being equipment popularly chosen for photography, filming and inspections. Based on the number of rotors they have on their platform, they can have alternative names. [4]

1. Tricopter

The tricopter has as feature three controllers, four gyroscopes and a servo. Each rotor is usually placed at the ends of the Drone's arms next to a location sensor. Fig. 2 shows an example of tricopter [5].

Fig. 2 Tricopter [5]

2. Quadcopter

This multirotor has four blades, two motors usually moving in a clockwise direction and the other two in a counterclockwise direction, always at opposite ends transversely. This helps to maintain the stability of the structure. Fig. 3 shows an example of quadcopter.

Fig. 3 Quadcopter [4]

3. Hexacopter

For the hexacopter the same occurs, three of the motors work in a clockwise direction and the other three in an anti-clockwise direction. Because of that, they are able to present greater lifting force than quadcopters. They present a good landing stability. Fig. 4 presents an example of hexacopter [6].

Fig. 4 Hexacopter [6]

4. Octacopter

The octacopter offers superior flight capabilities compared to the units seen previously, in addition to being much more stable, enabling regular shooting at high altitudes. Fig. 5 shows an example of octacopter [7].

Fig. 5 Octacopter [7]

C. Fixed Wing Drones

In the case of this model, it is the same as standard aircraft, thus obtaining the central body and two wings on the sides. The best feature about this is that it has a much greater flight capacity than the drones that are in other categories. Thus, it brings the possibility of its use for much longer. Fig. 6 shows an example of a fixed-wing drone.

Fig. 6 Fixed-wing Drones [4]

It is important to highlight that the greater the number of rotors, the lesser the flight range of an aircraft will be. In general, multirotor has problems with this characteristic, managing to remain in the air for about 20 to 30 minutes.

III. Materials and Methods

This chapter describes the materials and methods used in this paper.

A. The Drone Parrot

The PARROT minidrones are ultra-compact dynamic flight

systems with four propellers that make them a quadcopter and it can be controlled from a smartphone or tablet. They can be made to be very stable. It can combine signals from gyroscopes and 3-axis accelerometers, pressure sensor for flight altitude, ultrasonic sensor for precision flight close to the ground and a downward facing camera that can be used for optical flow and image processing.

The PARROT drone opens up an unlimited world of possibilities for learning science, technology, engineering, mathematics and programming. In addition, it is very easy to pilot indoors and outdoors. [8]

The minidrone is very versatile in both piloting and stability. It is 3.2 cm long and 4 cm high, with a flight time of at least one hour being defined, its weight is only 63 grams and has a range of up to 100 meters [4]. The drone can be seen in Fig. 7.

Fig. 7 Minrotone from PARROT [8]

B. MATLAB Simulink and Its Toolbox

A MATLAB tool is used for high-level programming, making it easier for programmers to implement different types of programming.

Simulink, as part of the many features of MATLAB, is a block diagram environment for simulating multiple domains and model-based design. This tool supports design, simulation, automatic code generation, testing and continuous verification of embedded systems. Besides that, it provides a graphical editor, customizable block libraries and solvers for modeling and simulating dynamic systems.

The Simulink add-ons are MathWorks tools that expand the features offered by Simulink. The Simulink product family includes complementary tools for, physical modeling, control systems modeling, signal processing, code generation, real-time simulation and real time testing, verification and validation, and 3D visualization.

Of the more than 40 complementary tools of Simulink, it will be focused on two that will be used to carry out this work specifically. The Aerospace Blockset is a complementary tool to Simulink that expands its resources with blocks for modeling and simulating UAVs. Simulink Coder is a complementary tool that generates and executes C and C ++ code from Simulink diagrams, Stateflow® graphics and MATLAB functions. The generated source code can be used for real-time applications, including simulation acceleration, rapid prototyping and hardware simulation in the loop [9].

C. The Simulator

MathWorks developed support for PARROT minidrones to help researches, teachers and instructors to teach students in model-based design using drones as a popular hardware platform, to help industry innovators understand and adopt using a proven teaching solution and to generate interest and awareness of the impact on critical real-world applications. [10]

The support package is an official addition to MathWorks hardware support, based on Simulink that allows design, simulate and deploy flight control algorithms on wireless PARROT minidrones via Bluetooth. This tool allows a combination of internal sensors to be used on development, simulation and test flight control algorithms. Using a PARROT minidrone and the Simulink support package is possible to obtain a low-cost laboratory to carry out feedback control system experiments and research [10].

Aerospace Blockset provides a ready-to-compile and fly example that was used to perform this work and present the results. In this chapter it will be explained how to find and open an instance of the quadcopter project using examples from the Aerospace Blockset. A brief tour on how the example uses the best practices for this application is presented, it will also show how to run a drone simulation, ways to view results, how to modify the model to customize the flight simulation configuration and an overview of the controller flight provided with the Aerospace Blockset [10].

