



ESCUELA POLITÉCNICA NACIONAL



FACULTAD DE INGENIERÍA MECÁNICA

DISEÑO Y SIMULACIÓN DE UN ESTABILIZADOR DE CÁMARA CONTROLADO POR VISIÓN COMPUTACIONAL PARA UN VEHÍCULO AÉREO NO TRIPULADO DE ALA FIJA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO MECÁNICO

ARIEL PATRICIO HERRERA AGUIAR

ariel.herrera@epn.edu.ec

DIRECTOR: ING. VALENCIA TORRES ESTEBAN ALEJANDRO, PhD.

esteban.valencia@epn.edu.ec

CODIRECTOR: ING. VANDEWALLE PATRICK LUC ELS, PhD.

patrick.vandewalle@kuleuven.be

Quito, Mayo 2022

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por ARIEL PATRICIO HERRERA AGUIAR bajo nuestra supervisión.

ING. VALENCIA TORRES ESTEBAN ALEJANDRO, PhD

DIRECTOR DEL PROYECTO

ING. VANDEWALLE PATRICK LUC ELS, PhD

CODIRECTOR DEL PROYECTO

DECLARACIÓN

Yo, **Ariel Patricio Herrera Aguiar**, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

Ariel Patricio Herrera Aguiar

DEDICATORIA

Dedico este trabajo a mis padres y abuelo por su amor y apoyo incondicional.

AGRADECIMIENTO

Siempre he creído que los pilares fundamentales de mi vida para haber cumplido mis metas han sido mi familia, profesores y amigos.

Soy muy afortunado de tener una familia que me ha inculcado buenos valores. Agradezco principalmente a mis padres, Lucila Aguiar y Patricio Herrera, a mis hermanos, Elián Herrera y Axel Herrera, y finalmente a mi querido abuelito Víctor Aguiar. Los amo mucho.

Agradezco a mis profesores que me han formado en las aulas de la Facultad de Ingeniería Mecánica de la Escuela Politécnica Nacional, principalmente a Esteban Valencia, mi director de tesis, por haber depositado su confianza en mí y permitir que varios proyectos de investigación innovadores se lleven a cabo en el Ecuador, a Edgar Cando por ser un profesor que me apoyó en varios momentos de mi vida estudiantil, a Víctor Alulema por ser un mentor del cual aprendí mucho para ser un gran investigador.

I thank Patrick Vandewalle for the support I received during my stay at KU Leuven and Xueyang Kang for his friendship and advice that I will apply throughout my professional life.

Agradezco a mis amigos de toda la vida Xylon Funes y Carlos Chávez, los cuales siempre me han brindado consejos de vida y me han colaborado en todo reto que se me ha presentado. De la Facultad me llevo grandes amistades, conocí a personas fantásticas con un gran amor por la ciencia y tecnología, principalmente quiero agradecer a César Guaila, Micaela Pazmiño, Mónica Jácome, Juan López, Sergio Torres, Byron Ayala, Danilo Pillajo, Steven Ramos, Carlos Porras, Eduardo Toapanta, Jorge Bravo, Benito Ribadeneira y Esteban Cúñez.

A mi querida sección ASME EPN por tantas aventuras, concursos, charlas técnicas, y que cada año nos sigan sorprendiendo, dejando en alto el nombre de la Facultad.

Finalmente, mi formación académica no fue solo técnica, sino que también mi pasión por la música me abrió las puertas para unirme al Coro Mixto Politécnico. Agradezco a Ramiro Analuisa y a cada integrante del grupo por su inmenso cariño, confianza y por tantos buenos momentos que pasamos. Nunca dejen de cantar.

CONTENTS

CERTIFICACIÓN.....	i
DECLARACIÓN.....	ii
DEDICATORIA.....	iii
AGRADECIMIENTO.....	iv
LIST OF FIGURES.....	viii
LIST OF TABLES.....	xi
ABSTRACT.....	xiii
RESUMEN.....	xiv
GLOSSARY OF TERMS.....	xv
INTRODUCTION.....	1
General Objective.....	2
Specific Objectives.....	2
Scope.....	2
1. BACKGROUND / LITERATURE REVIEW.....	3
1.1. Unmanned Aerial Vehicles.....	3
1.1.1. Multi-rotor UAV.....	3
1.1.2. Fixed-wing UAV.....	3
1.2. Camera gimbals.....	4
1.2.1. Components of a gimbal for a UAV.....	5
1.2.2. Types of gimbals (Benchmarking).....	10
1.3. Computer vision and OpenCV.....	14
1.3.1. Feature detection.....	15
1.3.2. Curve fitting algorithm.....	19
1.3.3. Machine Learning.....	20
2. METHODOLOGY.....	26
2.1. Gimbal Definition.....	26
2.1.2. QFD Matrix.....	27

2.1.3. Gimbal Technical Specifications	28
2.2. Conceptual Design	29
2.2.1. Functional analysis of the gimbal	29
2.2.2. Module Definition	30
2.2.3. Find and evaluate solution principles	31
2.2.3.1. Control module solution principles and evaluation	31
2.2.3.2. Structural module solution principles and evaluation	37
2.2.3.3. Morphological diagram of the solution for the gimbal	42
2.3. Materialization Design	43
2.3.1. Computer Vision algorithm programming	43
2.3.1.1. Sky line detection algorithm	45
2.3.1.2. Sky line coordinates storage	49
2.3.1.3. Linear regression algorithm	49
2.3.1.4. Sky line tracking	50
2.3.2. Hardware component selection	52
2.3.2.1. Servo motor selection	52
2.3.2.2. Power supply sizing	54
2.3.3. Structural module dimensioning	55
2.3.3.1. Materials selection	55
2.3.3.2. Bullet aerodynamic profile dimensioning	57
2.3.3.3. Mechanism dimensioning	60
2.3.3.4. Anti-vibration plate bolt dimensioning	65
2.3.3.5. Protective case dimensioning	67

2.4. Detail Design	68
3. RESULTS AND DISCUSSION	69
3.1. U-net Segmentation Neural Network results	69
3.2. Algorithm results and optimization	70
3.2.1. Boundary edge removal	70
3.2.2. Sky line motion prediction.....	71
3.3. Analysis of the final design of the gimbal	74
3.3.1. Gimbal project costs	75
3.3.2. Vision Range	76
4. CONCLUSIONS AND FUTURE WORK	77
4.1. Conclusions.....	77
4.2. Future work	77
BIBLIOGRAPHY	79
APPENDICES.....	86
APPENDIX I. Sift, Surf and Orb feature detection code	86
APPENDIX II. Canny Edge Detection code.....	87
APPENDIX III. U-Net Image Segmentation CNN Data Loading Code	89
APPENDIX IV. Sky line motion compensation algorithm.....	91
APPENDIX V. Mechanical properties of steel and aluminum	95
APPENDIX VI. Metric Threads.....	95
APPENDIX VII. Wind Atlas of Ecuador	96
APPENDIX VIII. Stress-Concentration Factors K_t	97
APPENDIX IX. Coefficient for rectangular bars in torsion.....	97
APPENDIX X. ABS Plastic Mechanical Properties.....	98
APPENDIX XI. Beam fixed at both ends equations	98
APPENDIX XII. Assembly and components drawings.....	99

LIST OF FIGURES

FIGURE 1.1. MULTI-ROTOR UAV	3
FIGURE 1.2. FIXED-WING UAV.....	4
FIGURE 1.3. FIXED-WING AIRCRAFT ROTATIONS.....	4
FIGURE 1.4. EO CAMERA GIMBAL.....	5
FIGURE 1.5. DIGITAL EAGLE CAMERA GIMBAL PARTS.....	5
FIGURE 1.6. ROTATION FOR ROLL, PITCH, AND YAW ANGLES.....	6
FIGURE 1.7. IMU SENSOR.....	7
FIGURE 1.8. DRONE GIMBAL CONTROL UNIT.....	8
FIGURE 1.9. BRUSHLESS MOTOR.....	8
FIGURE 1.10. UAV BATTERY.....	8
FIGURE 1.11. TAROT GOPRO CAMERA GIMBAL.....	9
FIGURE 1.12. HOMOGRAPHY MATRIX BETWEEN A WORLD PLANE AND AN IMAGE PLANE.....	9
FIGURE 1.13. RGB CUBE AND HSV CYLINDER.	15
FIGURE 1.14. HOMOGRAPHY MATRIX BETWEEN TWO IMAGE PLANES.	16
FIGURE 1.15. SIFT AND IMAGE STITCHING ALGORITHMS.	16
FIGURE 1.16. KERNEL EXAMPLE.	17
FIGURE 1.17. GAUSSIAN BLURRING.....	18
FIGURE 1.18. CANNY EDGE DETECTION ALGORITHM.....	19
FIGURE 1.19. CURVE FITTING.	19
FIGURE 1.20. ANN STRUCTURE.....	21
FIGURE 1.21. LAYERS OF A SUPERVISED LEARNING ARTIFICIAL NEURAL NETWORK.	21
FIGURE 1.22. SIGMOID AND RELU ACTIVATION FUNCTIONS.	22
FIGURE 1.23. LEARNING RATE CURVES.	24
FIGURE 1.24. LOSS AND ACCURACY CURVES OF A LEARNING NEURAL NETWORK.	24
FIGURE 1.25. CNN EXAMPLE.....	25
FIGURE 1.26. U-NET IMAGE SEGMENTATION.....	25
FIGURE 1.27. U-NET ARCHITECTURE.....	25
FIGURE 2.1. METHODOLOGY.	26
FIGURE 2.2. QFD MATRIX.	28
FIGURE 2.3. LEVEL 0 OF THE FUNCTIONAL STRUCTURE.	29
FIGURE 2.4. LEVEL 1 OF THE FUNCTIONAL STRUCTURE.	30

FIGURE 2.5. LEVEL 2 OF THE FUNCTIONAL STRUCTURE.	30
FIGURE 2.6. MODULAR STRUCTURE.	31
FIGURE 2.7. 9-DOF ABSOLUTE ORIENTATION IMU FUSION BREAKOUT.	32
FIGURE 2.8. OPENCN1.0 CONTROLLER.	33
FIGURE 2.9. HARDWARE SETTING SCHEMATIC.	37
FIGURE 2.10. CASE.	38
FIGURE 2.11. GIMBAL ANTI-VIBRATION PLATE.	39
FIGURE 2.12. STRUCTURAL MODULE SCHEMATIC.	42
FIGURE 2.13. GIMBAL CONCEPT MODEL.	43
FIGURE 2.14. COMPUTER VISION ALGORITHM METHODOLOGY.	44
FIGURE 2.15. INPUT AND DESIRED OUTPUT OF THE STABILIZATION ALGORITHM.	44
FIGURE 2.16. SIFT, SURF, AND ORB FEATURES.	45
FIGURE 2.17. CANNY EDGE DETECTION LIBRARY.	46
FIGURE 2.18. CANNY EDGE DETECTION LIBRARY USING HSV IMAGES.	47
FIGURE 2.19. SKYFINDER DATASET.	48
FIGURE 2.20. TRAINING RESULTS.	48
FIGURE 2.21. BINARY SEGMENTATION PREDICTION RESULT.	49
FIGURE 2.22. SKY LINE COORDINATES STORAGE.	49
FIGURE 2.23. STRAIGHT SKY LINE RESULT.	50
FIGURE 2.24. REFERENCE SKY LINE RESULT.	50
FIGURE 2.25. COMPENSATED IMAGE RESULT.	52
FIGURE 2.26. SERVO MOTOR TORQUE.	54
FIGURE 2.27. BULLET AERODYNAMIC PROFILE PARTS.	57
FIGURE 2.28. BULLET AERODYNAMIC PROFILE SHEAR STRESS ANALYSIS.	57
FIGURE 2.29. SHAPE EFFECTS ON DRAG.	59
FIGURE 2.30. STATIC ANALYSIS OF LINK 2 OF THE MECHANISM.	60
FIGURE 2.31. ABS PLASTIC STATIC ANALYSIS.	63
FIGURE 2.32. BOLTS STATIC ANALYSIS.	65
FIGURE 2.33. PROTECTIVE CASE STATIC ANALYSIS.	67
FIGURE 3.1. LEARNING RATE CURVE.	69
FIGURE 3.2. VALIDATION ACCURACY CURVE.	70
FIGURE 3.3. TRAIN LOSS CURVE.	70
FIGURE 3.4. BOUNDARY REMOVAL FUNCTION.	71
FIGURE 3.5. FIRST AND SECOND FRAME SKY LINE.	71

FIGURE 3.6. IDENTIFYING THE CORRECT SKY LINE.	73
FIGURE 3.7. SKY LINE AND MOTION PREDICTION LINE OF THE THIRD FRAME.	74
FIGURE 3.8. APPROXIMATE WEIGHT OF 3D PRINTED GIMBAL MECHANICAL PARTS.....	75
FIGURE 3.9. LOCATION OF THE GIMBAL ON THE UAV.	76
FIGURE 3.10. ROTATION ANGLE ANALYSIS.....	76

LIST OF TABLES

TABLE 1.1. TECHNICAL SPECIFICATIONS OF THE TAROT ZYX T2D-10X GIMBAL.	11
TABLE 1.2. TECHNICAL SPECIFICATIONS OF THE SKY EYE-36SZ 1080P 36X GIMBAL.....	12
TABLE 1.3. TECHNICAL SPECIFICATIONS OF THE DJI ZENMUSE X5S GIMBAL..	13
TABLE 2.1. USER'S VOICE AND ENGINEER'S VOICE.....	27
TABLE 2.2. GIMBAL TECHNICAL SPECIFICATIONS.....	29
TABLE 2.3. CORRECTED WEIGHTED CRITERIA ORDINAL METHOD.....	31
TABLE 2.4. SOLUTION ALTERNATIVES FOR THE CONTROL MODULE.	32
TABLE 2.5. SOLUTIONS TO "TAKE INPUT IMAGES".	33
TABLE 2.6. SOLUTIONS TO "PROCESS RECEIVED DATA".	34
TABLE 2.7. SOLUTIONS TO "CONVERT ELECTRICAL ENERGY INTO TORQUE".	34
TABLE 2.8. EVALUATION OF THE SPECIFIC WEIGHT OF EACH CRITERION.....	35
TABLE 2.9. EVALUATION OF THE SPECIFIC WEIGHT OF THE "HIGH PROCESSING CAPACITY" CRITERION.	36
TABLE 2.10. EVALUATION OF THE SPECIFIC WEIGHT OF THE "FRIENDLY ENVIRONMENT" CRITERION.....	36
TABLE 2.11. EVALUATION OF THE SPECIFIC WEIGHT OF THE "AVOID OVERHEATING" CRITERION.	36
TABLE 2.12. EVALUATION OF THE SPECIFIC WEIGHT OF THE "COMPACT" CRITERION.	36
TABLE 2.13. EVALUATION OF CONCLUSIONS OF THE CONTROL MODULE.	37
TABLE 2.14. SOLUTION ALTERNATIVES FOR THE STRUCTURAL MODULE.....	37
TABLE 2.15. SOLUTIONS TO "HOLD AND PROTECT CAMERA AND IMU".....	38
TABLE 2.16. SOLUTIONS TO "MOVE CAMERA".	39
TABLE 2.17. EVALUATION OF THE SPECIFIC WEIGHT OF EACH CRITERION....	40
TABLE 2.18. EVALUATION OF THE SPECIFIC WEIGHT OF THE "WIDE ANGLE ROTATION RANGE" CRITERION.	41
TABLE 2.19. EVALUATION OF THE SPECIFIC WEIGHT OF THE "SAFE" CRITERION.	41
TABLE 2.20. EVALUATION OF THE SPECIFIC WEIGHT OF THE "LOW WEIGHT" CRITERION.....	41
TABLE 2.21. EVALUATION OF THE SPECIFIC WEIGHT OF THE "EASY MANUFACTURING" CRITERION.....	41
TABLE 2.22. EVALUATION OF CONCLUSIONS OF THE STRUCTURAL MODULE.	42
TABLE 2.23. SELECTED SOLUTIONS FOR EACH GIMBAL MODULE.....	42

TABLE 2.24. TECHNICAL SPECIFICATIONS OF HARDWARE COMPONENTS	52
TABLE 2.25. TECHNICAL SPECIFICATIONS OF THE DYNAMIXEL XL SERIES SERVO MOTORS.	53
TABLE 2.26. POWER OF EACH HARDWARE COMPONENT.....	55
TABLE 2.27. CHARACTERISTICS SCORE.	56
TABLE 2.28. MECHANICAL PROPERTIES OF GIMBAL MATERIALS.	56
TABLE 2.29. MATERIAL SELECTION RESULTS.	56
TABLE 2.30. TECHNICAL SPECIFICATIONS OF THE BALL DAMPER.	66
TABLE 3.1. GIMBAL PROJECT COSTS	75

ABSTRACT

The use of Unmanned Aerial Vehicles (UAVs) currently allows environmental monitoring applications. The use of a camera incorporated in a UAV is essential to carry out this type of tasks, however, due to vibrations of the UAV and the forces of the wind, it is necessary to incorporate a gimbal to stabilize the camera to obtain excellent quality of images. The ATA research group of Escuela Politécnica Nacional together with the EAVISE research group of KU Leuven are currently leading projects related to the monitoring of volcanic activity in Ecuador. Thus, this thesis project focuses on designing a gimbal prototype that allows stabilizing a camera that will be placed on a fixed-wing UAV of the ATA research group.

The control of a gimbal is commonly based on an inertial measurement unit (IMU) sensor that detects the movement of the mechanical system and stabilizes it using motors. The gimbal design of this thesis is complemented with a stabilization control by open-source Computer Vision algorithms that will allow future researchers to adapt it to not only stabilize a camera but also to perform more applications. The algorithm takes outdoor images as input data and detects the sky line that is aligned with a reference image by controlling the roll rotation angle, which is the angle of rotation that presents the greatest instability during a flight test. This algorithm is simulated for images captured by a GoPro camera placed on a fixed-wing UAV belonging to the ATA research group during a flight test carried out in the Antisana Ecological Reserve.

On the other hand, the gimbal design is based on a 3-link mechanism that allows a Raspberry camera to rotate in roll and pitch using two servo motors that will be controlled by an NVIDIA Jetson Nano board. The structure of the gimbal is designed for a bullet aerodynamic profile with a low drag force does not affect the flight of the UAV. The design also seeks simple manufacturing and low weight to accommodate the maximum payload of the UAV.

Keywords: Gimbal, Computer Vision, algorithm, sky line, UAV, NVIDIA Jetson Nano, roll rotation angle.

RESUMEN

El uso de Vehículos Aéreos no Tripulados (UAVs) en la actualidad permite realizar aplicaciones de monitoreo ambiental. El uso de una cámara incorporada en un UAV es esencial para realizar este tipo de tareas, sin embargo, debido a vibraciones del UAV y a las fuerzas del viento es necesario añadir un gimbal para que la misma pueda estabilizarse y obtener una excelente calidad de imágenes. El grupo de investigación ATA de la Escuela Politécnica Nacional junto con el grupo de investigación EAVISE de KU Leuven lideran actualmente proyectos relacionados con el monitoreo de actividad volcánica en Ecuador. Es así que este proyecto de tesis se enfoca en diseñar un prototipo de gimbal que permita estabilizar una cámara que será colocada en un UAV de ala fija del grupo de investigación ATA.

El control de un gimbal comúnmente se basa en un sensor de unidad de medida inercial (IMU) que detecta el movimiento del sistema mecánico y lo estabiliza mediante motores. El diseño del gimbal de esta tesis es complementado con un control de estabilización por algoritmos de Visión Computacional de código abierto que permitirá a futuros investigadores adaptarlo para no solo estabilizar una cámara sino realizar más aplicaciones. El algoritmo toma como datos de entrada imágenes al aire libre y detecta la línea del horizonte que es alineada con una imagen de referencia controlando el ángulo de rotación de alabeo que es el ángulo de giro que presenta mayor inestabilidad durante una prueba de vuelo. Este algoritmo es simulado para imágenes capturadas por una cámara GoPro colocada en un UAV de ala fija del grupo de investigación ATA durante una prueba de vuelo realizada en la Reserva Ecológica Antisana.

Por otro lado, el diseño del gimbal se basa en un mecanismo de 3 eslabones que permitan rotar una cámara Raspberry en alabeo y cabeceo utilizando dos servomotores que serán controlados por una tarjeta NVIDIA Jetson Nano. La estructura del gimbal es diseñado para un perfil aerodinámico de bala para que su fuerza de arrastre no afecte el vuelo del UAV. También el diseño procura una manufactura sencilla y de bajo peso para adaptarse a la máxima carga útil del UAV.

Palabras clave: Gimbal, Visión Computacional, algoritmo, línea del horizonte, UAV, NVIDIA Jetson Nano, ángulo de alabeo.

GLOSSARY OF TERMS

Abbreviations

UAV	Unmanned Aerial Vehicles
IMU	Inertial Measurement Unit
VTOL	Vertical Takeoff and Landing
FPS	Frames Per Second
MP	Megapixels
RGB	Red, Green, and Blue
HSV	Hue, Saturation, and Value
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Feature
FAST	Features from Accelerated Segment Test
BRIEF	Binary Robust Independent Elementary Features
ORB	Oriented FAST and Rotated BRIEF
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
ReLU	Rectifier Linear Unit
CSI	Camera Serial Interface
QFD	Quality Function Deployment
CAD	Computer Aided Design
RPM	Revolutions Per Minute

Constants and symbols

\rightarrow_g	Acceleration of gravity (9.8 m/s^2)
R	Rotation matrix
H	Homography matrix
K	Kernel matrix
V	Voltage [V]
A	Current [A]
P	Power [W]
C	Battery capacity [mAh]
F	Force [N]
C_d	Drag Coefficient [N]

D Drag Force [N]

Greek letters

α Roll angle [rad]

β Pitch angle [rad]

γ Yaw angle [rad]

θ Rotation angle [rad]

σ Normal stress [MPa]

τ Shear stress [MPa]

ρ Air density [Kg/m^3]

DISEÑO Y SIMULACIÓN DE UN ESTABILIZADOR DE CÁMARA CONTROLADO POR VISIÓN COMPUTACIONAL PARA UN VEHÍCULO AÉREO NO TRIPULADO DE ALA FIJA

INTRODUCTION

The use of fixed-wing unmanned aerial vehicles (UAVs) has had a strong research impact in the development of applications for environmental monitoring, species protection, surveillance, and security control [1]. Due to its endurance and large coverage area, they are preferred to multi-rotor configurations. One of the most important devices for UAVs are their cameras, which mostly use computer vision algorithms to fulfill tasks such as identifying and tracking objects. These cameras use gimbals or stabilizers to focus the image correctly.