An example can be found and opened by typing doc at the MATLAB command window. Next step is to select Aerospace Blockset, click on examples and scroll to the quadcopter project. On the documentation page, more information about this example can be found [10].

After the environment setup, there will be three new windows for monitoring: the name of the Simulink quadcopter project, the top level of the quadcopter flight simulation model, and a window for the drone that employs a Simulink 3D Animation. The Simulink project is an environment that allows file organization, general settings and user-defined tasks related to this example, and contains an instance of them that can be modified locally in the user Simulink project folder without replacing the sample files modeled in the Aerospace Blockset [10].

The model that can be seen in Fig. 8 is composed of six main blocks of subsystems that contain mathematical representations of dynamic systems. There is a mathematical representation of the dynamics, the structure of the aircraft, the sensors, the environment and the flight control system. There are also blocks used to provide input commands for the simulation and to obtain a visualization output [11].

Four of the subsystems are variant subsystems, which allow switching between various subsystem options, hence its name variant. The block called FCS for Flight Control System seen in Fig. 8, in this case, is not a variant subsystem, but is a modeled subsystem, which means that the contents of the block refer to another Simulink model. There are some other blocks at the top level of this model to set the simulation pace and another to make the simulation stop if it reaches an unwanted flight condition.

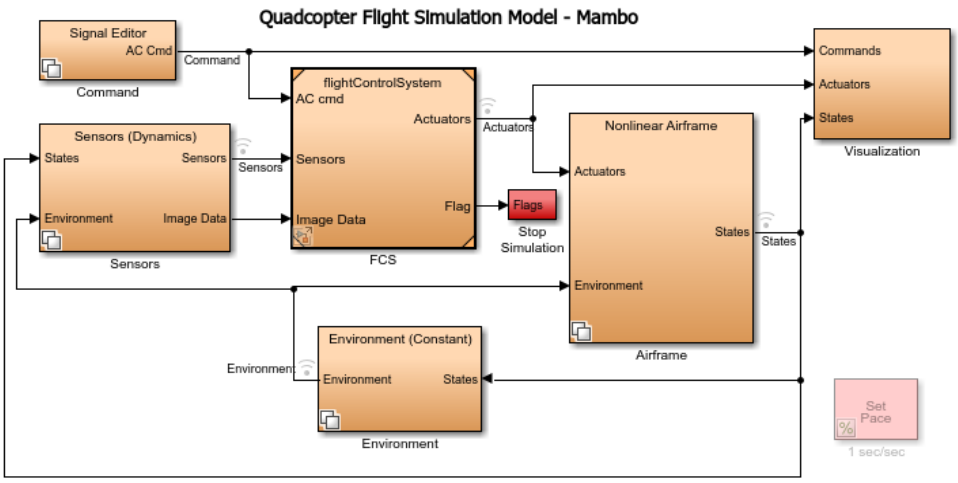

Fig. 8 Block model applied in the example of Aerospace Blockset [9]

With the simulation start, Simulink will present real-time response for the duration specified in final time parameter. The defined rhythm block can be used to change the speed at which the simulation run. For example, it is possible run just one tenth of a second of a simulation for every real second and increase the sample time to 100 μs to view a slow motion animation. During the simulation, it is possible to see the 3D model of the minidrone taking off and hovering. The screen that will open can be seen in Fig. 9 [10].

With double click on the visualization subsystem, which can be seen in Fig. 8, it is possible to find other options as state signals that are connected to the standard instrument displays available in the Aerospace Blockset. This instrument shows the measurements of aircraft variables as heading indicator, rate of climb indicator, percentage RPM indicators for each of the four thrusters, artificial horizon, altimeter and airspeed indicator. The associated signals are found in the flight instrument extraction subsystem [10].

Fig. 9 3D model of the minidrone [9]

To modify the quadcopter flight simulation parameters, a command subsystem can be used. It has four other variants that allow choosing the source of the input signals used for simulation. In the Simulink constructor approach, there is the joystick option, the data entry variant and the reading of a spreadsheet file.

The standard variant used for the control subsystem employs the signal builder. This variant allows to define the XYZ and Yaw, Pitch and Roll commands as reference signals from the flight controller. Any of these signals can be modified in the signal builder and we observe the change in the aircraft behavior in the simulation. The flight controller is found in the FCS subsystem. With a double-click, another model opens to which this block refers. This is a model subsystem and it is included in the base Simulink library. In this case, this model subsystem contains the flight control algorithms that will be implemented on the drone.

It is noteworthy that the simulation model is at the heart of model-based design and helps to test simulation projects before trying them out on real hardware. This helps refine the design of the control before to test it on the hardware and also avoid all kinds of problems during the test, including damage and failure.

In order to configure the hardware and wireless communication necessary to deploy flight controllers in the minidrones, the download of the hardware support package using the Add-On Explorer is needed, and we follow the hardware configuration steps provided. This will help install custom firmware provided by PARROT, allow Simulink users to design and integrate their own flight control algorithms into the aircraft software system. The firmware replacement takes place over a micro-USB connection.

After replacing the firmware, depending on the operating system, there are specific steps that must be taken to enable wireless communication between the host computer and the minidrone. For computers running Windows®, it is needed to activate a Bluetooth Low Energy interface with supported chipset and drivers or use a USB dongle with these features.

For each of the supported operating systems, Windows, Mac and Linux®, the support package documentation includes more detailed configuration and troubleshooting steps.

After the Bluetooth connection has been verified with a drone and before any flight test with a controller has been designed, it is worth using a model included in the support package to perform a table test in which only the propellers are moved. To do this, the model can be opened by typing PARROT getting started in the MATLAB command window and clicking on "*Deploy to Hardware*".

The diagnostic viewer shows information about code generation, compilation, file transfer and execution of the drone process. From this point, the model is ready to run on the drone. To run it, first we open the flight control user interface by clicking on the diagnostic viewer and, optionally, set the duration for which the model is to run. We should be careful not to increase the power gain on the flight control user interface at this time, as the propellers can produce enough momentum to initiate an uncontrolled flight. Then, clicking Start will execute the algorithms on the minidrone. There are no flight controller algorithms in the loop, and it is just sending control signals to the engines to ensure that the tool chain and Bluetooth are working well.

After verification, the algorithm execution can be stopped by clicking Stop. The flight control user interface allows us to download the flight log and the MAT file with flight data from the minidrone to the current directory. It is good practice to recover these files for recording and analysis.

Data stored in a MAT file can be used to plot and view sensor signals, orientation, engine output, altitude, trajectory, position, speeds, optical flow speeds and percentage of battery charge [12].

D. PID Controller

PID controller is certainly the most traditional control algorithm in the industry. This popularity is mainly due to the simplicity of adjusting parameters to obtain a good performance.

A PID controller initially calculates the error between its controlled variable and its desired value. As a result of this error, a control signal is generated to eliminate this deviation.

Equation (1) of the classic parallel PID controller position algorithm is:

$$u(t) = Kp\,.\,e(t) + Kp\,.\frac{1}{T1}\,.\int e(t)dt + Kp\,.\,Td\,.\frac{de(t)}{dt} + uo \quad (1)$$

In addition to the PID, the proportional controller (P) and the proportional and integral controller (PI) are also found in the industry.

The proportional controller generates its output in proportion to the error, with Kp as the gain of the controller. Equation (2) is the P position controller algorithm:

$$u(t) = Kp\,.\,e(t) + uo \quad (2)$$

where uo is the initial value [13].

The proportional controller is represented by the block diagram in Fig. 10.

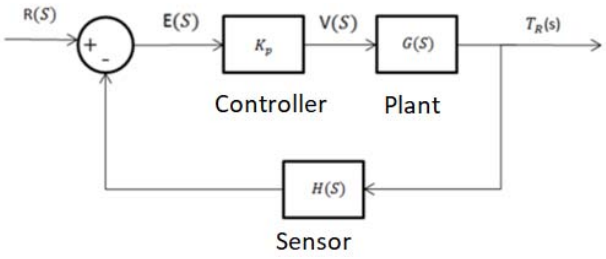

Fig. 10 Proportional Controller [13]

The integral proportional controller (PI) generates its output proportional to the error P and proportional to the integral of the error. Equation (3) is the PI controller position algorithm:

$$u(t) = Kp\,.\,e(t) + Kp\,.\frac{1}{T1}\,.\int e(t)dt + u \quad (3)$$

where 1 / t1 is the integral gain [12].

E. Controller Design

Using the classical controller model as a reference, some formulas necessary to find the Newton-Euler angle value will be shown below [14].

The equation in the frequency domain can be shown in (4):

$$U(S) = \left(Kp + \frac{Ki}{S} + KdS\right)E(S) \quad (4)$$

For a closed loop system, (5)-(7) are shown:

$$Am = \frac{Km^2}{Rm(Jm+Jr)} + \frac{d\Omega i}{Jm+Jr} \quad (5)$$

$$Bm = \frac{Km}{Rm(Jm+Jr)} \quad (6)$$

$$\Omega\cdot i = -Am\Omega i + BmV \quad (7)$$

where Km is the electromotive force constant, Rm is the internal resistance of the motor, Jm is the motor inertia and Am and Bm are terms of the motor equation. For a quadcopter, (8)-(10) are found in the Laplace domain [15]:

$$\emptyset(S) = \frac{Bm^2L}{S^2(S+Am)^2Ixx}U2(S) \quad (8)$$

$$\theta(S) = \frac{Bm^2L}{S^2(S+Am)^2Iyy}U3(S) \quad (9)$$

$$\psi(S) = \frac{Bm^2L}{S^2(S+Am)^2Izz}U4(S) \quad (10)$$

where U2, U3 and U4 are given by the PID control, Ixx, Iyy and Izz are the moments of inertia for each axis. The closed-loop equation to stabilize in flight hovering at the roll angle is given by replacing (4) in (8) as shown in (11):

$$\emptyset(S) = \frac{[S^2KpKdBm^2l+SBm^2l[KpTf(Kp+Kd)+KiKd]+KiKpTfBm^2l]E(S)}{S^6KdIxx+S^5Ixx(KpTf+2KdAm)+S^4AmIxx(2KpTf+KdAm)+S^3KpTfAm^2Ixx} \quad (11)$$

For pitch and yaw angles, the equation is found in the same way as (11) [15].

IV. Simulation and Comparisons

Using the mathematical model shown in the previous section, it is possible to find the value of Newton-Euler angles, so that it is possible to stabilize the quadcopter in a certain position hovering in the air, starting from the initial position that will be the ground. With these values, it is possible to change the angles in the block diagram of Fig. 8 and test the values by means of simulation.

The example adopts the values of standard controllers such as, takeoff controller gain equal to 0.45, relative maximum thrust controller gain of 0.92 and maximum thrust controller gain of 0.3266.

In order to understand the impact of the controller gain values on the stabilization of the Euler-Newton angles, 3 conditions will be tested in the simulations with different controller gain combinations as shown in Table I.

TABLE I
Earnings Comparison

	Condition 1	Condition 2	Condition 3
Takeoff Gain	0.35	0.45	0.55
Relative Maximum Thrust Gain	0.82	0.92	1.02
Maximum Thrust Gain	0.2266	0.3266	0.4266

Where condition 2 shows the standard earnings values suggested by the simulator, condition 1 is gained with 0.10 unless condition 2 and condition 3 are earned with 0.10 more than condition 2.

A. Euler Angle Results

With reference to the value of the angles at the origin and Table I, Figs. 11-13 can be observed, showing the pitch, roll and yaw angles being compared, respectively for conditions 2, 1 and 3.

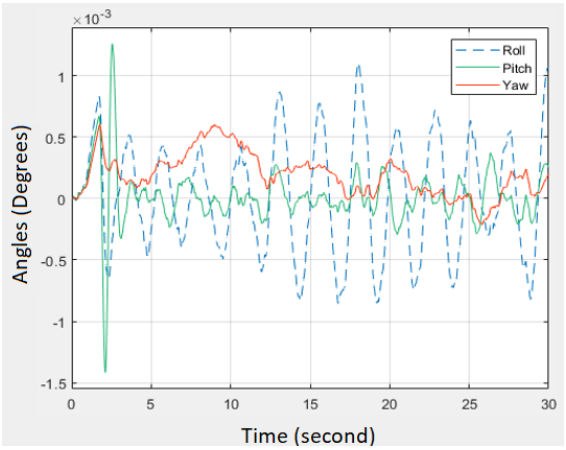

Fig. 11 Comparison of Newton-Euler angles (Condition 2)

The results observed according to the condition 2 gains suggested that by the simulator, the Roll angle varied a lot if compared to the other angles. The Yaw angle varied with the same intensity during the 30 seconds. The pitch angle started with a very high variation and decreased over time. Therefore, Pitch, even starting with a very high variation, was the one that had the best stability over time, it was closer to the 0 ° angle that was desired.

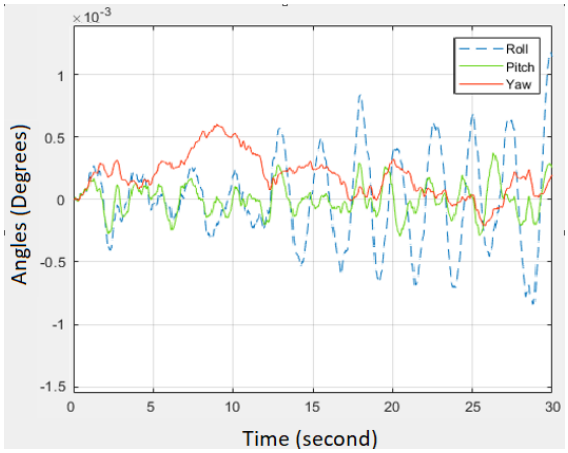

Fig. 12 Comparison of Newton-Euler angles (Condition 3)

The results observed in terms of condition 3 show that the Roll angle started with small variations and got bigger over time. The Yaw angle started with a high variation and became more stabilized over time. The Pitch angle maintained a pattern over time, varying little compared to the other angles. Therefore, the pitch was the one that had the best stability over time, it was closer to the 0 ° angle that was desired.