In the industry, there are gimbals for UAVs like DJI [2], which are made for multicopters, but due to their configuration, the gimbal system is assembled together with the drone to fulfill only specific tasks. UKR Spect Systems [3] are gimbals for fixed wing UAVs, which has functionalities such as mapping, object tracking and their ability to be coupled to different types of aerial vehicles. However, their main disadvantages are their very high cost since they have additional elements such as a laptop with its own graphical interface. Furthermore, their computer vision algorithms are not free to use, preventing the user from modifying them.

In Ecuador, the research group in AERONAUTICS AND APPLIED THERMOFLUIDS (ATA) of Escuela Politécnica Nacional (EPN) has led several projects using fixed-wing UAVs, however, they do not have a gimbal to stabilize a camera in high wind environments. Videos taken by a GoPro camera mounted on a UAV have stabilization issues in flight tests carried out in the Antisana Ecological Reserve.

Thus, this thesis as part of a VLIR-UOS joint project between the ATA research group of the EPN and the research group in EMBEDDED AND ARTIFICIALLY INTELLIGENT VISION ENGINEERING (EAVISE) of KU Leuven, "Real-time volcano monitoring for early eruption prediction using unmanned aerial vehicles and image processing methods" [4] focuses on the design and simulation of a gimbal system for a fixed-wing UAV capable of stabilizing by computer vision algorithms with the great advantage of having an open source code to modify it for different applications.

General Objective

To design and simulate of a gimbal for fixed wing UAVs, computational vision controlled.

Specific Objectives

- To do a review of the state of the art related to UAV gimbals and computer vision algorithms for feature detection, matching and segmentation of images.
- To train a neural network capable of stabilizing the movement of the gimbal by allowing the identification and tracking of a target.
- To select the most suitable materials and electronic components for the design of a fixed-wing UAV gimbal.
- To design an aerodynamic structure for a fixed-wing UAV gimbal.
- To design assembly and gimbal components drawings.

Scope

In the present work, the design of a gimbal will be carried out in order to select the materials and electronic components necessary to control the roll and pitch rotation angle of a camera placed on a fixed-wing UAV. On the other hand, an algorithm will be developed using the Python programming language to detect and track the line of the sky that will allow control of the roll rotation angle from a set of images provided by a camera of a fixed-wing UAV of the ATA research group.

1. BACKGROUND / LITERATURE REVIEW

In this chapter the definition of gimbal, its parts and the main Computer Vision algorithms that will be used to stabilize it will be developed. In addition, the basic concepts of image processing with OpenCV and Machine Learning are presented.

1.1. Unmanned Aerial Vehicles

An unmanned aerial vehicle (UAV) is an aircraft without any passengers controlled through electromechanical components. In recent years, these vehicles have been used for scientific and environmental applications. The two most common types include multi-rotor and fixed-wing [5].

1.1.1. Multi-rotor UAV

A multi-rotor UAV as shown in Figure 1.1 is an aircraft that lifts thanks to the moments and forces generated by the rotors. Among the main advantages of using this aircraft is the ease of Vertical Takeoff and Landing (VTOL) and hover flight. In addition, autonomous flights can be carried out through control algorithms. On the other hand, they have the great disadvantage of flying at short times and distances at a low altitude, that is, their endurance is low [6].



Figure 1.1. Multi-rotor UAV

(Source: [6])

1.1.2. Fixed-wing UAV

A fixed-wing UAV as shown in Figure 1.2 is an aircraft that lifts thanks to the lift force generated by a rigid wing structure. Its thrust force is generated by the use of propellers or rotors located mainly on the sides or nose of the aircraft. Its main advantage is its high endurance, being able to cover large areas in just one flight. For this reason, the use of fixed-wing UAVs is preferred for research, mainly for environmental monitoring. On the other hand, its disadvantage is that the pilot must have great maneuverability to make a soft landing to prevent damage to electronic components [5].



Figure 1.2. Fixed-wing UAV

(Source: [6])

In aeronautics, 3 coordinate axes perpendicular to the aircraft's center of gravity and 3 rotation angles that rotate through them are defined to control altitude and orientation during a flight.

For a fixed-wing aircraft, the yaw angle is the rotational movement made by the aircraft with respect to the vertical axis perpendicular to it; the pitch angle is the rotation movement made by the aircraft with respect to the axis parallel to the wings of it, and the roll angle is the rotation movement made by the aircraft with respect to the axis that joins the nose and tail of the aircraft as shown in Figure 1.3 [7].

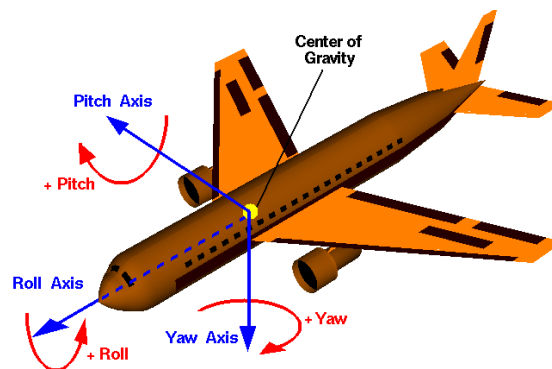


Figure 1.3. Fixed-wing aircraft rotations.

(Source: [7])

1.2. Camera gimbals

A gimbal is a mechanical device that by rotating its body through two or three pivot axes allows a camera to stabilize an image frame obtaining optimal and stable captures. These devices are commonly used for professional cameras, smartphones, or UAVs. They can range from simple designs such as a tripod or selfie stick, to electromechanical systems that use microprocessors and Inertial Measurement Units (IMU) sensors [8].

Currently, the use of gimbals for UAVs has reinvented the industry since an aircraft will always be exposed to wind forces and vibrations. The gimbal allows the images or videos captured by the camera have a high quality like the DJI brand that manufactures multi-rotor

UAVs with gimbals. In addition, to protect the camera and obtain less drag, they use aerodynamic structures like a sphere or bullet type as shown in Figure 1.4 [9].



Figure 1.4. Eo Camera Gimbal.

(Source: [10])

1.2.1. Components of a gimbal for a UAV

A gimbal for a UAV is mainly assembled with the following components: Mechanical system, control system, power supply, and camera. Figure 1.5 shows the main parts of a gimbal for UAVs.

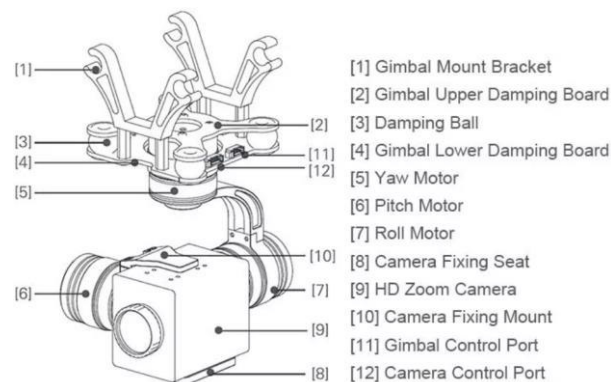


Figure 1.5. Digital EAGLE Camera Gimbal Parts.

(Source: [11])

Mechanical system

It is the structure where the camera will be placed, commonly with dampers to isolate vibrations. In addition, it has a mechanism that will allow movements of up to 3 degrees of freedom through roll-pitch-yaw rotation angles similar to those of an aircraft as shown in Figure 1.6 [12].

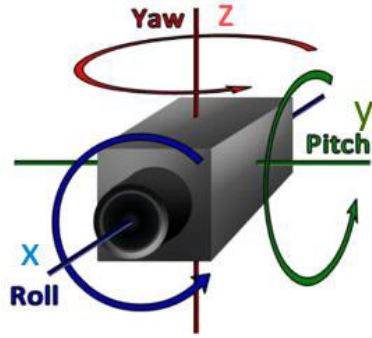


Figure 1.6. Rotation for roll, pitch, and yaw angles.

(Source: [13])

Rotation matrices are used to represent the rotation of a body in Euclidean space. Equations (1.1), (1.2), and (1.3) show the rotation matrices around the x, y, and z axes, respectively.

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \quad (1.1)$$

$$R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (1.2)$$

$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

The general rotation matrix R is shown in equation (1.4).

$$R = R_z(\gamma) R_y(\beta) R_x(\alpha) \quad (1.4)$$

$$R = \begin{bmatrix} \cos \gamma \cos \beta & \cos \gamma \sin \beta \sin \alpha - \sin \gamma \cos \alpha & \cos \gamma \sin \beta \cos \alpha + \sin \gamma \sin \alpha \\ \sin \gamma \cos \beta & \sin \gamma \sin \beta \sin \alpha + \cos \gamma \cos \alpha & \sin \gamma \sin \beta \cos \alpha - \cos \gamma \sin \alpha \\ -\sin \beta & \cos \beta \sin \alpha & \cos \beta \cos \alpha \end{bmatrix} \quad (1.5)$$

Where:

α Roll angle [rad]

β Pitch angle [rad]

γ Yaw angle [rad]

Given an arbitrary rotation matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{24} & r_{25} & r_{26} \end{bmatrix} \quad (1.6)$$

Assuming that $r_{11} \neq 0$ and $r_{33} \neq 0$. The equations to calculate the rotation angles Yaw, Pitch and Roll are shown in equations (1.7), (1.8) and (1.9) respectively.

$$\gamma = \tan^{-1} \left(\frac{r_{21}}{r_{11}} \right) \quad (1.7)$$

$$\beta = \tan^{-1} \left(\frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}} \right) \quad (1.8)$$

$$\alpha = \tan^{-1} \left(\frac{r_{32}}{r_{33}} \right) \quad (1.9)$$

Control system

The control of a gimbal is determined by 3 elements:

Sensor: It measures system disturbances. An IMU sensor as shown in Figure 1.7 is commonly used because it has 3 essential sensors included, these are an accelerometer, gyroscope, and magnetometer.

The accelerometer measures linear acceleration through 3 perpendicular axes located at the center of gravity of the IMU sensor. In the vertical axis Z, there will always be an acceleration due to gravity of approximately 9.8 m/s^2 . The gyroscope measures angular velocity in degrees per second or revolutions per second (RPS). Finally, the magnetometer measures the magnetic field strength in Teslas (T) of the sensor with respect to the Earth [14].

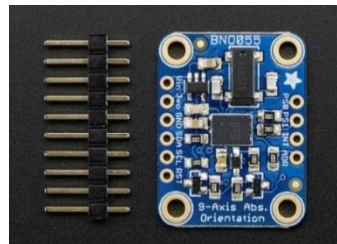


Figure 1.7. IMU sensor.

(Source: [15])

Processor: It receives the data measured by the sensor to execute a response through the actuators. A Drone Gimbal Control Unit (GCU) board as shown in Figure 1.8 is commonly used [16].



Figure 1.8. Drone Gimbal Control Unit.

(Source: [17])

Actuators: They allow the mechanical system to move due to the signals emitted by the processor. DC motors or servo motors can be used, but brushless as shown in Figure 1.9 are commonly used [16].



Figure 1.9. Brushless Motor

(Source: [18])

Power Supply

Batteries supplies electrical energy through chemical energy to small aircraft. Generally, the battery used by the UAV as shown in Figure 1.10 is the one that supplies power to the gimbal. For the correct selection of a battery, it is necessary to consider the maximum consumption voltage, its capacity expressed in milliamp hours, its discharge rate, and its energy density [19].



Figure 1.10. UAV Battery.

(Source: [20])

Camera

A camera is an electronic device that has light sensors to produce images. An aircraft can use different types of cameras depending on its application. The most used cameras are

the action cameras that are not directly connected to the aircraft and store their own information on a SD card; GoPro cameras belong to this type as shown in Figure 1.11.

On the other hand, the First View Person cameras (FVP) are directly connected to the aircraft control system to develop specific tasks such as target tracking; DJI brand cameras belong to this type [21].



Figure 1.11. Tarot GoPro Camera Gimbal.

(Source: [22])

A camera works by transforming real-world 3D coordinates to 2D coordinates on a plane as shown in Figure 1.12. The coordinates of a plane in the real world (x, y) are transformed to coordinates of an image (u, v) by a homography matrix H which is shown in equation (1.10) [23].

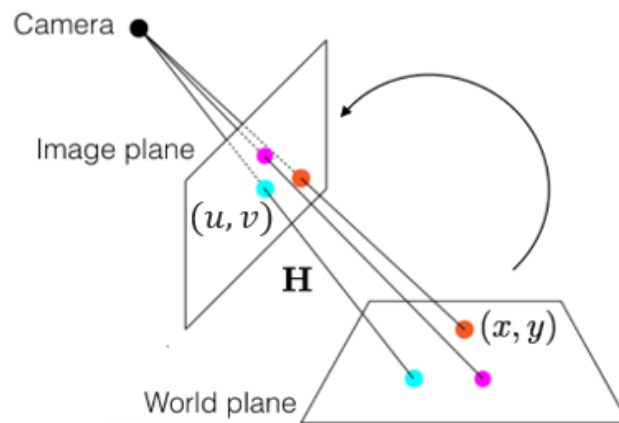


Figure 1.12. Homography matrix between a world plane and an image plane.

(Source: [24])

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.10)$$

The 3x3 homography matrix is the product of the intrinsic parameters of a camera (focal length, optical center) times its extrinsic parameters (rotation and translation matrix) which is shown in equation (1.11).

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & 1 \end{bmatrix} \quad (1.11)$$

$$C_x = \frac{w_c}{2} \quad (1.12)$$

$$C_y = \frac{h_c}{2} \quad (1.13)$$

Where:

r_{ij} Element from a rotation matrix [rad]

f Focal length (distance between the image plane and the focal point) [pixels]

c_x Optical center width [pixels]

c_y Optical center height [pixels]

w_c Width of the captured image [pixels]

h_c Height of the captured image [pixels]

t_x Translation in x axis [pixels]

t_y Translation in y axis [pixels]


1.2.2. Types of gimbals (Benchmarking)

In this section, a review of gimbals for UAVs in the market is carried out to identify design improvements.

Tarot ZYX T2D-10X gimbal

It is a gimbal with 2 degrees of freedom (Pitch and Yaw) that can be used in both fixed-wing and multirotor UAVs. Can be installed normally or inverted (turned upside down). Its technical specifications are shown in Table 1.1.

Table 1.1. Technical specifications of the Tarot ZYX T2D-10X gimbal.


Technical specifications of the Tarot ZYX T2D-10X gimbal			
			
Mechanical system	<p>Dimensions: 70x81x109 mm</p> <p>Material: Composite material</p> <p>Weight: 303 g</p> <p>Working Temperature: -10 °C to +60 °C</p> <p>Image stabilization accuracy: ± 0.01°</p> <p>Controllable rotation range:</p> <p>Yaw Angle: ±200° (Lock Mode)</p> <p>Pitch Angle: +30° to -120° (Normal installation)</p>	Control system	<p>Control Method: S.bus; Can; Serial Port</p> <p>Control Interface: Micro HDMI</p> <p>Actuators: Brushless Motors</p>
Power supply	<p>Working Voltage: 12V-25 V</p> <p>Power: 6.5 W</p>	Camera	<p>Lens: 10x optical zoom f4.9-49 mm</p> <p>Working Mode: Tacking pictures; recording</p> <p>Resolution: 1920x1080 30 FPS</p>
Cost	699 USD		

(Source: [25])

SKY EYE-36SZ 1080P 36X gimbal

It is a gimbal with 3 degrees of freedom capable of surveillance, monitoring, inspection, and rescue applications. Its technical specifications are shown in Table 1.2.

Table 1.2. Technical specifications of the SKY EYE-36SZ 1080P 36X gimbal.

Technical specifications of the SKY EYE-36SZ 1080P 36X gimbal			
			
Mechanical system	<p>Dimensions: 106x188x151.5 mm</p> <p>Material: Composite material</p> <p>Weight: 945 g</p> <p>Image stabilization accuracy: $\pm 0.01^\circ$</p> <p>Controllable rotation range:</p> <p>Pitch Angle: $\pm 90^\circ$</p> <p>Roll Angle: $\pm 45^\circ$</p> <p>Yaw Angle: $\pm 150^\circ$</p>	Control system	<p>Control Interface: PWM and serial command</p> <p>Sensor: 1/1.9 inch 2.38 MP CMOS sensor</p> <p>Actuators: Brushless Motors</p>
Power supply	<p>Working Voltage: 12V</p> <p>Power: ≤ 6.6 W</p>	Camera	<p>Lens: 36X optical zoom, 12X digital zoom</p> <p>Working Mode: Tracking pictures;</p>

			recording; auto object tracking; geotagging; night vision capability Resolution: 1080P 30 FPS
Cost	4750 USD		

(Source: [26])

DJI ZENMUSE X5S gimbal

It is a gimbal with 3 degrees of freedom manufactured exclusively for the DJI Inspire 2 aircraft (multirotor UAV) which can be controlled through the DJI app. Its technical specifications are shown in Table 1.3.

Table 1.3. Technical specifications of the DJI ZENMUSE X5S gimbal.

Technical specifications of the DJI ZENMUSE X5S gimbal.			
			
Mechanical system	Dimensions: 140x98x132 mm Material: Composite material Weight: 461 g Working Temperature: -20 °C to 40 °C Controllable rotation range: Pitch Angle: -130° to 40° Roll Angle:	Control system	Control Method: DJI GO 4 (App) Actuators: Brushless Motors

	±320° Yaw Angle: ±320°		
Power supply	Aircraft DJI Inspire 2 battery: 6000 mAh 2S LiPo	Camera	Lens: DJI MFT 15 mm/1.7 ASPH (it supports up to eight professional lenses) Working Mode: Tacking pictures; recording Resolution: 5280×2970
Cost	2049 USD		

(Source: [27])

1.3. Computer vision and OpenCV

Computer Vision is a multidisciplinary field that uses digital image acquisition, processing, and analysis methods to produce numerical or symbolic information to accomplish a given task. OpenCV (Open-Source Computer Vision Library) is an open-source image processing library created by Intel in 2000. The algorithms in this library can be run in programming languages such as Python and allow to solve Computer Vision problems [12].

Using the OpenCV shape function we can access the properties of an image. The function returns a tuple with the rows (pixel height), columns (pixel width), and channels of the image. The channels represent the composition of colors that an image has; in the case of an RGB image (Red, Green, and Blue) it is 3, and in the case of gray scale it is 1. The pixel of an image can be accessed by its row and column coordinates. For an RGB pixel, the returned value is an array of three values between 0 and 255, which are the intensity of the red, green, and blue colors, respectively. For a grayscale pixel, only a value of intensity between 0 and 255 is returned; 0 is black and 255 is white [28].

On the other hand, graphic programs also work with another color model such as HSV (Hue, Saturation and Value) due to the advantage of adjusting a color by its brightness and saturation, being more similar to the perception of color by the human eye. In OpenCV the Hue value is between 0 and 180, the Saturation between 0 and 255 and the Value (brightness) between 0 and 255 [29]. The RGB and HSV color models can be represented by a cube and a cylinder, respectively, as shown in Figure 1.13.

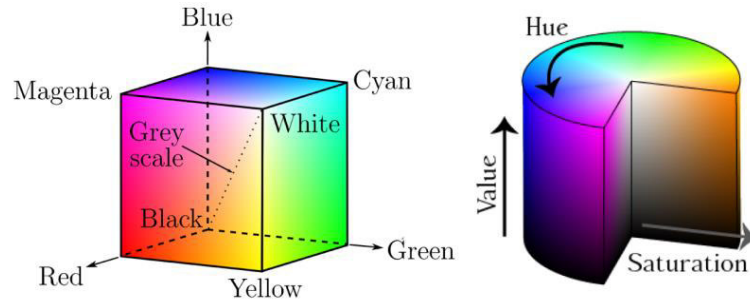


Figure 1.13. RGB Cube and HSV Cylinder.

(Source: [30])

This section will explain the main algorithms used for the gimbal stabilization.

1.3.1. Feature detection

Feature Detection is a branch of Computer Vision widely used for target detection and camera calibration. A feature is a piece of information of an image that can be detected as corners, edges, lines, etc. Once a point of interest is identified, it is described according to its neighboring pixels that there is no sudden change of properties such as its brightness. Finally, matching is applied, which consists of comparing the feature points of two or more images of the same scene [31].

SIFT, SURF, and ORB libraries

SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Feature), and ORB (Oriented FAST and Rotated BRIEF) are OpenCV libraries that allow the detection, description and matching of features. SIFT and SURF have the advantage of detecting more feature points than ORB, however, ORB is faster [32].

In addition, this algorithm usually works together with the Image Stitching algorithm, which consists of applying the concept of the homography matrix as shown in Figure 1.14 to project an image on another and obtain an enlarged image. This process is applied in panoramic photography. An example of using the SIFT and Image Stitching algorithms is shown in Figure 1.15.

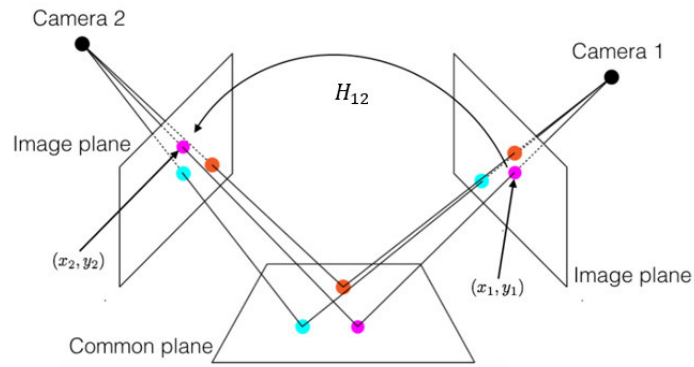


Figure 1.14. Homography matrix between two image planes.

(Source: [24])

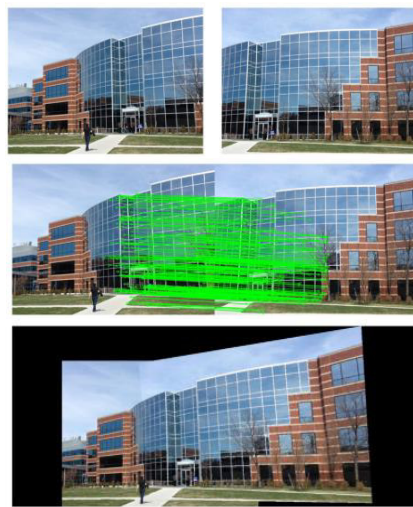


Figure 1.15. SIFT and Image Stitching algorithms.

(Source: [31])

Canny Edge Detection library

To explain how the edge detection algorithm works, it is necessary to explain what a kernel matrix is. A kernel is a square convolution matrix of odd order which is widely applied for image filters. The center value of the kernel matrix is multiplied by the intensity value of the image pixel to which the filter is applied. Furthermore, the other values of the matrix will be multiplied by their corresponding pixel intensities of the image. The sum of all the final values replaces the intensity value of the current pixel. This process is shown in equation (1.14) and Figure 1.16. The result of applying this process to all image pixels is called a convolved image [33].

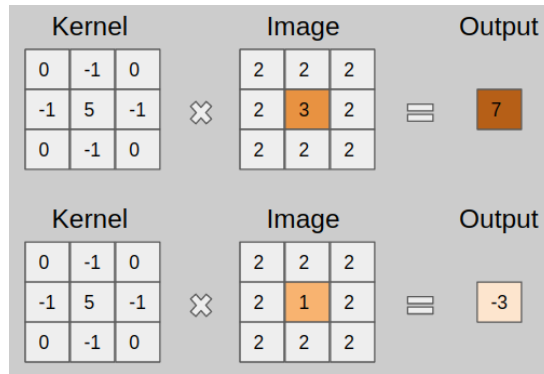


Figure 1.16. Kernel example.

(Source: [34])

$$I' = K * I = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mn} \end{bmatrix} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} y_{(1+i)(1+j)} \quad (1.14)$$

Where:

I' Pixel intensity from the center of the image matrix

K Kernel matrix

I Image matrix

$*$ Convolution operator

x_{mn} Kernel matrix element

y_{mn} Image pixel intensity

The 3x3 kernel matrix for applying a Gaussian blurring filter can be approximated as shown in equation (1.15).

$$K = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (1.15)$$

OpenCV has many filters. For blurring images, the median filter (cv.MedianBlur), the 3x3 Gaussian filter, the 5x5 Gaussian filter (cv.GaussianBlur) as shown in Figure 1.17, the bilateral filter (cv.bilateralFilter), and a custom kernel filter (cv.filter2D) are mainly used [35].



Figure 1.17. Gaussian Blurring.

(Source: [35])

Canny Edge Detection is a library available in OpenCV that allows identifying edges (boundaries) of regions of interest in an image. The process consists of transforming the image into gray scale, then the noise in the image is removed by applying a Gaussian filter to calculate the gradient magnitude of each pixel as shown in equation (1.16). After that, false edges that are not in the direction of the gradient shown in the equation (1.17) are removed. Finally, an image processing known as Hysteresis Thresholding is applied. An example of this algorithm is shown in Figure 1.18 [36].

$$G = \sqrt{G_x^2 + G_y^2} \quad (1.16)$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (1.17)$$

G_x and G_y are the intensity of the gradient in the x and y directions, respectively, and are calculated using equations (1.18) and (1.19).

$$G_x = K_A * I \quad (1.18)$$

$$G_y = K_B * I \quad (1.19)$$

Where:

K_A X – Direction Sobel Kernel Matrix

K_B Y – Direction Sobel Kernel Matrix

I Image matrix

* Convolution operator

K_A and K_B are called Kernel Sobel matrices which are shown in equations (1.20) and (1.21), respectively.

$$K_A = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad (1.20)$$

$$K_B = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (1.21)$$



Figure 1.18. Canny Edge Detection algorithm.

(Source: [37])

Hysteresis threshold image processing identifies if an edge is really an edge. For this, the magnitude of the calculated pixel gradient is compared between a maximum and minimum threshold value (relative image lightness). If the gradient is greater than the maximum threshold value, it is considered an edge. If the gradient is less than the minimum threshold value, it is not considered an edge [38].

1.3.2. Curve fitting algorithm

Curve Fitting is an optimization problem to get a line in a plane that best fits an arbitrary data collection as shown in Figure 1.19. The least squares method is a numerical analysis technique applied to curve fitting that minimizes the sum of the squares of the residuals (difference between an arbitrary value and a fixed value provided by a model) to find a continuous function [39].

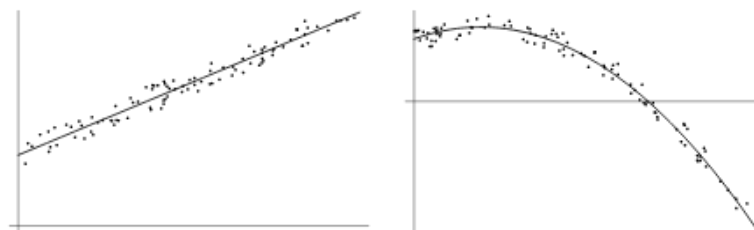


Figure 1.19. Curve Fitting.

(Source: [39])

To fit the data to a straight line as shown in equation (1.22), linear regression equations (1.23) and (1.24) are considered.

$$Y = aX + b + \epsilon \quad (1.22)$$

$$a = \frac{\sum_{i=1}^n (x_i y_i) - n \bar{x} \bar{y}}{\sum_{i=1}^n (x_i^2) - n \bar{x}^2} \quad (1.23)$$

$$b = \bar{y} - a \bar{x} \quad (1.24)$$

Where:

a Slope

b y – axis intersection

ϵ Error

n Number of values

\bar{x} Mean of x values

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1.25)$$

\bar{y} Mean of y values

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (1.26)$$

1.3.3. Machine Learning

Machine Learning is a branch of artificial intelligence that allows machines to learn patterns from input data to obtain output predictions without programming it. The "learning" process is called training, which provides the machine with input data and its desired outputs (ground truths). For this, Machine Learning uses Artificial Neural Networks (ANN).

There are three types of machine learning. The first is supervised learning, which consists of each input data being paired with its corresponding known classification label. The second is unsupervised learning, which consists of the algorithm learning only from its trained input data by finding similarities. The third is semi-supervised learning where part of the data is labeled and other parts are unlabeled [40].

Artificial Neural Network (ANN)

An artificial neuron is a mathematical function shown in equation (1.27) that resembles the process of biological neurons. The structure of an artificial neuron is shown in Figure 1.20 [41].

$$Z = f(b + \sum_{i=1}^m (W_i X_i)) \quad (1.27)$$

Where:

X_i Inputs (Values between 0 and 1)

Z Output (Values between 0 and 1)

W_i Weights

b Bias

f Activation Function

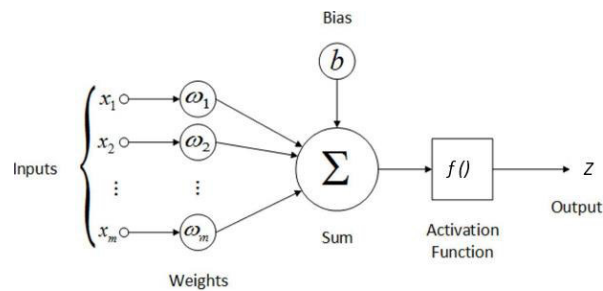


Figure 1.20. ANN structure.

(Source: [42])

The weight of a neuron allows the model to learn. The bias term allows the model to adjust to the activation function. Initially, the neurons have random values of weights and bias. During the first iteration, the neural network can predict the class to which the trained data belongs; this process is called forward propagation. On the other hand, the weight and bias values will be adjusted for each cycle that is repeated; this process is called backpropagation [41].

The clustered and connected set of artificial neurons is called an Artificial Neural Network (ANN), and its structure for a supervised learning is shown in Figure 1.21. A neural network model can improve its performance by parameters that can be entered manually, also called hyperparameters. These are: The number of layers, activation function, optimizer, lost function, epoch, batch size, and learning rate [43].

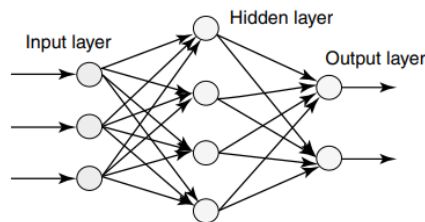


Figure 1.21. Layers of a supervised learning Artificial Neural Network.

(Source: [41])

Number of layers

A layer of a neural network consists of neurons. There are three types of layers:

- **Input layer:** It receives the training input data. Its inputs are equal to its neurons.
- **Hidden layer:** It is responsible for improving the performance of the neural network. Hidden layers can be one or more in number.
- **Output layer:** This layer returns the results of the training of the neural network. For a classification problem, its neurons are equal to its outputs [41].

Activation Function

They are used by each neuron in the network and indicate when a neuron turns on or off, depending on which activation function is used. The most used activation functions are Sigmoid and ReLU (Rectifier Linear Unit) as shown in Figure 1.22 [43].

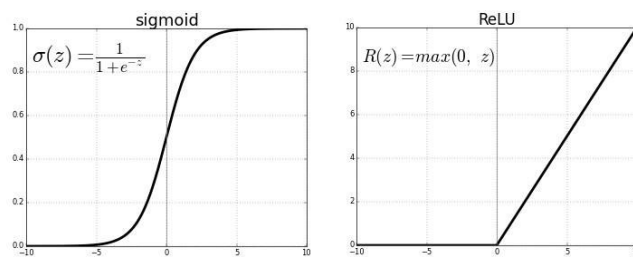


Figure 1.22. Sigmoid and ReLU activation functions.

(Source: [44])

Optimizer

It is an algorithm in charge of optimizing the neural network. The most used is Adam [43].

Loss Function

This parameter, also called Cost Function, shows whether the neural network is performing well or not. This function is minimized when the backpropagation process is performed. For classification problems, the Cross Entropy function is used. When the number of classes (targets or categories) equals 2, cross-entropy can be calculated as shown in equation (1.28) [45].

$$CE = -(y \log(p) + (1 - y) \log(1 - p)) \quad (1.28)$$

Where:

CE Cross – entropy loss function

y Ground Truth probability (0 or 1)

p Class probability (Activation Function)

Epoch

It is the number of times the forward propagation and backpropagation algorithms are executed, passing all the training data through the neural network [46].

Batch size

It is the number of data that each epoch has [46].

Step

It indicates the number of steps to execute on the input data as shown in equation (1.29) [46].

$$\text{Number of steps} = \frac{\text{Number of input data}}{\text{Batch Size}} \quad (1.29)$$

Iterations

One iteration corresponds to each time a batch of data is passed through the neural network as shown in equation (1.30) [46].

$$\text{Number of iterations} = \frac{\text{Number of epochs}}{\text{Batch Size}} \quad (1.30)$$

Learning rate

The learning rate indicates how much the value of the weights of a network is adjusted with respect to its loss gradient as shown in equation (1.31).

$$W' = W - a \frac{\partial}{\partial W} J(W) \quad (1.31)$$

Where:

W' New Weight

W Old Weight

a Learning rate

J Loss Function

The learning results of a neural network are determined by the learning rate curve shown in Figure 1.23, and through the validation and training curves shown in Figure 1.24 [47].

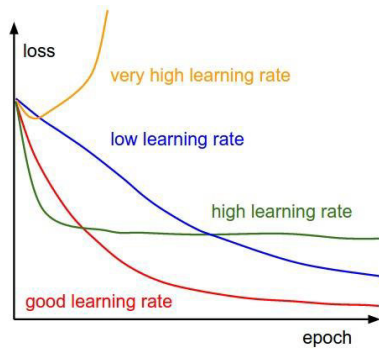


Figure 1.23. Learning rate curves.

(Source: [48])

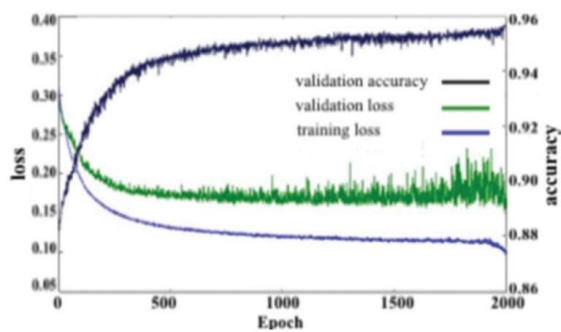


Figure 1.24. Loss and accuracy curves of a learning neural network.

(Source: [47])

Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is a type of supervised learning artificial neural network that uses images as inputs. The network works with the pixels of an image, therefore their intensity values ranging from 0 to 255 are normalized to values from 0 to 1 that a neural network can identify.

Convolutions consist of taking pixels from an image and multiplying them by a set of kernel matrices, also called filter. Then an activation function is applied, the most common for CNN is ReLU. The set of output matrices of each convolution is called feature mapping, which is a hidden layer of neurons. The first hidden layer can identify simple features like lines or points and the last hidden layer can identify more complex features like objects [49].

Working with image pixels, a very large number of matrices can be obtained. For this reason, it is necessary to do a subsampling, which reduces the size of the filtered image, prevailing the most important features detected by each filter. There are different types of subsampling, the most used is called Max-Pooling. Finally, the Softmax function is applied to the last layer, connects the number of neurons with their classes. The structure of a convolutional neural network is shown in Figure 1.25 [49].

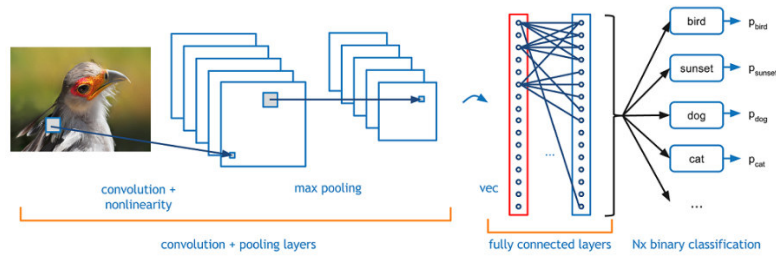


Figure 1.25. CNN example.

(Source: [50])

Image Segmentation based on Convolutional Neural Network

Image Segmentation is a Computer Vision technique that partitions an image into multiple segments allowing easy analysis of it. There are many algorithms that allow segmenting an image, however, due to the advantage of being applied in real time, a Convolutional Neural Network is used [51].

"U-Net: Semantic Segmentation" is a neural network used for multiclass segmentation of images with their corresponding masks (ground truths) as shown in Figure 1.26. This network uses the Python programming language and PyTorch as a framework. Figure 1.27 shows the architecture of this CNN [52].

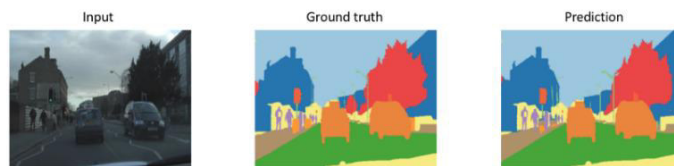


Figure 1.26. U-Net Image Segmentation.

(Source: [53])

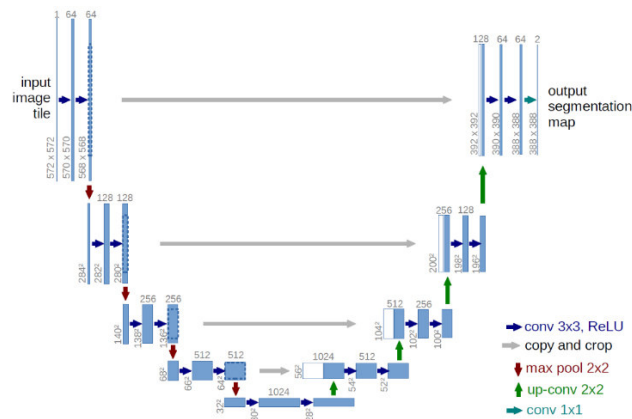


Figure 1.27. U-Net Architecture.

(Source: [52])

2. METHODOLOGY

This chapter develops the methodology of this thesis project is based on Carles Riba's Concurrent Design Methodology [54]. As shown in Figure 2.1, the methodology has been divided into 5 stages: Definition of the gimbal, conceptual design, materialization design, detailed design, and results.

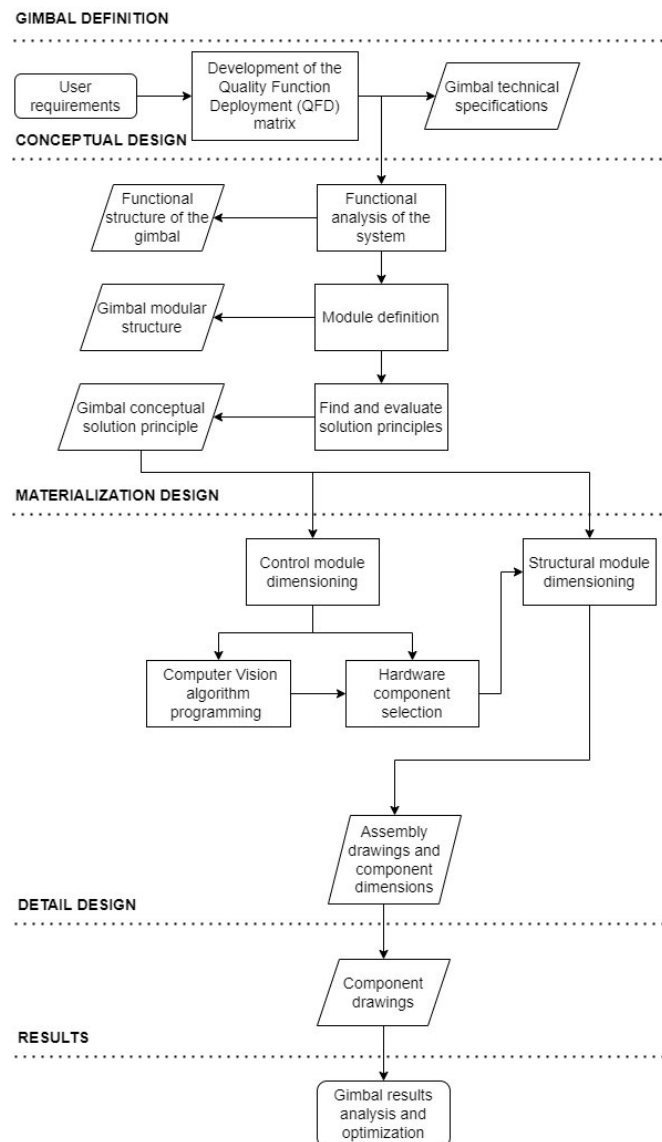


Figure 2.1. Methodology.

(Source: Own)

2.1. Gimbal Definition

To ensure that the user's requirements are considered as a priority in the definition of the gimbal, a Quality Function Deployment (QFD) is established. A QFD is a structured and systematic process to translate the requirements and desires of the user (user's voice) into technical or engineering requirements (engineer's voice) [54].

2.1.1. User's Voice and Engineer's Voice

Professors and researchers from the ATA research group at EPN and the EAVISE research group at KU Leuven are the user's voice. These ones are translated to the engineer's voice as show in the Table 2.1.

Table 2.1. User's voice and Engineer's Voice

User requirements	
User's voice	Engineer's voice
The gimbal mechanism must have 2 degrees of freedom (Pitch and Roll)	Degrees of freedom
Stabilize the Roll angle using Computer Vision	Algorithm Working Mode [Hz]
Have a wide rotation range	Angle rotation range [°]
Low cost	Life cycle cost [USD]
Low weight	Gimbal weight [Kg]
Small gimbal	Gimbal dimensions [mm]
The aerodynamics of the gimbal must not affect the flight of the UAV	Drag Coefficient
Low power consumption	Voltage [V]; Capacity [mAh]
Easy to assemble	Number of mechanical components

(Source: Own)

2.1.2. QFD Matrix

The QFD matrix shown in Figure 2.2 converts user requirements into technical product specifications. The matrix follows these steps:

1. **User's voice and engineer's voice:** It summarizes the user requirements which are shown in Table 2.1.
2. **Competitiveness analysis:** It describes, according to the user, the degree of satisfaction of the existing products on the market. For this section, the gimbals in Table 1.1 and Table 1.2 will be taken as reference.
3. **Correlations:** It indicates the relationship between the user's voice and the engineer's voice.
4. **Technical comparison:** It compares the product to be manufactured with products from other companies.
5. **Technical commitments:** Establishes the potential commitments of the technical characteristics of the product.

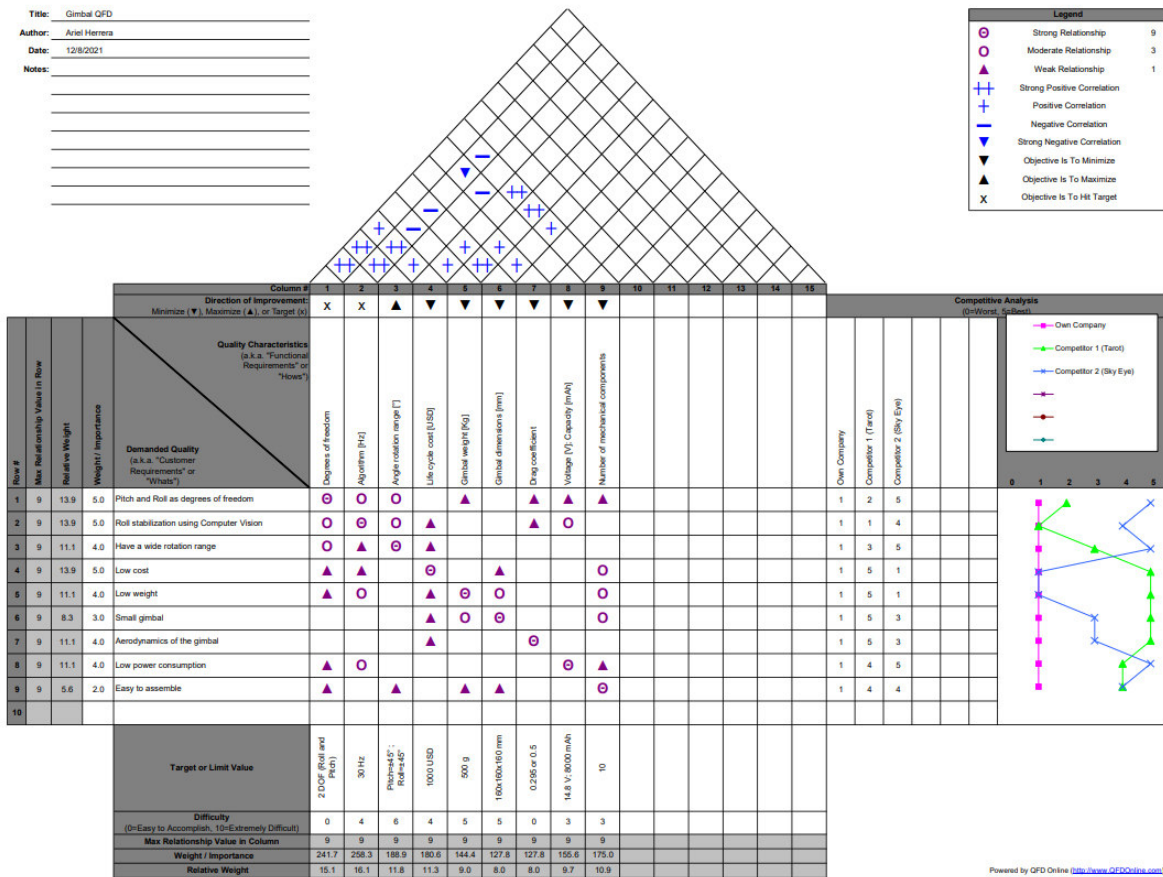


Figure 2.2. QFD matrix.
(Source: Own)

The requirements of the engineer’s voice with the highest incidence in the calculated QFD matrix are: The stabilization using a Computer Vision algorithm (16.1%), the number of degrees of freedom of the gimbal (15.1%), the angle rotation range (11.8%), and the cost of the product (11.3%). Furthermore, the stabilization algorithm has the strongest positive correlation in the calculated QFD Matrix. Therefore, the gimbal system will be design on meeting these requirements, giving more priority to the Computer Vision algorithm.