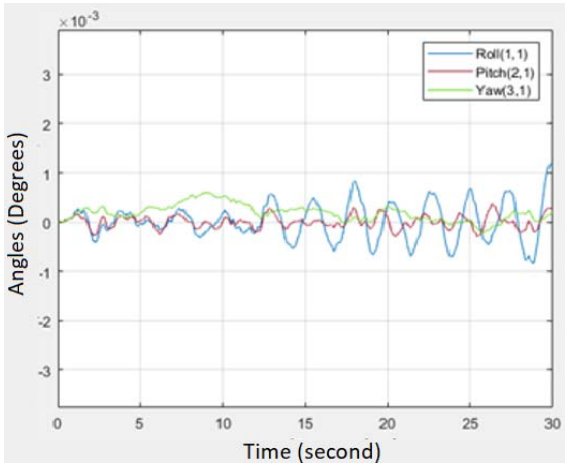

Fig. 13 Comparison of Newton-Euler angles (Condition 1)

The results observed in terms of condition 1 gains show that the Roll angle started varying little and in the end it had minor variations. The Yaw angle varied with the same intensity during the 30 seconds. The Pitch angle kept its variations in the same way until the end of the 30 seconds. Therefore, the

Pitch was the one that had the best stability over time, it was closer to the 0 ° angle that was the desired one.

To verify the efficiency of the Parrot's stability, several values of drone controllers were simulated, having the angles at zero as reference, changing the behavior of the structure in several ways, in order to also verify the time it takes the controller to stabilize the vehicle, knowing that the desired value of the angles would be equal to 0 °.

By analyzing the graphs, it can be observed that the noise of the sensors becomes much smaller when the gains are smaller. This is because buoyancy is a force that arises when a body occupies space within a fluid, and in some cases, it may be equivalent to the value of the body's mass, this force impacts on the stability of drones, however, when the buoyancy gain is lower and takeoff gain, better drone flight, making it stabilize faster [15].

The position, speed and acceleration of each of the Newton - Euler angles will be compared in the topics below, so that the behavior in relation to each condition in Table I is observed.

B. Yaw Control Results

In this section, the behavior of the simulations in relation to the Yaw angle will be observed.

Fig. 14 shows that the behavior of the position of Yaw, given the set point of -1.5 meters.

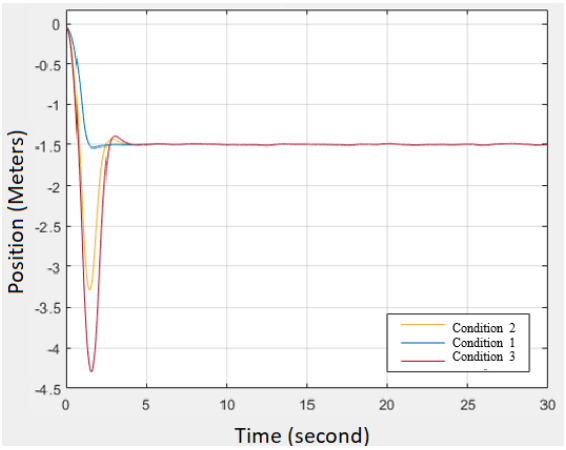

Fig. 14 Comparison of the position in relation to the Yaw

It is observed that the position in Yaw had its greatest variation in condition 3, this because its gains are greater, causing the noises to increase. Condition 2, on the other hand, also had a high variation at the beginning, but not as high as condition 3. Condition 1 performed better, because the gains are smaller, making the variation less.

In the next simulation shown in Fig. 15, the graph of the speed comparison in relation to each condition can be seen and the behavior of the same in relation to time can be observed.

It is observed that the speed in Yaw had its greatest variation in condition 3, this is because its gains are smaller, causing the noise to increase. Condition 2, on the other hand, also had a high variation at the beginning, but not as high as condition 3. Condition 1 performed better, because the gains are smaller, making the variation less.

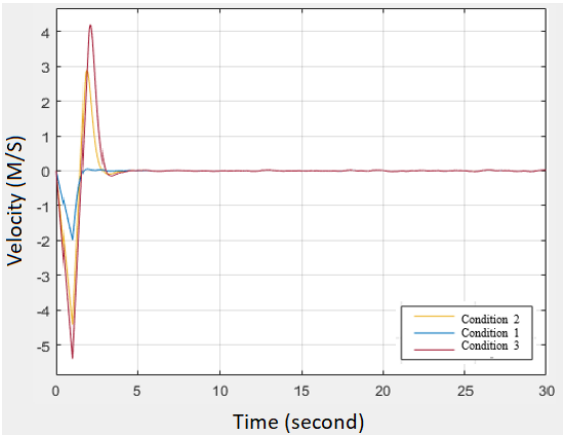

Fig. 15 Graph of velocity in relation to Yaw

Finally, the simulation of Fig. 16 was made, with the acceleration in relation to each of the different conditions of Table I, in which it is possible to observe the acceleration peaks that occurred during the drone's flight.