2.1.3. Gimbal Technical Specifications

Table 2.2 shows the technical specifications of the gimbal taken from the results of the QFD matrix.

Table 2.2. Gimbal Technical Specifications

Client company: ATA Research Group and EAVISE Research Group		Product: Computer vision-controlled gimbal for a fixed-wing UAV	
Designer: Ariel Patricio Herrera Aguiar			
Specifications			
Concept	Who proposes	R/D	Description
Degrees of freedom	C	R	The gimbal mechanism must have 2 degrees of freedom (Pitch and Roll).
Angle rotation range	E	MR	Roll angle $\approx \pm 45^\circ$; Pitch angle $\approx \pm 45^\circ$
Computer Vision algorithm	C	R	Roll angle must be stabilized by computer vision algorithms at 30 Hz
Cost	C	R	Maximum budget: 1000 USD
Weight	C	MR	≤ 500 g
Dimensions	C+E	R	$< 160 \times 160 \times 160$ mm
Drag Coefficient	C+E	MR	Bullet (0.295) / Sphere (0.5)
Voltage	C	R	< 14.8 V
Capacity	C	R	< 8000 mAh
Number of mechanical components	E	D	≤ 10
C=Client; E=Engineering; R=Requirement; MR=Modify Requirement; D=Desire			

(Source: Own)

2.2. Conceptual Design

In this design phase, a conceptual model of the gimbal will be obtained.

2.2.1. Functional analysis of the gimbal

The functional analysis identifies all the functions that the gimbal must perform to stabilize its roll angle using Computer Vision algorithms. According to a hierarchical order, the functions are divided into levels and their input and output flows are established as shown in Figure 2.3, 2.4 and 2.5.

Level 0.

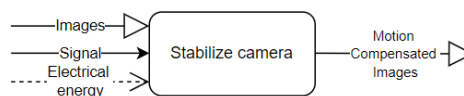


Figure 2.3. Level 0 of the functional structure.

(Source: Own)

Level 1.

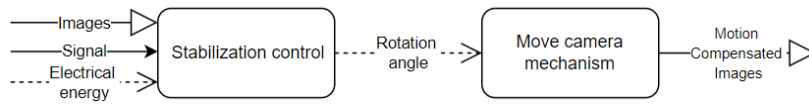


Figure 2.4. Level 1 of the functional structure.

(Source: Own)

Level 2.

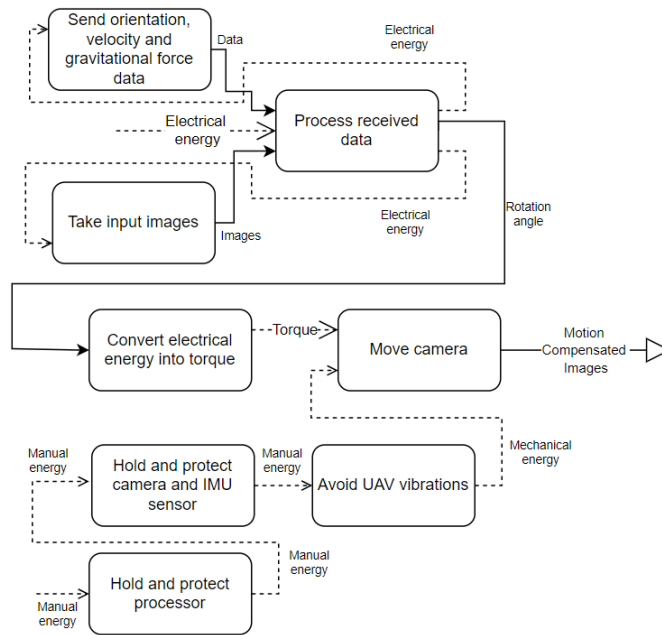


Figure 2.5. Level 2 of the functional structure.

(Source: Own)

2.2.2. Module Definition

Based on the functional structure, the modular structure is established, which groups the gimbal functions into blocks to facilitate design analysis as shown in Figure 2.6.

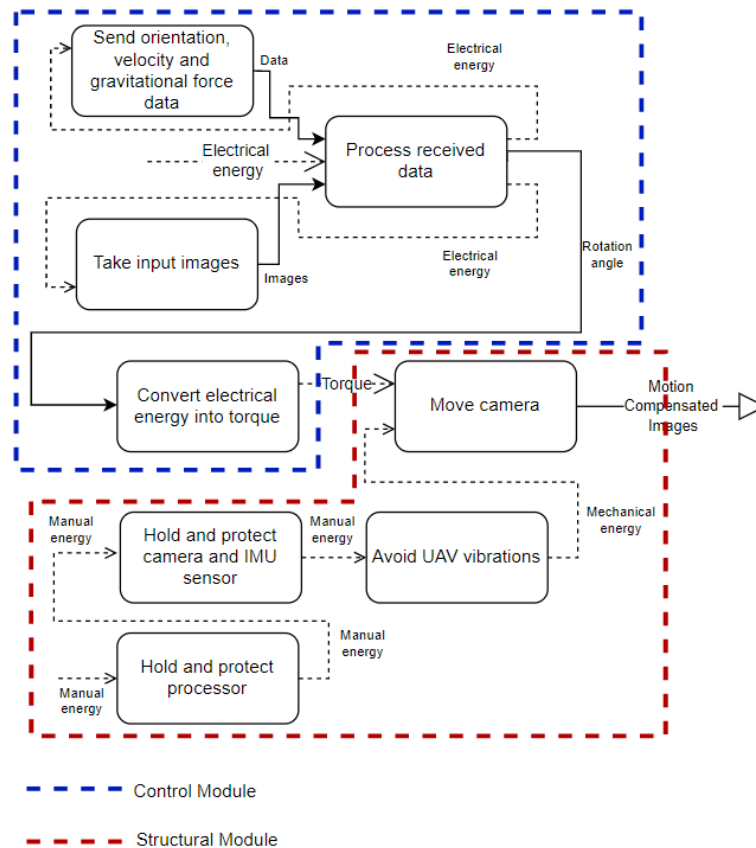


Figure 2.6. Modular structure.

(Source: Own)

2.2.3. Find and evaluate solution principles

In this section, alternatives are evaluated using qualitative criteria to fulfill the functions of each module. The evaluation is performed using the corrected weighted criteria ordinal method shown in Table 2.3. The solution principle with the highest score will be chosen.

Table 2.3. Corrected weighted criteria ordinal method.

Rating	Description
1	If the criterion of the rows is higher (or better <) than that of the columns
0.5	If the criterion of the rows is equivalent (=) that that of the columns
0	If the criterion of the rows is lower (or worse <) than that of the columns

(Source: Own)

2.2.3.1. Control module solution principles and evaluation

Table 2.4 shows the solutions and alternatives for the control module.

Table 2.4. Solution alternatives for the control module.

Function	Solution for functions
Send orientation, velocity, and gravitational force data	
Take input images	
Process received data	
Convert electrical energy into torque	
Alternative solutions	

(Source: Own)

Detailed solutions for each function are shown below.

Send orientation, velocity, and gravitational force data:

The IMU sensor is the only solution for this function because it combines three sensors which are the magnetometer, accelerometer, and gyroscope. For this design, an ADAFRUIT BNO055 IMU sensor with 9 degrees of freedom has been chosen, as shown in Figure 2.7. This sensor works with open-source codes, throwing data instantly.

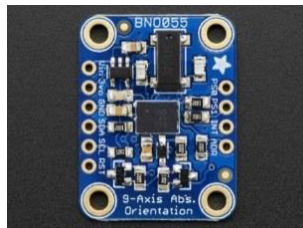


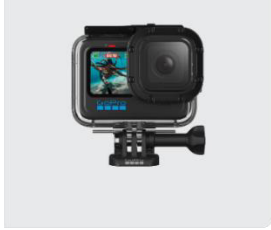

Figure 2.7. 9-DOF Absolute Orientation IMU Fusion Breakout.

(Source: [15])

Take input images:

There are two solutions to capture images for the control system with advantages and disadvantages as shown in Table 2.5.

Table 2.5. Solutions to “Take input images”.

Solutions	Advantages	Disadvantages	Figure
GoPro Camera	The quality of captured images is quite good, its USB connection makes it easy to assemble.	It is expensive, its weight is higher, it can be difficult to control for Computer Vision applications.	
Raspberry Pi Camera	It is light, its price is low, it can be used for Computer Vision applications, and it is easy to control.	Its Camera Serial Interface (CSI) cable cannot be very long. It would be necessary to look for an extension.	

(Source: [55] [56])

Process received data:

This is a very important function because it requires a processor that allows working with images and video in real time. There are 2 solutions for this function which are shown in Table 2.6. This processor will be connected to an OpenCR board as shown in Figure 2.8 which works as a voltage regulator and will control the gimbal motors.

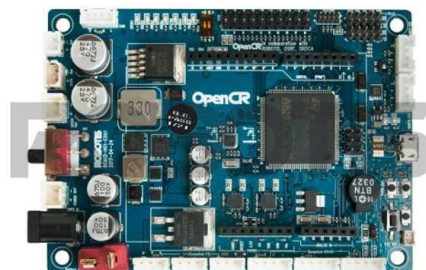




Figure 2.8. OpenCR1.0 controller.

(Source: [57])

Table 2.6. Solutions to “Process received data”.


Solutions	Advantages	Disadvantages	Figure
Jetson Nano	It has a dedicated NVIDIA GPU to work with images and video in real time, its processing speed is very good, it is compatible with Raspberry Pi accessories, it has ventilation to avoid overheating.	It is expensive, and it is difficult to acquire it in Ecuador.	 <p>The image shows the NVIDIA Jetson Nano 2GB development board. It is a small, green PCB with a black heat sink on top. The NVIDIA logo is visible in the top left corner, and the text '2GB' is printed in the bottom right corner.</p>
Raspberry Pi	It is light, cheap, its offline image processing is good, and it is easy to acquire in Ecuador.	It overheats easily, its video processor is insufficient for complex Computer Vision tasks.	 <p>The image shows a Raspberry Pi Model A+ V1.1. It is a small, green PCB with various ports and components. The Raspberry Pi logo is visible in the center.</p>


(Source: [58] [59])

Convert electrical energy into torque:

The actuators that can fulfill this function are motors. Table 2.7 shows 2 types of motors commonly used by a gimbal with their advantages and disadvantages.

Table 2.7. Solutions to “Convert electrical energy into torque”.

Solutions	Advantages	Disadvantages	Figure
Servomotors	It allows easy control of its speed, position, and torque. It is highly functional and high performance.	Its weight is higher, and it is expensive.	 <p>The image shows a Dynamixel servomotor. It is a black, rectangular motor with a circular top and a small window on the side. The Dynamixel logo is visible at the bottom.</p>

Brushless motors	It is light, cheap, compact, it has stable speed control.	Its control can be complicated since we can only control its speed.	
-------------------------	---	---	---

(Source: [60] [61])

To choose an alternative solution, the following criteria are formulated:

- Friendly environment that allows an easy and correct interaction between the hardware components and the Computer Vision algorithms.
- High processing capacity since the algorithms will use a lot of computational resources.
- Avoid overheating to prevent burning of hardware components.
- Compact since the hardware components must be adapted to the payload of the UAV and the maximum volume of its fuselage.

Using the corrected weighted criteria ordinal method, we proceed to evaluate each criterion and select the best alternative solution for the control module as shown in the following tables:

Table 2.8. Evaluation of the specific weight of each criterion.

High processing capacity > Friendly environment > Avoid overheating = Compact						
Criterion	High processing capacity	Friendly environment	Avoid overheating	Compact	$\Sigma +1$	Weight
High processing capacity		0.5	1	1	3.5	0.35
Friendly environment	0.5		0.5	1	3	0.30
Avoid overheating	0.5	0		0.5	2	0.20
Compact	0	0	0.5		1.5	0.15
Total					10	1.00

(Source: Own)

Table 2.9. Evaluation of the specific weight of the “High processing capacity” criterion.

Alternative 1 = Alternative 2 > Alternative 3					
High processing capacity	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		0.5	1	2.5	0.42
Alternative 2	0.5		1	2.5	0.42
Alternative 3	0	0		1	0.16
Total				6	1.00

(Source: Own)

Table 2.10. Evaluation of the specific weight of the “Friendly environment” criterion.

Alternative 1 > Alternative 3 > Alternative 2					
Friendly environment	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		1	1	3	0.50
Alternative 2	0		0	1	0.17
Alternative 3	0	1		2	0.33
Total				6	1.00

(Source: Own)

Table 2.11. Evaluation of the specific weight of the “Avoid overheating” criterion.

Alternative 1 > Alternative 2 > Alternative 3					
Avoid overheating	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		1	1	3	0.50
Alternative 2	0		1	2	0.33
Alternative 3	0	0		1	0.17
Total				6	1.00

(Source: Own)

Table 2.12. Evaluation of the specific weight of the “Compact” criterion.

Alternative 3 > Alternative 1 > Alternative 2					
Compact	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		1	0	2	0.33
Alternative 2	0		0	1	0.17
Alternative 3	1	1		3	0.50
Total				6	1.00

(Source: Own)

Table 2.13. Evaluation of conclusions of the control module.

Conclusions	High processing capacity	Friendly environment	Avoid overheating	Compact	Σ	Priority
Alternative 1	0.35*0.42	0.30*0.50	0.20*0.50	0.15*0.33	0.397	1
Alternative 2	0.3*0.17	0.35*0.42	0.20*0.33	0.15*0.17	0.264	2
Alternative 3	0.3*0.33	0.35*0.16	0.20*0.17	0.15*0.50	0.189	3

(Source: Own)

According to the results obtained in Table 2.15, Alternative 1 is determined as a priority for the control module. Figure 2.9 shows a schematic of Alternative 1.

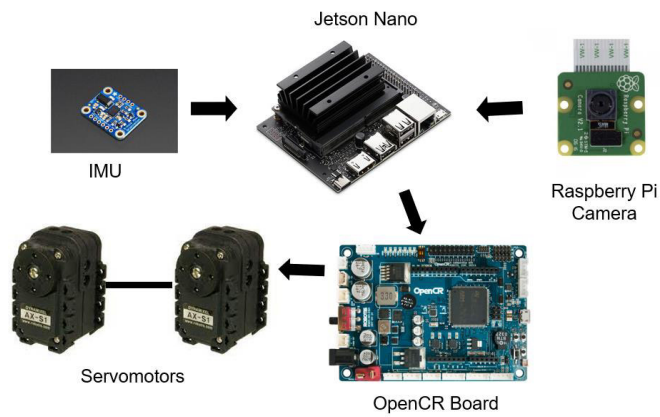


Figure 2.9. Hardware setting schematic.

(Source: Own)

2.2.3.2. Structural module solution principles and evaluation

Table 2.14 shows the solutions and alternatives for the structural module.

Table 2.14. Solution alternatives for the structural module.

Function	Solution for functions
Hold and protect processor	<pre> graph TD A(Protective case) --> B(Bullet aerodynamic profile) A --> C(Sphere aerodynamic profile) B --> D(Anti-Vibration Plate) C --> D </pre>
Hold and protect camera and IMU sensor	
Avoid UAV vibrations	

Move camera	
Alternative solutions	

(Source: Own)

Detailed solutions for each function are shown below.

Hold and protect processor:

The boards used to control the gimbal will be located inside the fuselage of the UAV to protect them. Only one solution is presented for this function given the dimensions of the UAV fuselage consisting of a case as shown in Figure 2.10.

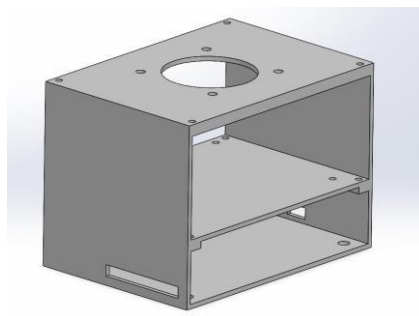


Figure 2.10. Case.


(Source: Own)

Hold and protect camera and IMU:

As solutions for this function, 2 aerodynamic profiles were chosen with a low drag coefficient that will not affect the aerodynamics of the UAV. Their advantages and disadvantages are shown in Table 2.15.

Table 2.15. Solutions to “Hold and protect camera and IMU”.

Solutions	Advantages	Disadvantages	Figure
Bullet aerodynamic profile	Its manufacture allows it to easily place the camera and the IMU sensor inside.	Its drag coefficient is 0.295, greater than a small sphere.	

<p align="center">Sphere aerodynamic profile</p>	<p>For a small sphere its drag coefficient is 0.07.</p>	<p>Its manufacture is complicated to place the camera and the IMU sensor inside.</p>	
---	---	--	---

(Source: [62])

Avoid UAV vibrations:

All gimbals use an anti-vibration plate with rubber dampers to avoid UAV vibrations; therefore, it is the only solution that is presented for this function as shown in Figure 2.11.

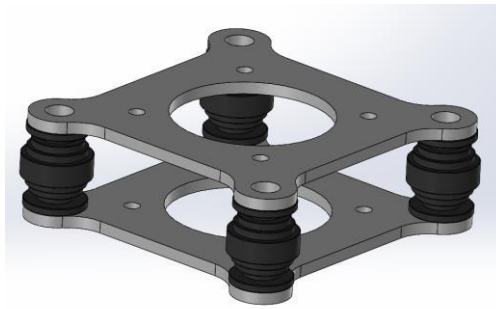


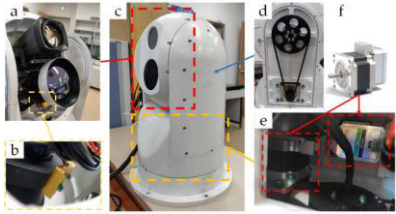
Figure 2.11. Gimbal anti-vibration plate.

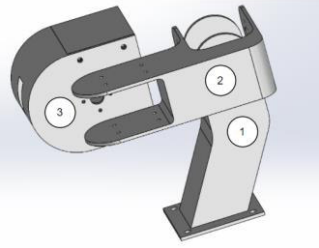
(Source: Own)

Move camera:

Two mechanisms that use gimbals to move their camera are shown in Table 2.16.

Table 2.16. Solutions to “Move camera”.

Solutions	Advantages	Disadvantages	Figure
<p align="center">Belt and pulley mechanism</p>	<p>It is safer because the motors are placed inside the gimbal structure.</p>	<p>It is heavier because it uses more mechanical elements. This type of mechanism limits the range of the roll rotation angle.</p>	

<p>Three link mechanism</p>	<p>It is compact, it is light, it is easy to manufacture, it allows a greater range of rotation in roll and pitch.</p>	<p>It is not safe enough because the structure does not protect the motors.</p>	
------------------------------------	--	---	---

(Source: [63])

To choose an alternative solution, the following criteria are formulated:

- Have a wide-angle rotation range since it will allow a better visualization during flight tests.
- The structure must be safe since the electronic components will be placed there.
- Have a low weight due to the maximum payload of the UAV.
- Easy manufacturing since it will save time for the assembly of the prototype.

Using the corrected weighted criteria ordinal method, we proceed to evaluate each criterion and select the best alternative solution for the structural module as shown in the following tables:

Table 2.17. Evaluation of the specific weight of each criterion.

Wide angle rotation range > Safe > Low weight > Easy manufacturing						
Criterion	Wide angle rotation range	Safe	Low weight	Easy manufacturing	$\sum +1$	Weight
Wide angle rotation range		1	1	1	4	0.40
Safe	0		1	1	3	0.30
Low weight	0	0		1	2	0.20
Easy manufacturing	0	0	0		1	0.10
Total					10	1.00

(Source: Own)

Table 2.18. Evaluation of the specific weight of the “Wide angle rotation range” criterion.

Alternative 2 = Alternative 3 > Alternative 1					
Wide angle rotation range	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		0	0	1	0.16
Alternative 2	1		0.5	2.5	0.42
Alternative 3	1	0.5		2.5	0.42
Total				6	1.00

(Source: Own)

Table 2.19. Evaluation of the specific weight of the “Safe” criterion.

Alternative 1 > Alternative 2 = Alternative 3					
Safe	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		1	1	3	0.50
Alternative 2	0		0.5	1.5	0.25
Alternative 3	0	0.5		1.5	0.25
Total				6	1.00

(Source: Own)

Table 2.20. Evaluation of the specific weight of the “Low weight” criterion.

Alternative 2 = Alternative 3 > Alternative 1					
Low weight	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		0	0	1	0.16
Alternative 2	1		0.5	2.5	0.42
Alternative 3	1	0.5		2.5	0.42
Total				6	1.00

(Source: Own)

Table 2.21. Evaluation of the specific weight of the “Easy manufacturing” criterion.

Alternative 2 > Alternative 3 > Alternative 1					
Compact	Alternative 1	Alternative 2	Alternative 3	$\sum +1$	Weight
Alternative 1		0	0	1	0.17
Alternative 2	1		1	3	0.50
Alternative 3	1	0		2	0.33
Total				6	1.00

(Source: Own)

Table 2.22. Evaluation of conclusions of the structural module.

Conclusions	Wide angle rotation range	Safe	Low weight	Easy manufacturing	Σ	Priority
Alternative 1	0.40*0.16	0.30*0.50	0.20*0.16	0.10*0.17	0.263	3
Alternative 2	0.40*0.42	0.30*0.25	0.20*0.42	0.10*0.50	0.377	1
Alternative 3	0.40*0.42	0.30*0.25	0.20*0.42	0.10*0.33	0.36	2

(Source: Own)

According to the results obtained in Table 2.22, Alternative 2 is determined as a priority for the mechanical module. Figure 2.12 shows a CAD schematic of Alternative 2.