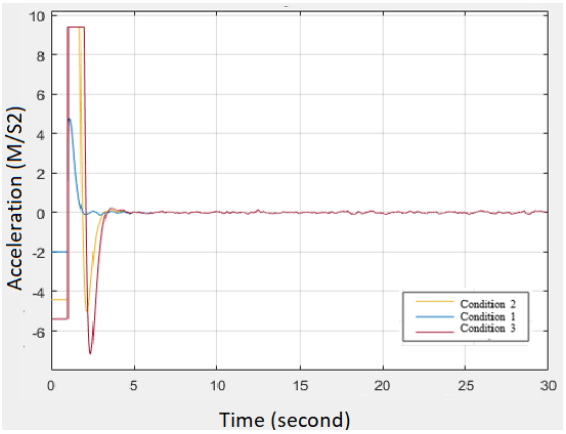

Fig. 16 Graph of acceleration in relation to Yaw

It is observed that the Yaw acceleration had saturation in conditions 2 and 3. This is because the simulator does not support values of gains greater than or equal to the standard (condition 2) suggested by the simulator, when submitted to a higher than acceptable value. The acceleration tends to saturate, having forced it to use its maximum strength. Condition 1 performed better, because the gains are smaller, making the variation less.

Note that in the previous three simulations, the comparison of condition 1 behaves better than the other conditions, this again proves that the lower the thrust gain and takeoff gain, the better the drone's flight, making it stabilize faster.

C. Roll Control Results

In this section, simulations performed in relation to Roll will be observed.

Fig. 17 shows the behavior of the Roll position with a set point of 0. It is observed that the position in Roll had its greatest variation in condition 2. Condition 3 also had a high variation in the beginning, but not as high as in condition 2. Condition 1 had better performance, because the gains are smaller, making the variation less. In the next simulation in Fig. 18, the graph of the comparison of speeds in relation to each condition can be seen and the behavior of the same in relation to time can be observed.

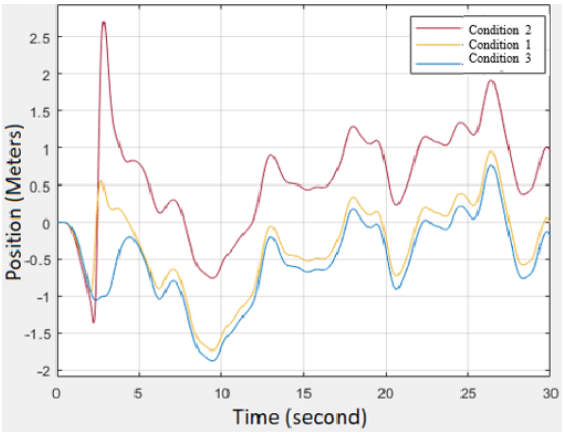

Fig. 17 Comparison of the position in relation to the Roll

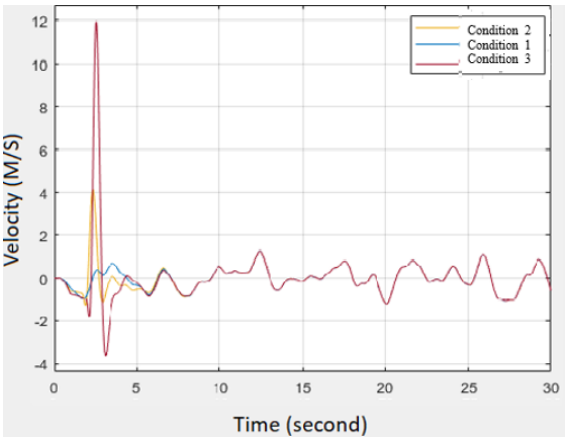

Fig. 18 Graph of velocity in relation to the Roll

It is observed that the speed in Roll had its greatest variation in condition 3. Condition 2 also had an average variation in the beginning when compared to condition 3 and 1. Condition 1 had better performance, because the gains are smaller, making with less variation.

Finally, the simulation of Fig. 19 was made, with the acceleration in relation to each condition in Table I, in which it is possible to observe the acceleration peaks that occurred during the drone's flight.

It is observed that the acceleration in Roll had its greatest variation in condition 3. Condition 2 also had an average variation at the beginning when compared to condition 3 and 1. Condition 1 had better performance, because the gains are smaller, causing the variation to be smaller.