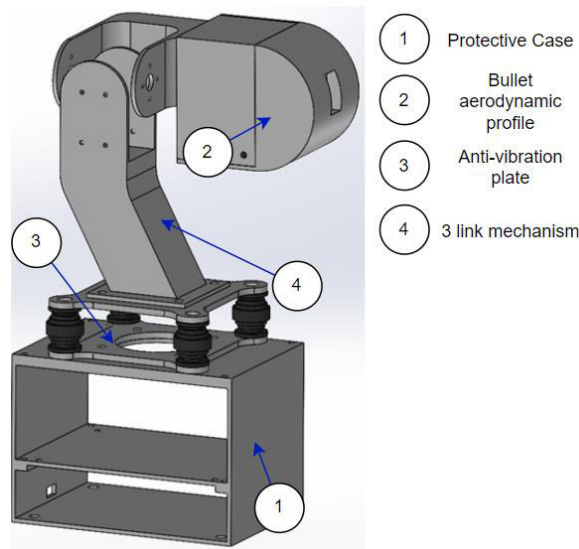


Figure 2.12. Structural module schematic.

(Source: Own)

2.2.3.3. Morphological diagram of the solution for the gimbal.

The final solution obtained is shown in Table 2.23. Figure 2.13 shows a CAD model assembling the control and structural modules.

Table 2.23. Selected solutions for each gimbal module.

Modules	Alternatives
Control Module	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid green; padding: 2px;">Alternative 1</div> <div style="padding: 2px;">Alternative 2</div> <div style="padding: 2px;">Alternative 3</div> </div>
Structural Module	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="padding: 2px;">Alternative 1</div> <div style="border: 1px solid red; padding: 2px;">Alternative 2</div> <div style="padding: 2px;">Alternative 3</div> </div>

(Source: Own)

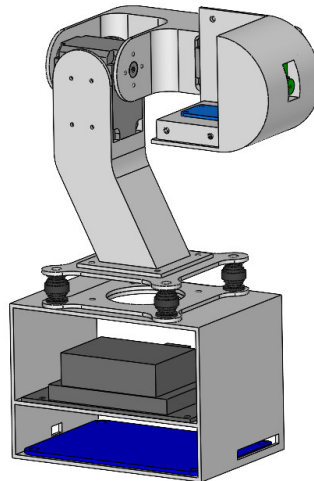


Figure 2.13. Gimbal concept model.

(Source: Own)

2.3. Materialization Design

This stage aims to develop the Computer Vision algorithm of the gimbal, select its hardware components, select its material, and establish the dimensions of its mechanical components to translate them into an assembly drawing.

2.3.1. Computer Vision algorithm programming

The methodology of the Computer Vision stabilization algorithm shown in Figure 2.14 is based on detecting the sky line from an outdoor image dataset and tracking it to compensate the roll rotation angle, which is the one that presents the greatest instability during a UAV flight. The algorithm will be programmed using Python language, and it will be simulated for a video (image dataset) captured in the Antisana Ecological Reserve by a GoPro camera without stabilization placed on a fixed-wing UAV of the ATA research group as shown in Figure 2.15.

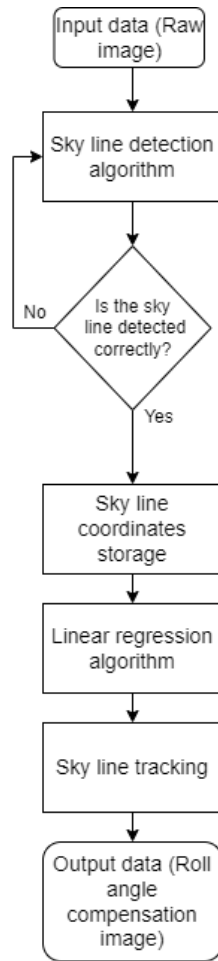


Figure 2.14. Computer Vision algorithm methodology.
(Source: Own)

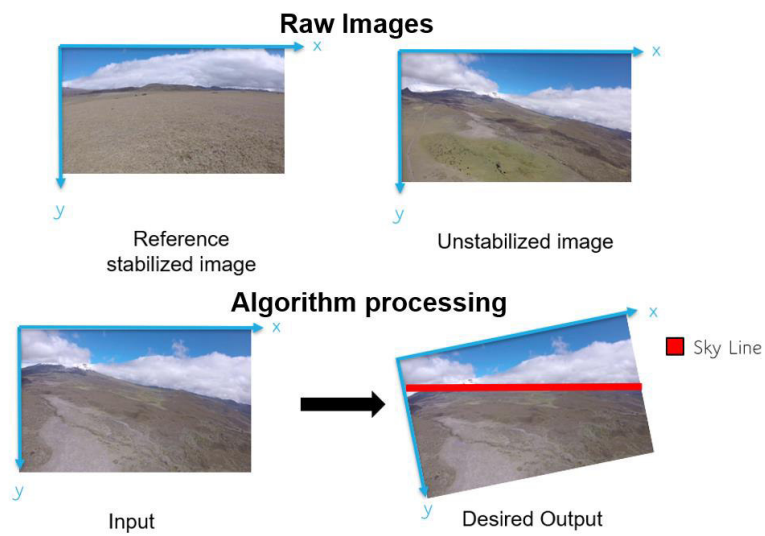


Figure 2.15. Input and desired output of the stabilization algorithm.
(Source: Own)

2.3.1.1. Sky line detection algorithm

Two options are proposed for this algorithm:

1. Feature detection using OpenCV libraries such as SIFT, SURF, ORB, and Canny Edge Detection.
2. Image Segmentation using the U-Net Convolutional Neural Network.

Feature detection algorithm using OpenCV libraries

Figure 2.16 shows the results applying the SIFT, SURF and ORB libraries to detect the sky line. SIFT and SURF are very noisy; ORB on the other hand identifies few features. The programmed code is shown in Appendix I, at the end of this thesis.

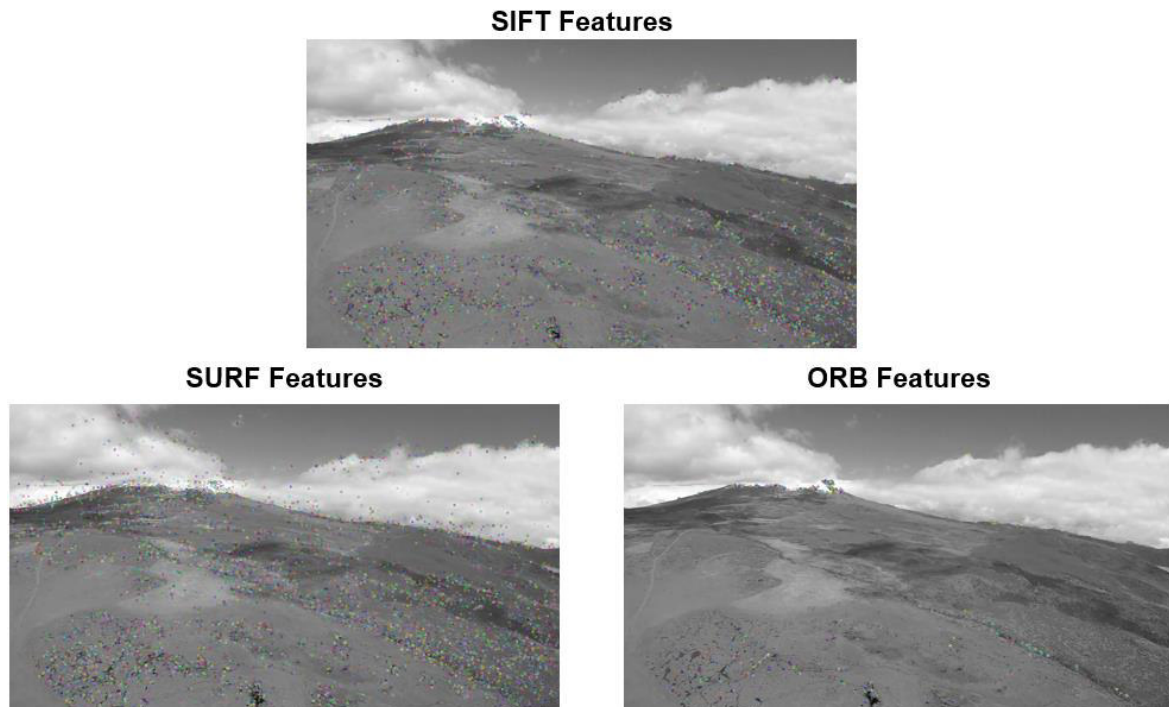


Figure 2.16. SIFT, SURF, and ORB features.

(Source: Own)

Using the Canny Edge Detection library with different filters, the results shown in Figure 2.17 are obtained. The Blur filter and the Bilateral filter present a better distribution of the edges; however, the line of the sky still cannot be isolated. The programmed code is shown in Appendix II, at the end of this thesis.

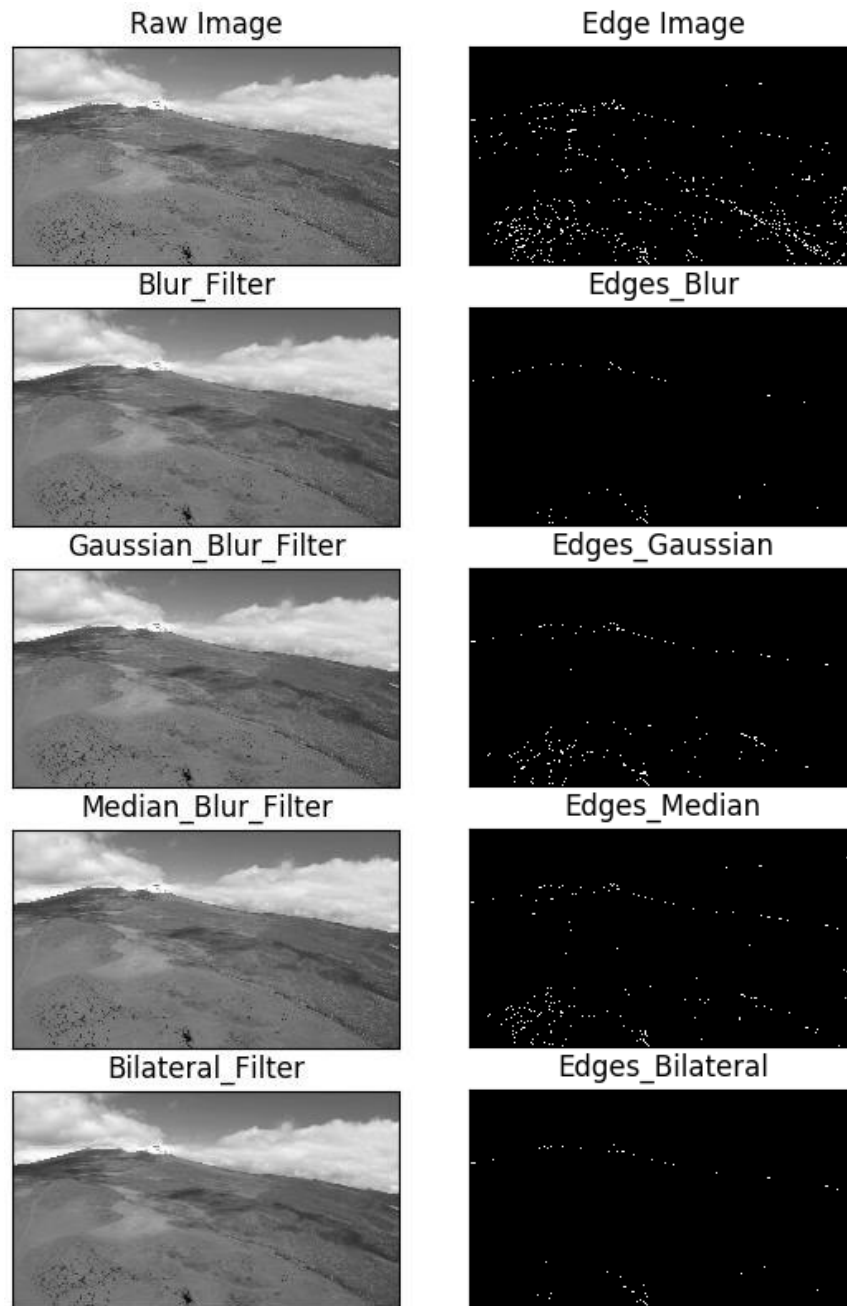


Figure 2.17. Canny Edge Detection library.

(Source: Own)

Given that the Canny Edge Detection library takes RGB images as input data to transform them into grayscale, it was decided to first transform the RGB images into HSV to increase the color intensity of the image to be able to better detect the sky line. The results applying different filters are shown in Figure 2.18. The Median Blur filter was able to isolate the sky line, however, the library also detects the edges of the clouds.

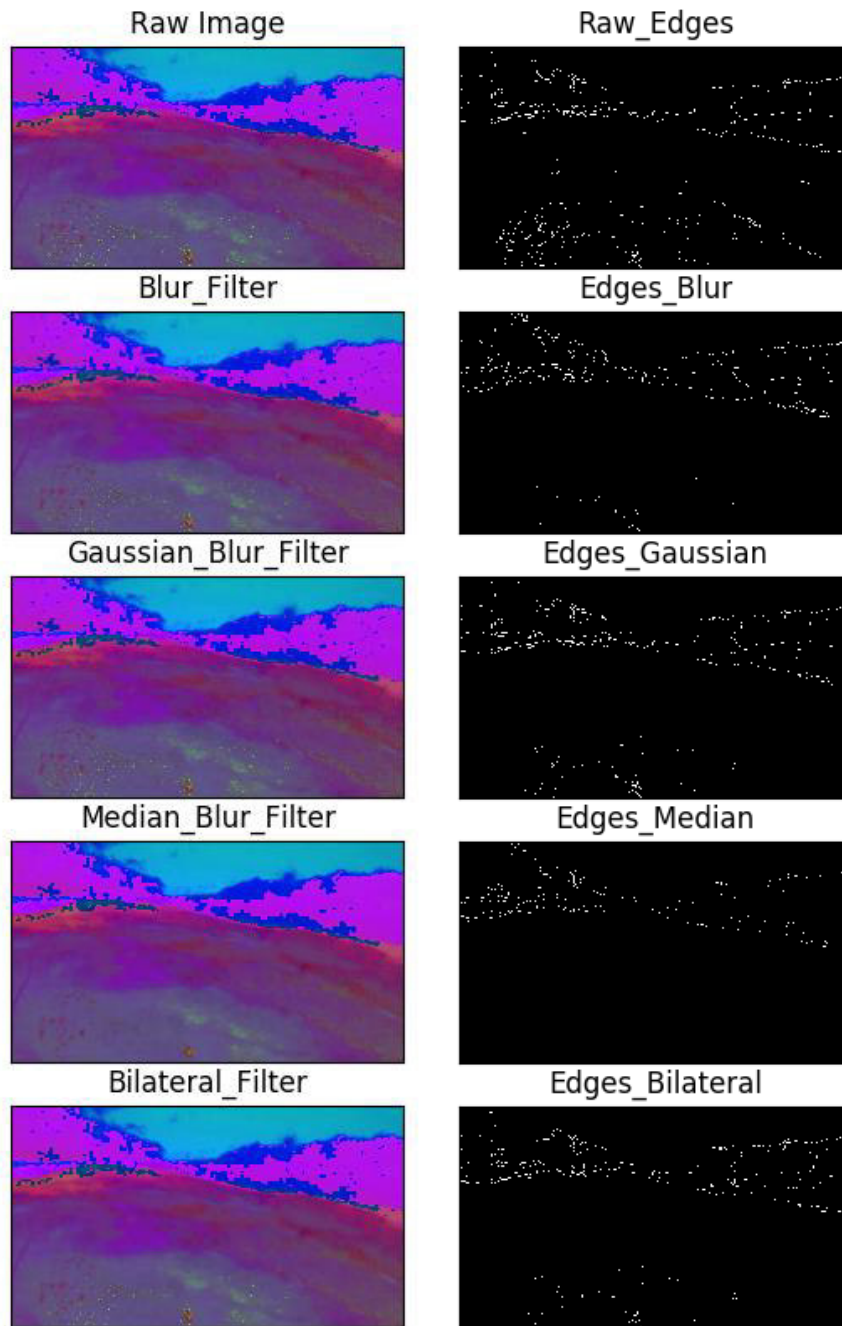


Figure 2.18. Canny Edge Detection library using HSV images.

(Source: Own)

Image Segmentation using the U-Net Convolutional Neural Network

To train the neural network, the Skyfinder Dataset [64] is used, which has 90000 labeled outdoor images taken at different times and seasons of the year. The dataset has 50 different scenarios with their respective masks that segment the sky and the ground, as shown in Figure 2.19.



Figure 2.19. Skyfinder Dataset.

(Source: [64])

The open source neural network was obtained from GitHub [65], however the code used to load the Skyfinder dataset can be seen in Appendix III.

To train the dataset, the images with the best resolution were chosen, as well as those with a scenario like the Antisana Ecological Reserve. 47 scenarios with 50 images each are trained, for a total of 2350 images with the following hyperparameters set by default:

- Activation Function: ReLU
- Loss function = Cross-Entropy
- Epochs = 5
- Batch size = 1

$$\text{Number of Steps} = \frac{47 \text{ scenarios}}{1} = 47 \text{ steps}$$

$$\text{Number of iterations} = \frac{5 \text{ epochs}}{1} = 5 \text{ iterations}$$

- Learning rate = 0.001
- Scale = 0.5 (Downscaling factor of the images)
- Validation size = 0.1 (Percentage of data used as validation)

The training code is run for approximately 1 hour as shown in Figure 2.20 obtaining a Pytorch model which is used in the prediction code.

```
Run summary:
train loss 0.21262
step 215
epoch 4
runtime 3711
timestamp 1647433700
_step 267
learning rate 0.0
validation Dice 0.95064
```

Figure 2.20. Training results.

(Source: Own)

The result shown in Figure 2.21 is obtained by the prediction code. The sky and the ground have been segmented allowing the sky line to be easily identified, therefore this option is chosen to meet this objective.

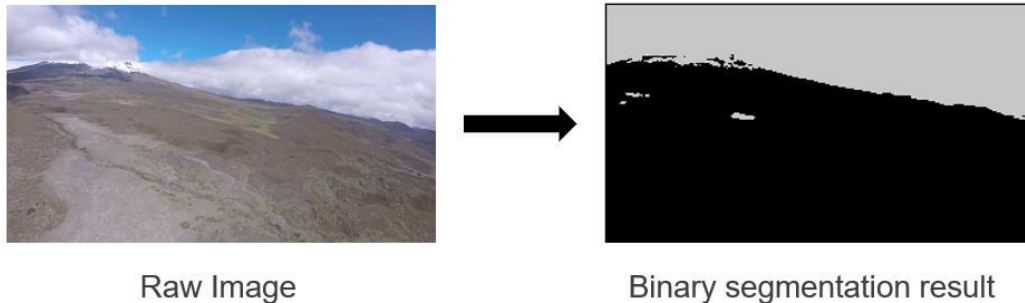


Figure 2.21. Binary segmentation prediction result.

(Source: Own)

2.3.1.2. Sky line coordinates storage

Since the image neural network results have certain errors to completely segment the sky from the ground, the Canny Edge Detection library is not used. Therefore, to store the coordinates of the sky line, a for loop is used to iterate over the pixels of the image as shown in Figure 2.22. If a black pixel appears (Intensity = 0) a point coordinate is stored.

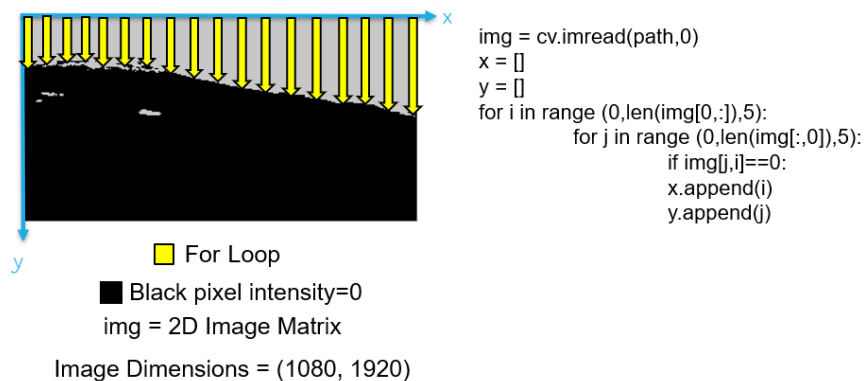
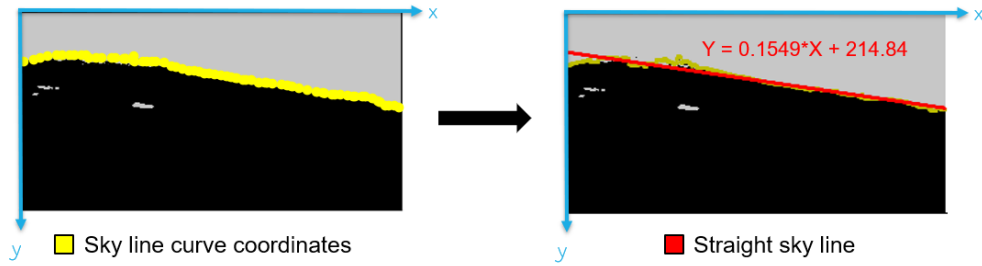


Figure 2.22. Sky line coordinates storage.

(Source: Own)

2.3.1.3. Linear regression algorithm

Since the sky line curve is not uniform and for ease of calculation, linear regression equations (1.18), (1.19), and (1.20) are applied, which will allow establishing a straight sky line for a better tracking of it as shown in Figure 2.23.



```

xbar = sum(x)/len(x)
ybar = sum(y)/len(y)
n = len(x) # or len(y)

numer = sum([xi*yi for xi,yi in zip(x, y)]) - n * xbar * ybar
denum = sum([xi**2 for xi in x]) - n * xbar**2

a = numer / denum
b = ybar - a * xbar

```

Figure 2.23. Straight sky line result.

(Source: Own)

2.3.1.4. Sky line tracking

To track the sky line, the angle of the sky line of a stabilized image is taken as reference, as shown in Figure 2.24. For a second image moved from the reference position, its angle of movement is also calculated as shown in equation (2.1). Finally, the angle that the second image must rotate to compensate for its movement will be the subtraction of the angle of the moved image minus the angle of the reference image, as shown in equation (2.2).

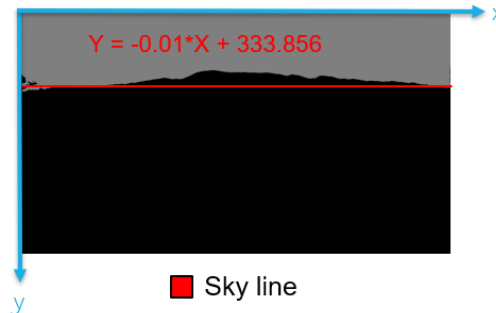


Figure 2.24. Reference sky line result.