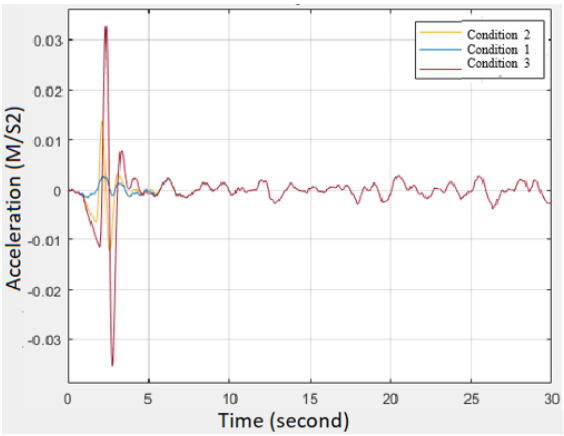

Fig. 19 Graph of acceleration in relation to the Roll

D. Pitch Control Results

In this section, it is possible to observe the variations to the pitch angle.

Fig. 20 shows the behavior of the Pitch position with set point at 0.

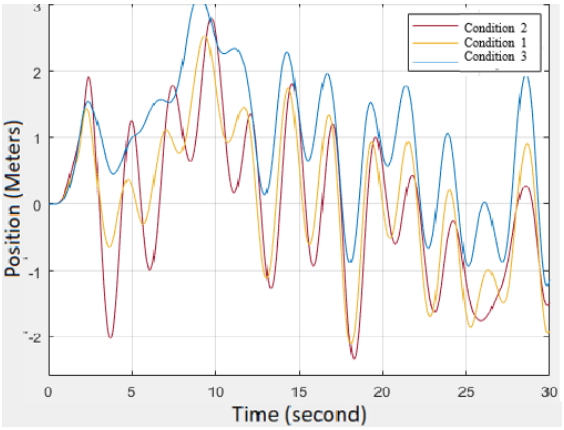

Fig. 20 Comparison of the position in relation to the pitch

It is observed that the position in Pitch had its greatest variation in condition 3. Condition 2 also had a high variation, but not as high as condition 3. Condition 1 performed better, this because the gains are smaller, making with less variation.

In the next simulation shown in Fig. 21, the graph of the comparison of speeds in relation to each condition can be seen.

It is observed that Pitch speed had its greatest variation in condition 3. Condition 2 also had a high variation, but not as high as condition 3. Condition 1 had better performance, because the gains are lower, making with less variation.

Finally, the simulation of Fig. 22 was made, with the acceleration in relation to the condition in Table I, in which it is possible to observe the acceleration peaks that occurred during the drone flight.

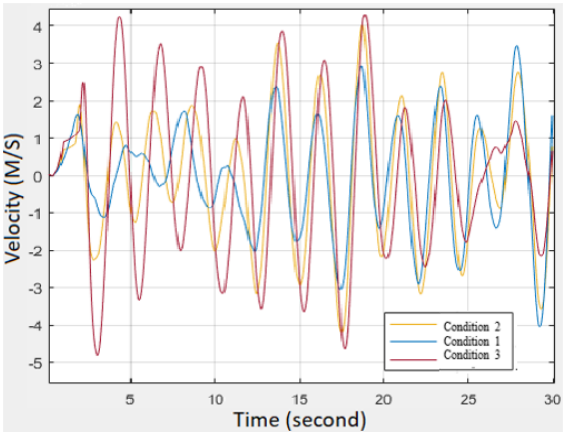

Fig. 21 Graph of velocity in relation to pitch

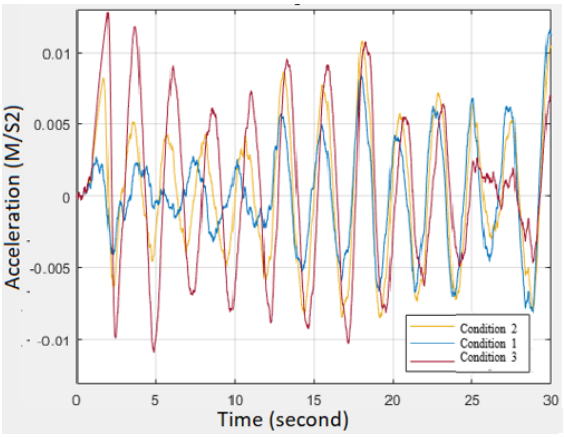

Fig. 22 Graph of acceleration in relation to the pitch

It is observed that Pitch acceleration had its greatest variation in condition 3. Condition 2 also had a high variation, but not as high as condition 3. Condition 1 had better performance, because the gains are smaller, making with less variation.

As can be seen, the pitch angle is the one that had the most difficulty in stabilizing when compared to the other angles. This is because the pitch of the pitch varied a lot during the flight; this causes its stability to be severely impaired by the noises.

E. Comparison of Results

The simulation tests show that the Yaw angle starts the simulation as desired but due to the variation in the Roll and Pitch angle it varies along the time, and only after the other angles stabilize it stabilizes as well. It can also be seen that condition 1 in all simulations stabilized very quickly and with little noise. This is because, as we have seen before, the lower the thrust gain and takeoff gain, the better the drone's flight, making it stabilize faster.

V. Conclusion and Future Work

It is concluded that the objective was successfully accomplished, several difficulties were faced, mainly when finding information about the simulator and how it worked.

When analyzing the results obtained in the stabilization of the quadcopter through the Newton – Euler angles, it was verified that there are many oscillations regarding the efficiency of the sensors, whether it is calibrated or not, if it has been correctly stabilized, no flight is completely perfect, but if stabilized with smaller gains have already observed that the noise is less.

The simulator shows what a PARROT drone would look like in real life, precisely so that errors are readjusted and do not damage the drone in practice.

The simulator has several resources that are very useful mainly for students and professors at universities, because it allows for several resources without needing much effort, precisely because it is so interesting and has so many resources, that for future work it is intended to deepen the studies on the simulator and tests will also be carried out in practice so that the accuracy level of the simulator and the stability of the drone are observed.

References

[1] Tuton Chandra Mallick, Mohammad Ariful Islam Bhuyan, Mohammed Saifuddin Munna. Design and implementation of a UVA (Drone) with flight data recording. Available: “https://ieeexplore.ieee.org/document/7856519” Oct, 2016.

[2] Jun Xiao, Weiwei Zhang, Yujie Han, Min Sun. Influencing factor and corresponding method of target drone dynamic stability. Available: “https://ieeexplore.ieee.org/document/5988138”. Jul, 2011.

[3] Nappaphol Siriphun, Shigeru Kashihara, Doudou Fall, Assadarat Khurat. Distinguishing Drone Types Based on Acoustic Wave by IoT Device. Available: “https://ieeexplore.ieee.org/document/8712755“. May, 2019.

[4] Seyit Alperen Celtek, Akif Durdu, Ender Kurnaz. Design and Simulation of the Hierarchical Tree Topology Based Wireless Drone Networks. Available: “https://ieeexplore.ieee.org/document/8620755”. Jan 2019.

[5] Majed Alwateer, Seng W. Loke, Niroshinie Fernando. Enabling Drone Services: Drone Crowdsourcing and Drone Scripting. Available: “https://ieeexplore.ieee.org/document/8788511”. Aug, 2019.

[6] Saeed Pirbodaghi, Dinesh Thangarajan, Teo Hung Liang, Madhavan Shanmugavel, Veera Ragavan, Joao Silva. Cooperative heterogeneous Unmanned Autonomous Systems solution for monitoring and inspecting power distribution system. Available: “https://ieeexplore.ieee.org/document/7475330”. Dec, 2015.

[7] Anupama Vijay , V. R. Jisha, Robin Emmanuel. Control and Development of Coaxial Octocopter for Human Transportation in Gazebo. Available: https://ieeexplore.ieee.org/document/9029020. Dec, 2019.

[8] Parrot. Mambo Fly Parrot. Available: “https://www.parrot.com/eu/drones/parrot-mambo-fly". 2020.

[9] Mathworks. Quadcopter Project. Available: "https://www.mathworks.com/help/aeroblks/quadcopter-project.html". Dec, 2019.

[10] Drayer Andrade, Greg *Drones with Simulink*. Available: “https://www.mathworks.com/videos/programming-drones-with-simulink-1513024653640.html.” . Feb, 2020.

[11] Drayer Andrade, Greg. Hardware Support. Available: “https://www.mathworks.com/hardware-support/parrot-minidrones.html". Dec, 2019.

[12] Asma Katiar, Rabbia Rashdi, Zeeshan Ali and Urooj Baig, Mario Cesar M., Teixeira, Herbert CG. Quadcopter control and stability analysis. Available: “https://ieeexplore.ieee.org/document/8346419/authors#authors”. Apr 2018.

[13] Nóbrega Velosa, Carlos Miguel. Control of the Position and Attitude of a Quadcopter by Programmable References. Available:

"https://ubibliorum.ubi.pt/bitstream/10400.6/3642/1/TESE%20VERS%C3%83O%20FINAL%20%2024%20Junho%202011.pdf". June, 2011.
[14] Cavalcante de Sá, Rejane. Construction, Dynamic Modeling and PID Control for the Stability of a Quadrotor Unmanned Aerial Vehicle. Available:"http://repositorio.ufc.br/bitstream/riufc/4097/1/2012_dis_rcsa.pdf. " 2012.
[15] Espiúca Monteiro, João Carlos. Modeling and Control of a Quadrotor Vehicle. Available: "http://monografia.poli.ufrj.br/monógrafo/monopoli10014928.pdf". March, 2015.