(Source: Own)

$$\theta_{movement} = \tan^{-1} a \quad (2.1)$$

Where:

$\theta_{movement}$ Movement rotation angle [rad]

a Slope of the sky line

$$\theta_{compensation} = -(\theta_{movement} - \theta_{reference}) \quad (2.2)$$

Where:

$\theta_{compensation}$ Compensation rotation angle [rad]

$\theta_{reference}$ Reference angle [rad]

Taking the example of Figure 2.23 as a moving image, and Figure 2.24 as a reference image, their angles of rotation with respect to the x-axis are calculated as follows:

$$\theta_{movement} = \tan^{-1} 0.1549 = 0.1537 \text{ rad}$$

$$\theta_{reference} = \tan^{-1} -0.01 = -0,01 \text{ rad}$$

Once these angles have been obtained, the compensation angle that the image must rotate to stabilize itself with respect to a reference is calculated:

$$\theta_{compensation} = -(0.1537 \text{ rad} - (-0.01 \text{ rad})) = -0.1637 \text{ rad} = -9.38^\circ$$

For this example, the image must be rotated 9.38° counterclockwise.

To visualize the simulation result, the `lmutils` library is used, which applies a rotation matrix around the z axis as shown in equation (2.3). The final code for this algorithm is shown in Appendix IV at the end of this thesis.

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_{compensation} & -\sin \theta_{compensation} & 0 \\ \sin \theta_{compensation} & \cos \theta_{compensation} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.3)$$

Where:

(x, y) Raw image coordinates

(x_2, y_2) Compensated image coordinates

For the previously calculated example the following is obtained:

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos -9.38^\circ & -\sin -9.38^\circ & 0 \\ \sin -9.38^\circ & \cos -9.38^\circ & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The result is shown in Figure 2.25.

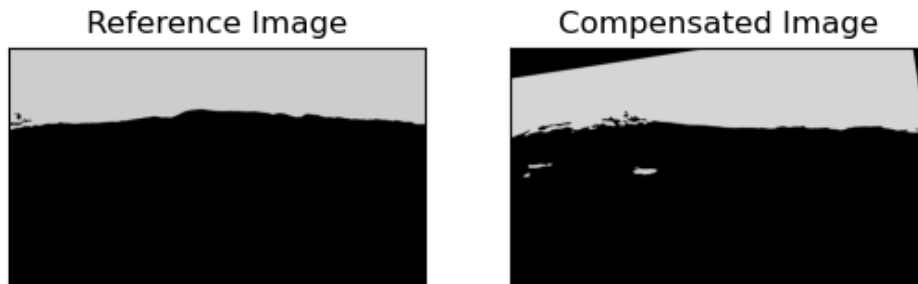


Figure 2.25. Compensated image result.

(Source: Own)

2.3.2. Hardware component selection

Since most of the hardware components were already chosen in the previous design stage, the main technical specifications of each one is shown in Table 2.24. In addition, in this section, two servomotors will be chosen to rotate the gimbal camera in the pitch and roll angle.

Table 2.24. Technical specifications of hardware components



Component	Weight [g]	Dimensions [mm]	Voltage [V]	Current [A]
NVIDIA Jetson Nano 2Gb + Active Heat Sink	100	100 x 80 x 29	5	3
OpenCR1.0 controller	60	100 x 75	11.1	1
BNO055 IMU sensor	3	20 x 27 x 4	3.3	0.0125
Raspberry Pi Camera	3	25 x 24 x 9	3.3	0.25

(Source: Own)

2.3.2.1. Servo motor selection

Robotis Dynamixel servomotors are widely used in the design of robots in universities and research laboratories in the world due to their great power and efficiency in the market. They are compatible with many programming languages, allowing feedback and control of their position, speed, torque, etc. The XL series is the smallest group of Dynamixel servo motors which are used for rapid prototyping such as 3D printing and small robots [66]. Table 2.25 shows the technical specifications of the two servomotors belonging to this group.

Table 2.25. Technical specifications of the Dynamixel XL series servo motors.

Dynamixel XL430-W250 servo motor	
Maximum Torque: 14 kg.cm / 1.4 N.m	
Maximum Speed: 61 rpm / 6.4 rad/s	
Input voltage: 11.1 V	
Current: 1.3 A	
Weight: 57.2 g	
Dimensions : 28.5 mm x 46.5 mm x 34 mm	
Degree of precision: 0.088°	
Dynamixel XL-320 servo motor	
Maximum Torque: 3.98 kg.cm / 0.39 N.m	
Maximum Speed: 114 rpm / 11.94 rad/s	
Input voltage: 7.4 V	
Current: 1.1 A	
Weight: 16.7 g	
Dimensions : 24 mm x 36 mm x 27 mm	
Degree of precision: 0.29°	

(Source: [67] [68])

To establish the minimum torque that the roll rotation servomotor must supply, as shown in equation (2.4), a static analysis is developed assuming a 45 mm link with a punctual force of 0.0632 Kgf corresponding to the sum of the weight of the IMU sensor, the Raspberry Pi camera, and the heaviest servo motor placed at the end of the link, as shown in Figure 2.26.

$$T = r \cdot F \quad (2.4)$$

Where:

T Torque [Kgf · cm]

r Distance perpendicular to the force [cm]

F Force [Kgf]

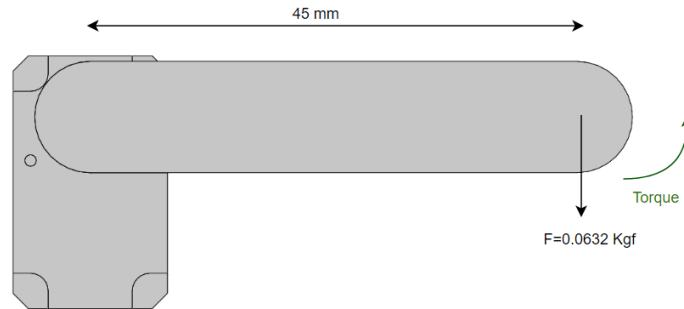


Figure 2.26. Servo motor torque.

(Source: Own)

$$T = 4.5 \text{ cm} \cdot 0.0632 \text{ Kgf} = 0.284 \text{ Kgf} \cdot \text{cm}$$

The roll rotation servomotor must supply a torque greater than 0.284 Kgf.cm to move the gimbal mechanism.

For the minimum speed that the servo motors must supply, an average value of angular speed of 4 rad/s for rotation in pitch and roll is taken into consideration, which was obtained by the IMU sensor of the UAV during flight tests.

Both motors meet the minimum technical specifications to move the gimbal mechanism, however, since it is a stabilization problem, the servo motor must provide accurate angle data. Therefore, the XL430-W250 servo motor is chosen for its precision degree of 0.088°.

2.3.2.2. Power supply sizing

To size the power supply for the gimbal, Watt's law is used as shown in equation (2.5) to calculate the powers of each of the hardware components as shown in Table 2.26.

$$P = I \cdot V \tag{2.5}$$

Where:

P Power [W]

I Current [A]

V Voltage [V]

Table 2.26. Power of each hardware component.

Components	Voltage [V]	Current [A]	Power [W]
NVIDIA Jetson Nano	5	3	15
OpenCR1.0 controller	11.1	1	11.1
BNO055 IMU sensor	3.3	0.0125	0.04125
Raspberry Pi Camera	3.3	0.25	0.825
2 Servo motors	11.1	1.3	14.43
Total			41.39625

(Source: Own)

Subsequently, the electrical current of all the hardware is calculated by dividing the sum of the power of all the components by the maximum voltage as shown in equation (2.6). In this case, the maximum voltage is 11.1 V, which corresponds to the servomotors and the OpenCR board. Finally, the battery capacity is calculated by equation (2.7).

$$I_{hardware} = \frac{P_{total}}{V_{maximum}} \quad (2.6)$$

$$C_{battery} = I_{hardware} \cdot t \quad (2.7)$$

Where:

$C_{battery}$ Battery capacity [Ah]

$I_{hardware}$ Current of all the hardware [A]

t Discharge time [h]

The capacity of the battery is calculated for a discharge time of 30 minutes, which is the flight test time of the UAV.

$$I_{hardware} = \frac{41.39625 \text{ W}}{11.1 \text{ V}} = 3.73 \text{ A}$$

$$C_{battery} = 3.73 \text{ A} \cdot 30 \text{ min} \cdot \frac{1 \text{ h}}{60 \text{ min}} = 1.865 \text{ Ah}$$

To supply power to the gimbal hardware it is necessary to have a battery that supplies a minimum of 11.1 V and 1865 mAh. The UAV battery supplies 14.8 V and 8000 mAh. Therefore, there is no need to use an additional battery.

2.3.3. Structural module dimensioning

2.3.3.1. Materials selection

For the material selection of the mechanical components of the gimbal, 4 important characteristics are evaluated:

- **Density:** It influences the weight of the gimbal.
- **Elastic modulus:** It influences the strength of the material.
- **Availability in the country:** The materials must be easy to obtain in Ecuador.
- **Ease of machining:** This allows easy manufacturing of replacement parts.

The score is shown in Table 2.27.

Table 2.27. Characteristics Score.

Score	Meaning
0	If the characteristic is considered low or bad
1	If the characteristic is considered acceptable
2	If the characteristic is considered good or high

(Source: Own)

Table 2.28 shows the mechanical properties of materials proposed for the manufacture of the gimbal.

Table 2.28. Mechanical properties of gimbal materials.

Material	Density [Kg/m ³]	Elastic modulus [GPa]
PLA plastic	1.27 x 10 ³	3.58
ABS plastic	1.06 x 10 ³	2.75
Aluminum	2.7 x 10 ³	81
Balsa wood	220	0.19
Carbon fiber	1.8 x 10 ³	225

(Source: Own)

Table 2.29 shows the results of the evaluation for each characteristic of the material.

Table 2.29. Material selection results.

Material	Density	Elastic modulus	Availability	Machining	Σ
PLA plastic	1	1	2	2	6
ABS plastic	1	1	2	2	6
Aluminum	0	2	2	0	4
Balsa wood	2	0	2	0	4
Carbon fiber	1	2	0	0	3

(Source: Own)

According to the results obtained, the ABS option is chosen because of its density and parts with complex geometry can be manufactured through 3D printing.

2.3.3.2. Bullet aerodynamic profile dimensioning

The aerodynamic profile is designed to protect the IMU sensor and camera. Therefore, the dimensions of its cross-sectional area resemble the dimensions of the Raspberry Pi camera. The length of the base of the profile is designed for the dimensions of the IMU sensor BNO055. The profile is directly attached to the servo motor and has a cover to be able to assemble and disassemble the electronic components with an opening for the cables as shown in Figure 2.27.

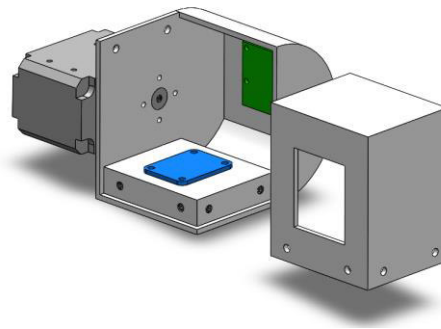


Figure 2.27. Bullet aerodynamic profile parts.

(Source: Own)

The diameters of the fastening screws of the servomotor, camera and IMU sensor are determined by the manufacturer, however aluminum alloy T6 screws are chosen since it is not a ferromagnetic material that can influence the operation of the magnetometer of the IMU sensor. To calculate diameters of the fastening screws of the profile cover, a shear stress analysis shown in equation (2.8) is carried out for a critical position at 90° as shown in Figure 2.28. The shear force is the weight of the cover (0.017 Kgf), which is obtained using the 3D printing software Ultimaker Cura with default settings.

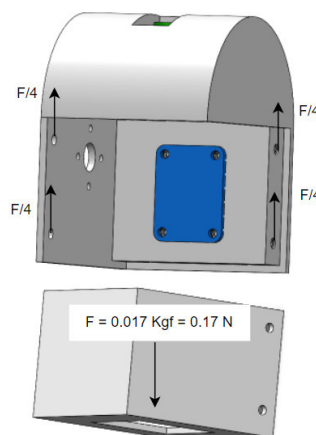


Figure 2.28. Bullet aerodynamic profile shear stress analysis.

(Source: Own)

$$\tau_{shear} = \frac{F}{4A} \quad (2.8)$$

Where:

τ_{shear} Shear stress [MPa]

F Shear [N]

A Cross – sectional area [mm^2]; screw cross – sectional area is shown in equation (2.9).

$$A = \frac{\pi \cdot D^2}{4} \quad (2.9)$$

Where:

D Screw diameter [mm]

Considering a safety factor of $n = 1.5$ for small unmanned aircrafts [69], the allowable shear stress is calculated as shown in equation (2.10).

$$n = \frac{0.577S_y}{\tau_{shear}} \quad (2.10)$$

Where:

S_y Yield strenght [MPa]

According to Appendix V, the aluminum alloy T6 yield strength is 169 MPa. With the calculated values, the diameter of the screw is obtained.

$$\tau = \frac{0.577 \cdot 169 \text{ MPa}}{1.5} = 65 \text{ MPa}$$

$$D = \sqrt{\frac{4 \cdot \frac{0.17 \text{ N}}{4}}{\pi \cdot 65 \text{ MPa}}} = 0.03 \text{ mm}$$

Since there are no standardized screws with a diameter of 0.03 mm, a screw with a diameter of 3 mm is selected in accordance with Appendix VI.

For the aerodynamic analysis of the profile, a drag coefficient of 0.295 is used, as shown in Figure 2.29.



Shape Effects on Drag

Glenn
Research
Center

The shape of an object has a very great effect on the amount of drag.

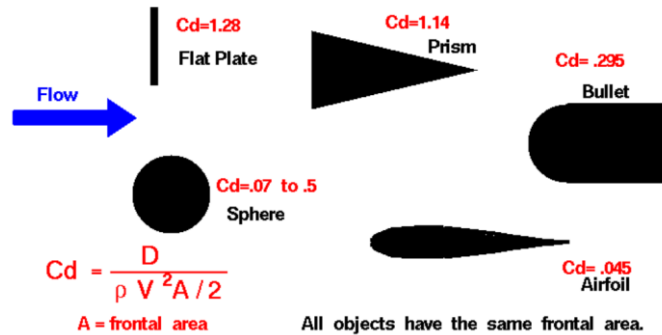


Figure 2.29. Shape effects on Drag.

(Source: [62])

The profile drag force is calculated as shown in equation (2.11).

$$D = C_d \cdot \rho \cdot V^2 \cdot \frac{A}{2} \quad (2.11)$$

Where:

D Drag Force [N]

C_d Drag coefficient

ρ Air density [Kg/m^3]

V UAV Velocity [m/s^2]

A Frontal Area [m^2]

According to the Wind Atlas of Ecuador shown in Appendix VII, the Antisana Ecological Reserve is located approximately 3800 meters above sea level, with an air density of $0.837 \text{ Kg}/\text{m}^3$. The speed of the UAV during flight tests is $12 \text{ m}/\text{s}$. The size of the frontal area of the aerodynamic profile of the bullet is a rectangle of $50 \times 60 \text{ mm}$.

$$D = 0.295 \cdot 0.837 \frac{\text{Kg}}{\text{m}^3} \cdot \left(12 \frac{\text{m}}{\text{s}}\right)^2 \cdot \frac{0.05 \text{ m} \cdot 0.06 \text{ m}}{2} = 0.053 \text{ N}$$

The drag force of the aerodynamic profile is very small compared to the aerodynamic forces of the UAV. Therefore, the gimbal will not affect the aerodynamics of the UAV.

2.3.3.3. Mechanism dimensioning

The lengths of the mechanism are dimensioned according to the volume of the fuselage and a stress analysis of the rolling rotation link is carried out by applying a punctual force of 0.055 Kgf, which is the weight of the aerodynamic profile link with its electronic components and a punctual force of 0.057 Kgf, which is the weight of the servo motor to determine whether T6 aluminum screws with a diameter of 2 mm can be applied, as shown in Figure 2.30.

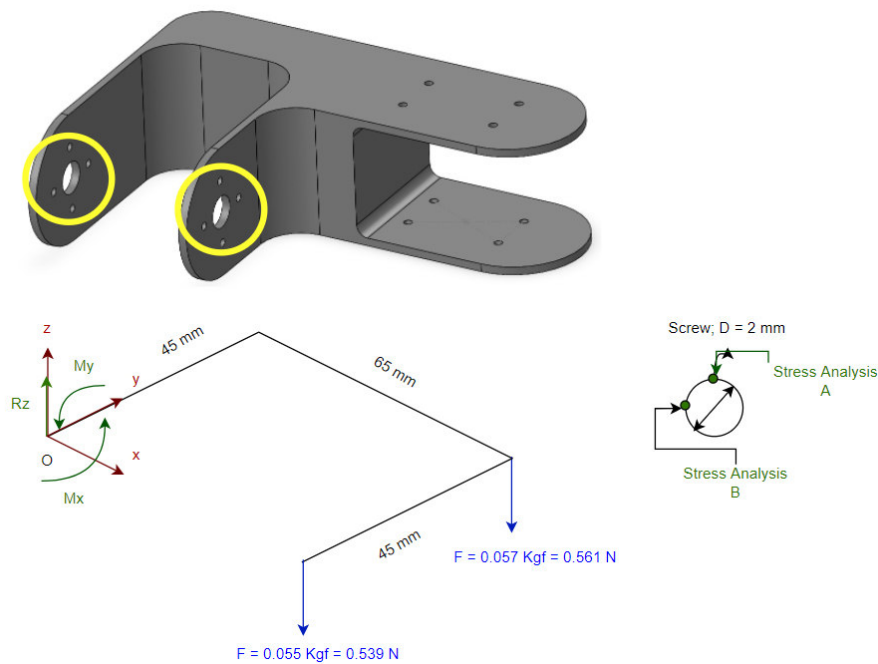


Figure 2.30. Static analysis of link 2 of the mechanism.

(Source: Own)

The reactions at point O are calculated.

$$\sum F_z = 0$$

$$R_z = 0.539 \text{ N} + 0.561 \text{ N} = 1.1 \text{ N}; \text{ For 1 screw: } \frac{R_z}{8} = 0.138 \text{ N}$$

$$\sum T_x = 0$$

$$M_x = 0.561 \text{ N} \cdot 45 \text{ mm} = 25.245 \text{ N} \cdot \text{mm}; \text{ For 1 screw: } \frac{M_x}{8} = 3.156 \text{ N} \cdot \text{mm}$$

$$\sum T_y = 0$$

$$M_y = 0.561 \text{ N} \cdot 65 \text{ mm} + 0.539 \text{ N} \cdot 65 \text{ mm} = 71.5 \text{ N} \cdot \text{mm};$$

$$\text{For 1 screw: } \frac{M_y}{8} = 8.938 \text{ N} \cdot \text{mm}$$

Considering a safety factor of $n = 1.5$ and aluminum as material, the equivalent stress using the Von-Mises criterion is calculated as shown in equation (2.12).

$$n = \frac{S_y}{\sigma_{eq}} \quad (2.12)$$

Where:

σ_{eq} Equivalent stress [MPa]

$$\sigma_{eq} = \frac{169 \text{ MPa}}{1.5} = 112.67 \text{ MPa}$$

The maximum shear stress for circular shapes in bending shown in equation (2.13), the bending stress shown in equation (2.14), and the torsional stress shown in equation (2.16) for the screw are calculated.

$$\tau_{shear-maximum_circular} = \frac{4F}{3A} \quad (2.13)$$

Where:

$\tau_{shear-maximum}$ Shear stress due to bending [MPa]

F Shear [N]

A Cross – sectional area [mm^2]

$$\sigma_{bending} = \frac{M \cdot c}{I} \quad (2.14)$$

Where:

$\sigma_{bending}$ Bending stress [MPa]

M Bending moment [$\text{N} \cdot \text{mm}$]

c Vertical distance away from the neural axis [mm]

I Moment of inertia around the neural axis [mm^4], screw inertia is shown in equation (2.15).

$$I = \frac{\pi \cdot D^4}{64} \quad (2.15)$$

$$\tau_{torsional} = \frac{T \cdot c}{J} \quad (2.16)$$

Where:

$\tau_{torsion}$ Torsional stress [MPa]

T Torsional moment [N · m]

r Vertical distance away from the neural axis [mm]

J Polar moment of inertia [mm⁴], screw polar moment of inertia is shown in equation (2.17).

$$J = \frac{\pi \cdot D^4}{32} \quad (2.17)$$

Substituting the values into the equation for points A and B considering a screw diameter of 2 mm we have:

$$\sigma_{bending_A} = \frac{8.938 \text{ N} \cdot \text{mm} \cdot 1 \text{ mm}}{\frac{\pi \cdot (2 \text{ mm})^4}{64}} = 11.38 \text{ MPa}$$

$$\tau_{torsional_A} = \tau_{torsional_B} = \frac{3.156 \text{ N} \cdot \text{mm} \cdot 1 \text{ mm}}{\frac{\pi \cdot (2 \text{ mm})^4}{32}} = 2.009 \text{ MPa}$$

$$\tau_{shear_B} = -\frac{4 \cdot 0.138 \text{ N}}{3 \cdot \frac{\pi \cdot (2 \text{ mm})^2}{4}} = -0.059 \text{ MPa}$$

The Von-Mises criterion as shown in equation (2.18) is used to evaluate the equivalent stress against the yield strength of T6 aluminum. A material will not fail when its calculated equivalent stress is less than its yield strength.

$$\sigma_{eq} = \sqrt{\sigma^2 + 3\tau^2} \quad (2.18)$$

Where:

σ Total normal stresses [MPa]

τ Total shear stresses [MPa]

Substituting the values into the equation for points A and B we have:

$$\sigma_{eq_A} = \sqrt{(11.38 \text{ MPa})^2 + 3(2.009 \text{ MPa})^2} = 11.9 \text{ MPa}$$

$$\sigma_{eq_B} = \sqrt{(0 \text{ MPa})^2 + 3(2.009 \text{ MPa} - 0.059 \text{ MPa})^2} = 3.377 \text{ MPa}$$

Since σ_{eq_A} and σ_{eq_B} are the closest values to σ_{eq} with a safety factor of 1.5, the 8 aluminum screws of 2 mm diameter of the servomotor support the applied load.

Finally, a stress analysis is carried out to determine whether the minimum thickness of 2 mm of ABS plastic is correct for the manufacture of the mechanism as shown in Figure 2.31.

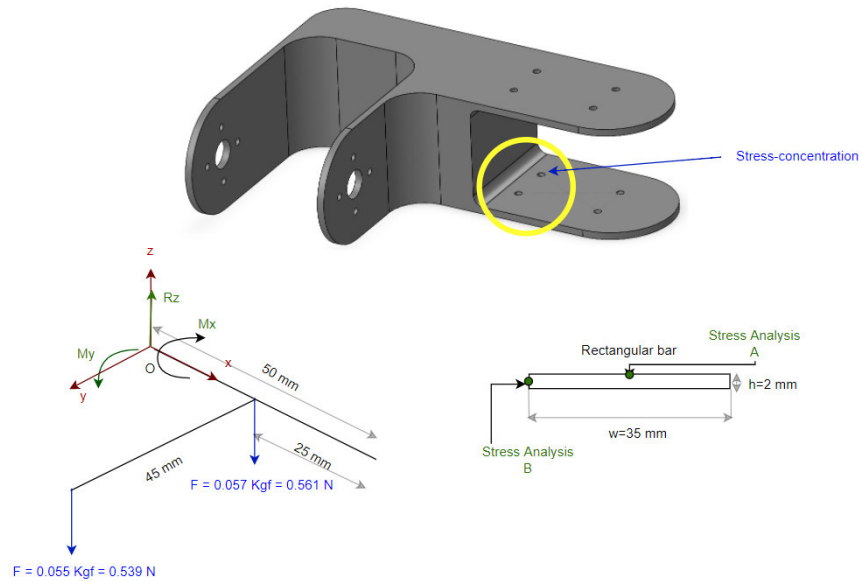


Figure 2.31. ABS plastic static analysis.

(Source: Own)

$$\sum F_z = 0$$

$$R_z = 0.539 N + 0.561 N = 1.1 N$$

$$\sum T_x = 0$$

$$M_x = 0.539 N \cdot 45 mm = 24.255 N \cdot mm$$

$$\sum T_y = 0$$

$$M_y = 0.539 N \cdot 25 mm + 0.561 N \cdot 25 mm = 27.5 N \cdot mm$$

The mechanical properties for 3D printing of ABS plastic are shown in Appendix X.

$$S_{y-ABS\ plastic} = 39 MPa$$

Considering a safety factor of $n = 1.5$ and ABS plastic as material, the equivalent stress using the Von-Mises criterion is calculated.

$$\sigma_{eq} = \frac{39 MPa}{1.5} = 26 MPa$$

The inertia of a rectangle with a transverse hole in bending is shown in equation (2.19).

$$I = \frac{(w-d) \cdot h^3}{12} \quad (2.19)$$

Where:

I Moment of inertia around the neural axis [mm^4]

w Base length [mm]

h Height [mm]

d Hole diameter (Stress – concentration) [mm]

Stress concentration factors for holes in bars in bending are shown in Appendix VIII. For two holes of 2 mm diameter each, we have:

$$\sigma_{bending_A} = \frac{27.5 \text{ N} \cdot \text{mm} \cdot 1 \text{ mm}}{\frac{(35 \text{ mm} - 4 \text{ mm})(2 \text{ mm})^3}{12}} = 1.33 \text{ MPa}$$

$$K_t = 1.8$$

$$\sigma_{bending_A_max} = 1.33 \text{ MPa} \cdot 1.8 = 2.395 \text{ MPa}$$

The maximum shear stress for bending rectangular shapes is shown in equation (2.20).

$$\tau_{shear-maximum_rectangular} = \frac{3F}{2A} \quad (2.20)$$

$$\tau_{shear_B} = -\frac{3 \cdot 1.1 \text{ N}}{2 \cdot 35 \text{ mm} \cdot 2 \text{ mm}} = -0.021 \text{ MPa}$$

The maximum torsional stress for rectangular shapes is shown in equation (2.21).

$$\tau_{torsional-maximum_rectangular} = \frac{T}{c_1 \cdot a \cdot b^3} \quad (2.21)$$

Where:

T Torsional moment [$\text{N} \cdot \text{m}$]

a Base length [mm]

b Height [mm]

c_1 Coefficient for rectangular bars in torsion [mm]. Shown in Appendix IX.

$$\tau_{torsional-A} = \frac{24.255 \text{ N} \cdot \text{mm}}{0.333 \cdot 35 \text{ mm} \cdot (2 \text{ mm})^3} = 0.26 \text{ MPa}$$

$$\tau_{torsional-B} = -0.26 \text{ MPa}$$

Substituting the values into the equivalent stress equation for points A and B we have:

$$\sigma_{eq_A} = \sqrt{(2.395 \text{ MPa})^2 + 3(0.26 \text{ MPa})^2} = 2.437 \text{ MPa}$$

$$\sigma_{eq_B} = \sqrt{3(-0.021 \text{ MPa} - 0.26 \text{ MPa})^2} = 0.487 \text{ MPa}$$

Since σ_{eq_A} and σ_{eq_B} are the closest values to σ_{eq} with a safety factor of 1.5, the 2 mm thickness of the ABS plastic link supports the applied load.

2.3.3.4. Anti-vibration plate bolt dimensioning

The diameter of the bolts to join the gimbal body with the anti-vibration plate is calculated by static analysis. The Von-Mises criterion is applied for the point of maximum stress as shown in Figure 2.32. The load of 0.126 Kgf is the weight of the two links of the mechanism obtained by the 3D printing software plus the weight of the servomotor for roll rotation.

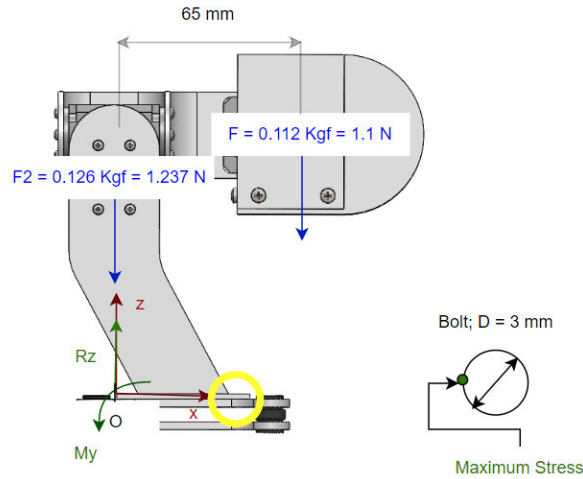


Figure 2.32. Bolts static analysis.

(Source: Own)

$$\sum F_z = 0$$

$$R_z = 1.1 \text{ N} + 1.237 \text{ N} = 2.337 \text{ N}; \text{ For 1 bolt: } \frac{R_z}{4} = 0.584 \text{ N}$$

$$\sum T_y = 0$$

$$M_y = 1.1 \text{ N} \cdot 65 \text{ mm} = 71.5 \text{ N} \cdot \text{mm}; \text{ For 1 bolt: } \frac{M_y}{4} = 17.875 \text{ N} \cdot \text{mm}$$

Considering a safety factor of $n = 1.5$ and aluminum as material, the equivalent stress using the Von-Mises criterion is calculated.

$$\sigma_{eq} = \frac{169 \text{ MPa}}{1.5} = 112.67 \text{ MPa}$$

The axial stress formula is shown in equation (2.22). A minimum standardized aluminum bolt diameter of 3 mm is assumed and substituting the values in the equations we have:

$$\sigma_{axial} = \frac{F}{A} \tag{2.22}$$

Where:

F Axial force [N]

A Area [mm^2]

$$\sigma_{axial} = -\frac{0.584 \text{ N}}{\frac{\pi \cdot (3 \text{ mm})^2}{4}} = -0.083 \text{ MPa}$$

$$\sigma_{bending} = \frac{17.875 \text{ N} \cdot \text{mm} \cdot 1.5 \text{ mm}}{\frac{\pi \cdot (3 \text{ mm})^4}{64}} = 6.743 \text{ MPa}$$

$$\sigma_{eq} = \sqrt{(6.743 \text{ MPa} - 0.083 \text{ MPa})^2} = 6.66 \text{ MPa}$$


Since σ_{eq} calculated assuming a minimum bolt diameter of 3 mm is the closest value to σ_{eq} with a safety factor of 1.5, the 4 aluminum bolts of 3 mm diameter are applied.

Selection of dampers

Anti-vibration plates use plastic ball dampers and are chosen for the weight they will support. The weight of the gimbal is 238 gf and considering a safety factor $n = 1.5$ for small aircrafts, the value of the damper will be 357 gf. Therefore, 4 dampers are chosen that have a lift weight of 100 g each. The technical specifications of the ball damper are shown in Table 2.30.

Table 2.30. Technical specifications of the ball damper.

Technical specifications of the ball damper	
Lift weight: 100 g	
Weight: 1.5 g	

Height: 20 mm	
Width: 19 mm	
Mounting: 10 mm flange or bolt	
Height across plates: 15 mm	

(Source: [70])

2.3.3.5. Protective case dimensioning

The lengths of the protective case for the boards were carried out according to the maximum volume that can be placed inside the fuselage of the UAV. Furthermore, the case does not close to avoid overheating of the boards. A stress analysis is carried out to determine the minimum thickness of ABS plastic that can support the full weight of the gimbal body as shown in Figure 2.33.

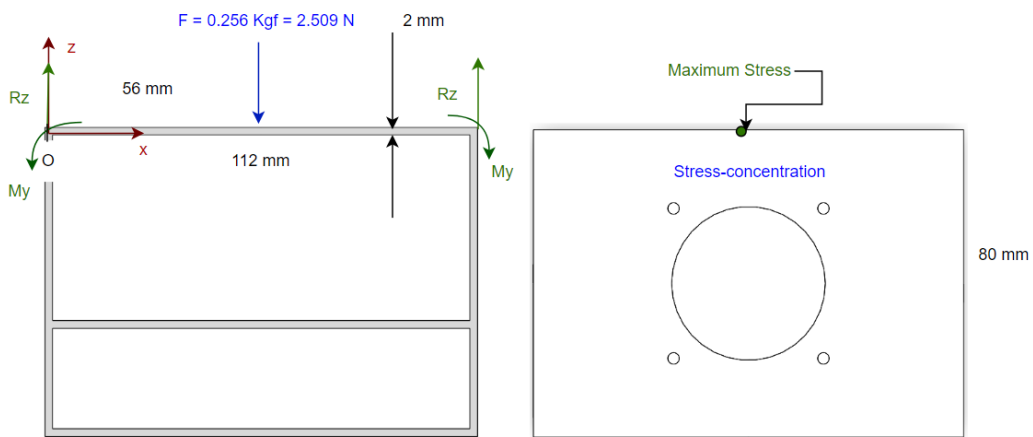


Figure 2.33. Protective case static analysis.

(Source: Own)

To calculate the reactions of a beam fixed at both ends, the equations in Appendix XI are taken and calculated below.

$$R_z = \frac{F}{2} \tag{2.23}$$

$$M_y = \frac{FL}{8} \tag{2.24}$$

Where:

F Force applied to the center of the beam [N]

L Beam length [mm]

$$R_z = \frac{2.509 N}{2} = 1.255 N$$

$$M_y = \frac{2.509 N \cdot 112 mm}{8} = 35.126 N \cdot mm$$

Considering a safety factor of $n = 1.5$ and ABS plastic as material, the equivalent stress using the Von-Mises criterion is calculated.

$$\sigma_{eq} = \frac{39 MPa}{1.5} = 26 MPa$$

For a stress-concentration hole of 46 mm diameter, we have:

$$\sigma_{bending} = \frac{35.126 N \cdot mm \cdot 1 mm}{\frac{(80 mm - 46 mm)(h)^3}{12}} = \frac{12.397}{h^3}$$

$$K_t = 1.3$$

$$\sigma_{max} = \frac{12.397}{h^3} \cdot 1.3 = \frac{16.117}{h^3}$$

Equating the stress equations, the minimum thickness that the material must have is calculated.

$$\sigma_{eq} = \sigma_{max}$$

$$26 = \frac{16.117}{h^3}$$

$$h = 0.85 mm$$

Since it is difficult to manufacture parts with sizes less than 1 mm, a minimum thickness of 2 mm is chosen.

2.4. Detail Design

In this stage, the mechanical components drawings of the gimbal are developed, which can be seen in Appendix XII together with the assembly drawing at the end of this thesis. For this, the technical/mechanical drawing standard CPE INEN 003 is followed.

3. RESULTS AND DISCUSSION

This chapter shows the results achieved for the design and simulation of the gimbal according to the proposed methodology.

3.1. U-net Segmentation Neural Network results

The learning rate of the neural network is shown in Figure 3.1, which indicates that it is good since it converges in a lower value of the loss function ($1e-8$) with a greater number of steps (264).



Figure 3.1. Learning rate curve.

(Source: Own)

The validation accuracy curve as shown in Figure 3.2 tends to the value of 1 as the number of steps increases, being a value of 0.96 at the end of the steps. The loss training curve as shown in Figure 3.3 decreases with training, however it shows peaks that may be due to the small number of epochs and the small number of scenarios used as input images. The network can be optimized by increasing these values, but given the prediction results obtained, it is verified that the network learned.

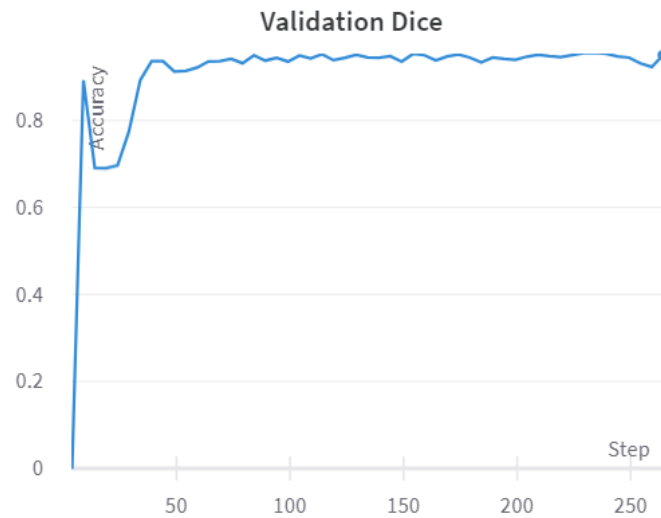


Figure 3.2. Validation Accuracy curve.
(Source: Own)

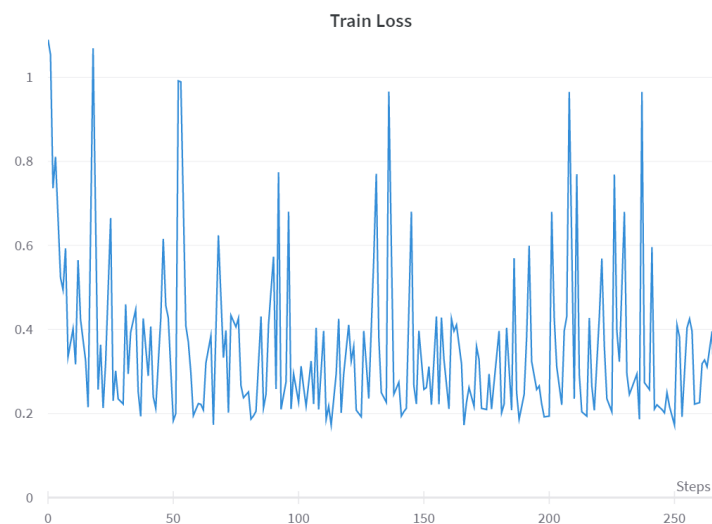


Figure 3.3. Train Loss curve.
(Source: Own)

3.2. Algorithm results and optimization

This section shows the results obtained from the gimbal motion stabilization simulation using Computer Vision.

3.2.1. Boundary edge removal

Certain images do not present a completely segmented sky. Its edges have pixels of intensity 0 affecting the detection of the sky line. That is why the intensity of the first 13 pixels on each side of the image is replaced by their corresponding neighbors obtaining the result as shown in Figure 3.4.

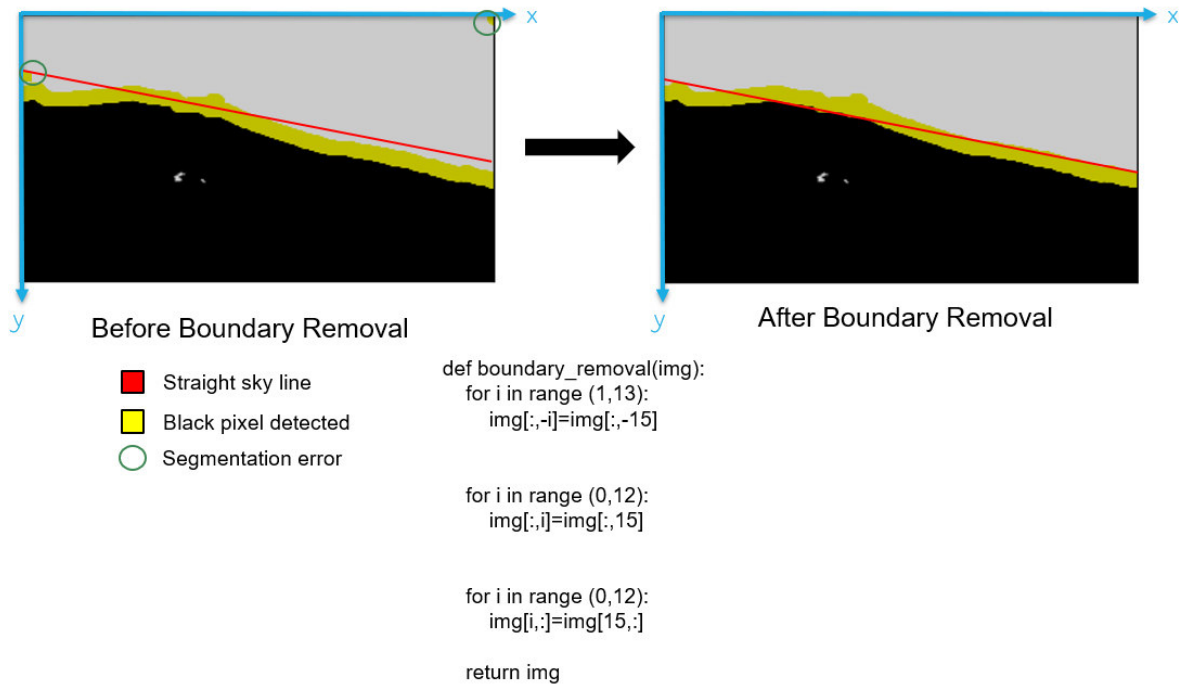


Figure 3.4. Boundary removal function.

(Source: Own)

3.2.2. Sky line motion prediction

The time consumed by the algorithm to detect and rotate the sky line of 3 subsequent frames of the video dataset is 0.21 seconds, which is not ideal for implementation on a real-time platform. To reduce the time consumed, the prediction of a uniform motion of the sky line is programmed. First, the values of a and b of the sky line equation of the first and second frame are stored applying the previously developed algorithm as shown in Figure 3.5. Then, the increment value of the angle of the slope a and b of the first two frames is stored as shown in equation (3.1) and (3.2).

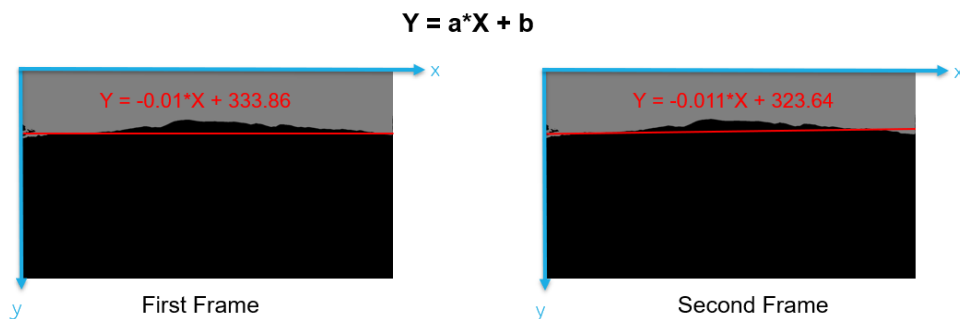


Figure 3.5. First and second frame sky line.

(Source: Own)

$$\theta_{increment} = \theta_{i+1} - \theta_i \quad (3.1)$$

$$b_{increment} = b_{i+1} - b_i \quad (3.2)$$

Where:

θ_i Angle of the slope of the first sky line frame

θ_{i+1} Angle of the slope of the second sky line frame

b_i y – axis intersection of the first sky line frame

b_{i+1} y – axis intersection of the second sky line frame

These values are incremented to those stored in the sky line of the second frame as shown in equation (3.3) and (3.4) to predict the location of the sky line of the third frame as shown in the equation (3.5).

$$\theta_{prediction} = \theta_{i+1} + \theta_{increment} \quad (3.3)$$

$$b_{prediction} = b_{i+1} + b_{increment} \quad (3.4)$$

$$Y_{prediction} = a_{prediction} \cdot X_{prediction} + b_{prediction} \quad (3.5)$$

Where:

$a_{prediction}$ Slope of the prediction angle as shown in equation (3.6)

$(X_{prediction}, Y_{prediction})$ Motion prediction line equation coordinates

$$a_{prediction} = \tan \theta_{prediction} \quad (3.6)$$

Finally, once the sky line of the third frame is predicted, the neighboring pixels with intensity 0 are searched to store them and thus determine the correct sky line as shown in Figure 3.6.

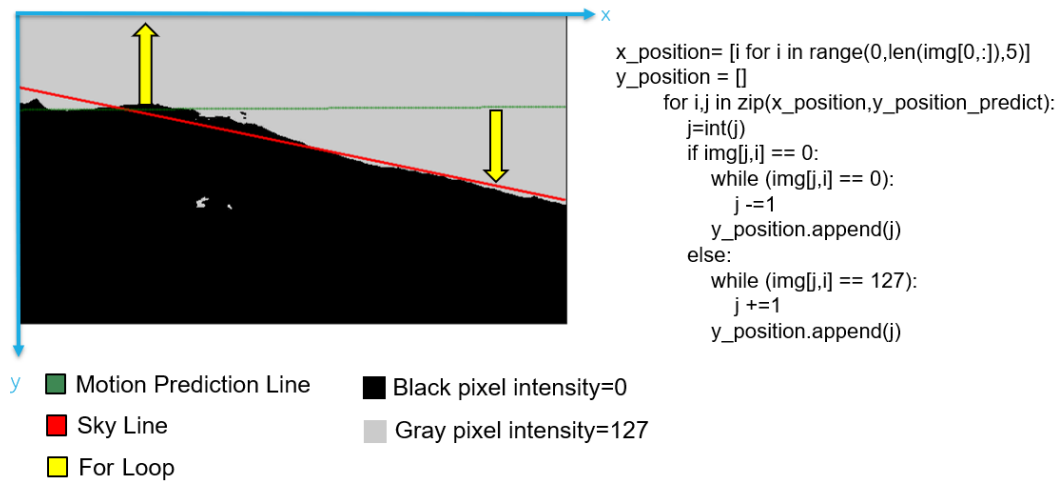


Figure 3.6. Identifying the correct sky line.

(Source: Own)

For the proposed example, the angle of the slope of each frame is calculated below:

$$\theta_1 = \tan^{-1} -0.01 = -0.01 \text{ rad} = -0.57^\circ$$

$$\theta_2 = \tan^{-1} -0.011 = -0.011 \text{ rad} = -0.63^\circ$$

$$\theta_{\text{increment}} = -0.63^\circ - (-0.57^\circ) = -0.06^\circ$$

$$b_{\text{increment}} = 323.64 - 333.86 = -10.22$$

$$\theta_{\text{prediction}} = -0.63^\circ + (-0.06^\circ) = -0.69^\circ$$

$$a_{\text{prediction}} = \tan -0.69^\circ = -0.012$$

$$b_{\text{prediction}} = 323.64 + (-10.22) = 313.42$$

$$Y_{\text{prediction}} = -0.012 \cdot X_{\text{prediction}} + 313.42$$

For the proposed example, the result is shown in Figure 3.7.

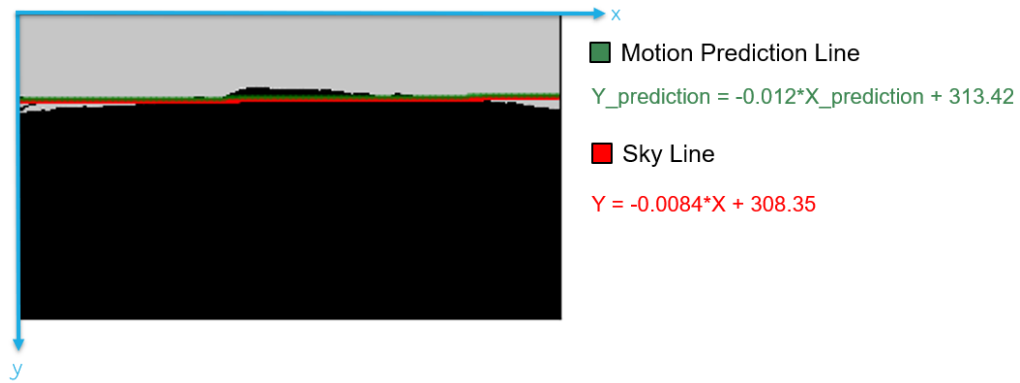


Figure 3.7. Sky line and motion prediction line of the third frame.

(Source: Own)

The result of the sky line equation of the third frame before optimizing the algorithm is:

$$Y = -0.0086 \cdot X + 313.13$$

The approximation error is calculated for the values of a and b as shown in equation (3.7).

$$\%Error = \frac{|Approx-Exact|}{Exact} \cdot 100 \quad (3.7)$$

$$Error_a = \frac{|0.0084 - 0.0086|}{0.0086} \cdot 100 = 2.33\%$$

$$Error_b = \frac{|308.35 - 313.13|}{313.13} \cdot 100 = 1.53\%$$

The computed errors are small, therefore optimized algorithm can be used. Furthermore, the time consumed by the optimized algorithm is 0.03 seconds, which is ideal to apply it to a real-time platform. The algorithm optimizations are added to Annex IV at the end of this thesis.

3.3. Analysis of the final design of the gimbal

The weight of 214 g of the mechanical parts of the gimbal are shown in Figure 3.8. The value is approximated since the printer prints additional layers of material around 20 g. Adding to this value the weight of the hardware components of 280.4 g, a total weight of 494.4 g is obtained. The UAV payload is maximum 500 g, therefore, the gimbal can place in it. It is necessary to obtain the real weigh of the gimbal to ensure the UAV can take off.

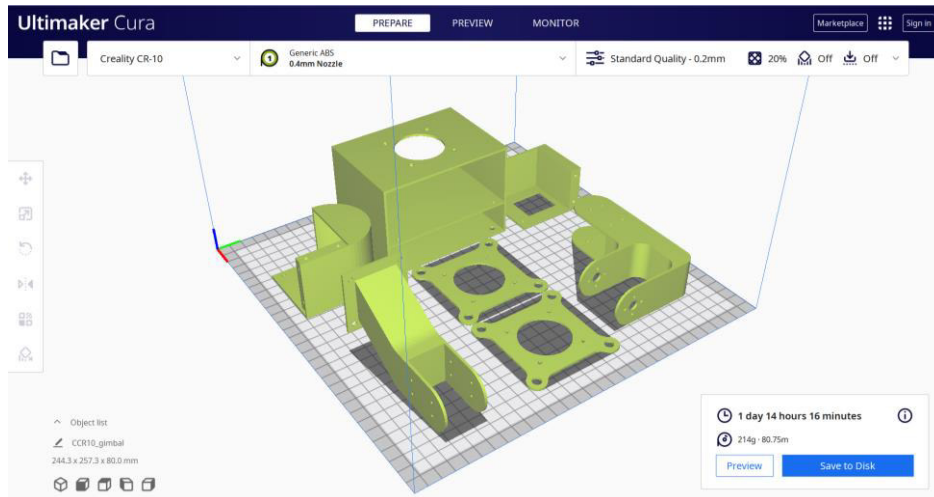


Figure 3.8. Approximate weight of 3D printed gimbal mechanical parts.

(Source: Own)

3.3.1. Gimbal project costs

Gimbal costs are shown in Table 3.1 with a total of 737.27 USD. The direct cost of the gimbal is less than the maximum budget, therefore the objective of manufacturing a cheap product is met.

Table 3.1. Gimbal Project Costs

Part Name	Quantity	Unit cost	Total Cost
NVIDIA Jetson Nano 2 GB Developer Kit (with WiFi)	1	92.44 USD	92.44 USD
Micro SD Card 64 GB	1	38.41 USD	38.41 USD
Huawei Mobil WiFi 3s	1	79 USD	79 USD
WiFi Antenna	1	17 USD	17 USD
OpenCR1.0 controller	1	218.40 USD	218.40 USD
Dynamixel XL430- W250-T servomotor	2	54.46 USD	108.92 USD
Raspberry Pi Camera Module V2	1	32.90 USD	32.90 USD
9-DOF Absolute Orientation IMU Sensor	1	46.11 USD	46.11 USD
Ball dampers	4	1.63 USD	6.52 USD
Screws	54	0.03 USD	1.62 USD
Nuts	16	0.03 USD	0.48 USD
Breadboard Cable	-	-	2 USD

3D printing parts	7 (214 [g], 38.3 [h])	2.50 USD/hour	95.75 USD
Total			739.55 USD

(Source: Own)

3.3.2. Vision Range

The gimbal is placed in the UAV ensuring that the camera is in the middle of it as shown in Figure 3.9.

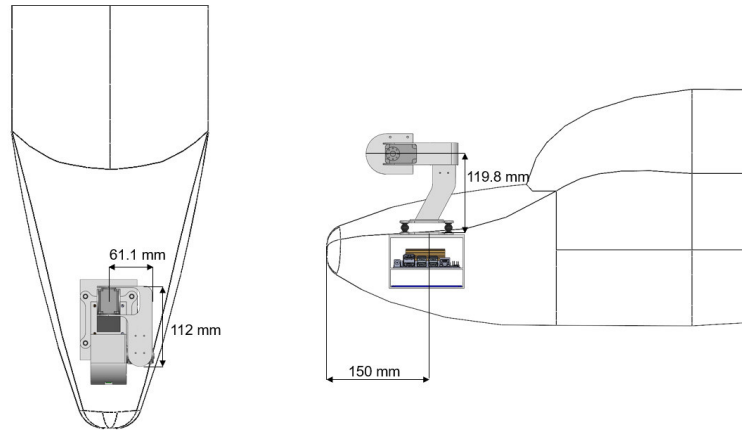


Figure 3.9. Location of the gimbal on the UAV.

(Source: Own)

As shown in Figure 3.10, the vertical field of view of the Raspberry Pi camera (48°) does not intersect with the nose of the UAV at a height of 100 mm and at 0° of rotation. At -45° of pitch rotation, the vertical field of view intersects with the nose of the UAV. Therefore, it is determined that the rotation range in pitch is between -30° and 45° and for roll between -45° and 45° to prevent the CSI cable of the camera from being damaged.

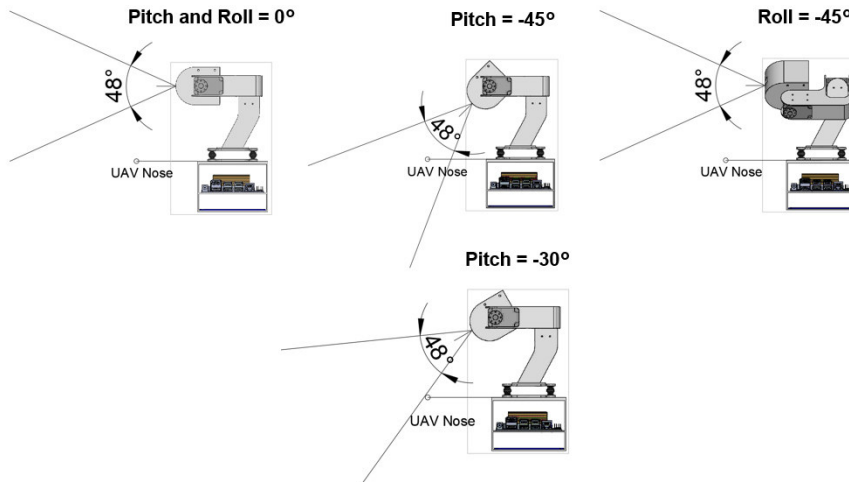


Figure 3.10. Rotation angle analysis.

(Source: Own)

4. CONCLUSIONS AND FUTURE WORK

4.1. Conclusions

A gimbal with two degrees of freedom capable of being installed on a UAV with a payload of 500 g was designed with a Computer Vision algorithm that allows its stabilization in roll rotation. The algorithm was programmed using the Python language to allow identifying the sky line of outdoor images using a convolutional neural network for segmentation.

The developed methodology allows the stabilization of a gimbal with open-source codes, having a great advantage over gimbals on the market that are limited by their own software. For this, OpenCV libraries were used to isolate the sky line from outdoor images, however the data obtained was very noisy, therefore it was decided to use machine learning to segment the sky and the ground and correctly identify the sky line and track it.

The results of the image segmentation are correctly predicted; however, they have small errors which can be improved by increasing the number of epochs and improving the training dataset. In addition, the algorithm was optimized to be able to run it in real time on embedded platforms by selecting a Jetson Nano capable of working with a dedicated GPU to fulfill these tasks.

The final design of the gimbal meets the users' requirements by not affecting the aerodynamics of the UAV and having a low cost. However, its range of vision for both rotations is limited for pitch from -30 to $+45^\circ$ and for roll from -45° to 45° due to the intersection with the nose of the UAV and to avoid damaging the CSI cable of the camera. This could be improved by installing the gimbal upside down, which is possible, but the UAV must have landing gear.

4.2. Future work

It is proposed as a future work to increase the pitch stabilization in the algorithm to later merge them together with the IMU sensor data and in this way obtain more accurate stabilization results. This will allow more applications to be carried out, such as the 3D reconstruction of volcanoes through images that need to be taken with high precision.

As future work, a more accurate polynomial curve fitting will be implemented since the mountain peaks of the sky line do not form a straight line.

A control module will be programmed to send the signal to the gimbal actuators. For this, both the camera and the IMU sensor will be calibrated using open-source codes. In addition, the gimbal will be manufactured to carry out tests in outdoor spaces.

The segmentation dataset was difficult to obtain since it was the only one that has images of outdoor landscapes, however, not all scenarios resemble the landscape simulated in this thesis. For that reason, it is intended to create our own dataset with images of different flight tests, and in this way retrain the neural network to obtain better segmentation results.

The mechanical components of the gimbal were design statically, therefore, to obtain a more robust design, the loads generated by the wind should be considered through flight tests with the prototype.

BIBLIOGRAPHY

- [1] A. Idries, N. Mohamed, I. Jawhar, F. Mohamed, and J. Al-Jaroodi, "Challenges of developing UAV applications: A project management view," *IEOM 2015 - 5th Int. Conf. Ind. Eng. Oper. Manag. Proceeding*, Apr. 2015, doi: 10.1109/IEOM.2015.7093730.
- [2] "DJI Drones and Handheld Products - DJI." <https://www.dji.com/products> (accessed May 16, 2021).
- [3] "UKRSPECSYSTEMS - USG-212." <https://ukrspecsystems.com/drone-gimbals/usg-212> (accessed May 16, 2021).
- [4] "Real time volcano monitoring for early eruption prediction using unmanned aerial vehicles and image processing methods | Vliruos." <https://www.vliruos.be/en/projects/project/22?pid=4535> (accessed Feb. 17, 2022).
- [5] M. A. Boon, A. P. Drijfhout, and S. Tesfamichael, "COMPARISON OF A FIXED-WING AND MULTI-ROTOR UAV FOR ENVIRONMENTAL MAPPING APPLICATIONS: A CASE STUDY," 2017, doi: 10.5194/isprs-archives-XLII-2-W6-47-2017.
- [6] H. T. Nguyen, T. V Quyen, C. V Nguyen, A. M. Le, H. T. Tran, and M. T. Nguyen, "Control Algorithms for UAVs: A Comprehensive Survey," 2020, doi: 10.4108/eai.18-5-2020.164586.
- [7] "Aircraft Rotations." <https://www.grc.nasa.gov/www/k-12/airplane/rotations.html> (accessed Feb. 17, 2022).
- [8] S. Rianmora, K. Poulpanich, J. Rattanagosol, and V. Jitpaisarnwattana, "JMM Gimbal Stabilizer," Jun. 2019. Accessed: May 15, 2021. [Online]. Available: <https://ph02.tci-thaijo.org/index.php/isjet/article/view/199024>.
- [9] "Phantom 4 Pro - DJI." <https://www.dji.com/phantom-4-pro> (accessed Jun. 10, 2021).
- [10] "30X Optical Zoom Eo Camera Gimbal for Drone Uav Vtol Fixed Wing Surveillance Inspection Research&Rescue - China Camera Gimbal and Zoom Camera for Drone price | Made-in-China.com." <https://m.made-in-china.com/product/30X-Optical-Zoom-Eo-Camera-Gimbal-for-Drone-Uav-Vtol-Fixed-Wing-Surveillance-Inspection-Research-Rescue-926421325.html> (accessed Feb. 23, 2022).
- [11] "Three-axis Gimbal Camera Combo 10X Optical Zoom camera - Buy Product on Jiangsu Digital Eagle Technology Development Co., Ltd." <https://www.digitaleagle->

- uav.com/Three-axis-Gimbal-Camera-Combo-10X-Optical-Zoom-Camera-pd6079284.html (accessed Feb. 23, 2022).
- [12] D. J. Regner, "Object tracking using a camera gimbal mechanism," Florianópolis, SC., Dec. 2020. Accessed: May 15, 2021. [Online]. Available: <https://repositorio.ufsc.br/handle/123456789/218934>.
- [13] "Instrumentation - The Smart Gimbal." <https://sites.google.com/site/projectsmartgimbal/home/TechnicalDetail?overrideobile=true> (accessed Feb. 17, 2022).
- [14] K. A. Lamkin-Kennard and M. B. Popovic, "Sensors: Natural and Synthetic Sensors," *Biomechatronics*, pp. 81–107, 2019, doi: 10.1016/B978-0-12-812939-5.00004-5.
- [15] "9-DOF Absolute Orientation IMU Fusion Breakout." https://www.generationrobots.com/en/403072-9-dof-absolute-orientation-imu-fusion-breakout.html?fbclid=IwAR0iz1Cmqy6ZnBkP2hW89tJ-17dB2b_QCkxtb57t4tn8xOAOAikfEKbr2rU (accessed Feb. 17, 2022).
- [16] M. S. Avilés, "PROYECTO FINAL DE CARRERA DISEÑO DE UN SISTEMA DE ESTABILIZACIÓN UTILIZANDO SENSORES INTELIGENTES (DESIGN OF A STABILIZATION SYSTEM USING INTELLIGENT SENSORS) Estudios: Ingeniería de Telecomunicaciones."
- [17] "DJI Gimbal Control Unit for Zenmuse H3-3D (Part 46) CP.ZM.000069." https://www.bhphotovideo.com/c/product/1054661-REG/dji_cp_zm_000069_gcu_part_46_zh3_3d.html (accessed Feb. 17, 2022).
- [18] "T-Motor GB54-1 Gimbal BLDC Motor for Dji Phantom Digital Copro - China Indutry Drone and Brushless Motor price | Made-in-China.com." <https://m.made-in-china.com/product/T-Motor-GB54-1-Gimbal-BLDC-Motor-for-Dji-Phantom-Digital-Copro-849894475.html> (accessed Feb. 23, 2022).
- [19] M. Oña, "DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE PROPULSIÓN DISTRIBUIDA DE UNA AERONAVE NO TRIPULADA DE FUSELAJE INTEGRADO UTILIZADA EN AGRICULTURA DE PRECISIÓN ," Escuela Politécnica Nacional, Quito, 2020.
- [20] "Turnigy High Capacity 5200mAh 3S 12C Lipo Pack w/XT60 | HobbyKing." https://hobbyking.com/en_us/turnigy-high-capacity-5200mah-3s-12c-multi-rotor-lipo-pack-w-xt60.html?utm_source=Website&utm_medium=Home-

- Banner&utm_campaign=orderpopup_042021 (accessed Feb. 23, 2022).
- [21] B. Vergouw, H. Nagel, G. Bondt, and B. Custers, "Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments," pp. 21–45, 2016, doi: 10.1007/978-94-6265-132-6_2.
- [22] "China Tarot Gopro 3D V Metal 3 Axis Gimbal TI3t05 for Gopro Hero 5 Camera - China 3 Axis Gimbal and Tarot Gopro 3D V Gimbal price." <https://feixiang668.en.made-in-china.com/product/yCcxDIXPfkm/China-Tarot-Gopro-3D-V-Metal-3-Axis-Gimbal-TI3t05-for-Gopro-Hero-5-Camera.html> (accessed Feb. 23, 2022).
- [23] A. Peri, "Using Homography for Pose Estimation in OpenCV | by Abhinav Peri | Analytics Vidhya | Medium," Sep. 17, 2020. <https://medium.com/analytics-vidhya/using-homography-for-pose-estimation-in-opencv-a7215f260fdd> (accessed Feb. 27, 2022).
- [24] F. Qureshi, "Homography: A Case Study in Model Fitting." <http://csundergrad.science.uoit.ca/courses/cv-notes/notebooks/19-homography.html> (accessed Feb. 27, 2022).
- [25] "Tarot T2D 2 Axis 10X Optical Zoom Camera Gimbal HDMI Output Support Upside down/Inverted Installation." <https://www.rc-wing.com/tarot-2-axis-10x-optical-zoom-camera-gimbal-hdmi-output.html> (accessed Feb. 23, 2022).
- [26] "36X 1080P HD Optical Zoom Camera For Drone | 3-Axis Hight Performance Gimbal System." <https://www.yangdaonline.com/sky-eye-36sz-1080p-36x-zoom-camera-for-drone-with-object-tracking-and-geotagging/> (accessed Feb. 23, 2022).
- [27] "Zenmuse X5S - DJI." <https://www.dji.com/zenmuse-x5s> (accessed Feb. 23, 2022).
- [28] "OpenCV: Basic Operations on Images." https://docs.opencv.org/3.4/d3/df2/tutorial_py_basic_ops.html (accessed Feb. 26, 2022).
- [29] R. Berriel, "OpenCV: Color-spaces and splitting channels - Rodrigo Berriel." <https://rodrigoberriel.com/2014/11/opencv-color-spaces-splitting-channels/> (accessed Feb. 26, 2022).
- [30] P. A. de O. Morais, D. M. de Souza, M. T. de M. Carvalho, B. E. Madari, and A. E. de Oliveira, "Predicting soil texture using image analysis," *Microchem. J.*, vol. 146, pp. 455–463, May 2019, doi: 10.1016/J.MICROC.2019.01.009.

- [31] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," *2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. iCoMET 2018 - Proc.*, vol. 2018-January, pp. 1–10, Apr. 2018, doi: 10.1109/ICOMET.2018.8346440.
- [32] D. Tyagi, "Introduction To Feature Detection And Matching | by Deepanshu Tyagi | Data Breach | Medium," 2019. <https://medium.com/data-breach/introduction-to-feature-detection-and-matching-65e27179885d> (accessed Feb. 27, 2022).
- [33] "Image Filtering Using Convolution in OpenCV | LearnOpenCV." <https://learnopencv.com/image-filtering-using-convolution-in-opencv/> (accessed Feb. 28, 2022).
- [34] "Tipos de núcleos de convolución: simplificados." <https://ichi.pro/es/tipos-de-nucleos-de-convolucion-simplificados-264161077525559> (accessed Feb. 28, 2022).
- [35] "OpenCV: Smoothing Images." https://docs.opencv.org/3.4/d4/d13/tutorial_py_filtering.html (accessed Feb. 28, 2022).
- [36] M. K. M. Rabby, B. Chowdhury, and J. H. Kim, "A modified canny edge detection algorithm for fruit detection classification," *ICECE 2018 - 10th Int. Conf. Electr. Comput. Eng.*, pp. 237–240, Feb. 2019, doi: 10.1109/ICECE.2018.8636811.
- [37] "Edge Detection Using OpenCV | LearnOpenCV." <https://learnopencv.com/edge-detection-using-opencv/> (accessed Feb. 28, 2022).
- [38] "OpenCV: Canny Edge Detection." https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html (accessed Feb. 28, 2022).
- [39] "Least Squares Fitting -- from Wolfram MathWorld." <https://mathworld.wolfram.com/LeastSquaresFitting.html> (accessed Feb. 28, 2022).
- [40] I. El Naqa, M. J. Murphy, I. El Naqa, and M. J. Murphy, "What Is Machine Learning?," *Mach. Learn. Radiat. Oncol.*, pp. 3–11, 2015, doi: 10.1007/978-3-319-18305-3_1.
- [41] A. A.-H. of measuring system design and undefined 2005, "Artificial neural networks," *wsc10.softcomputing.net*, Accessed: Mar. 01, 2022. [Online]. Available: http://wsc10.softcomputing.net/ann_chapter.pdf.
- [42] R. Mendoza, "Entendiendo las Redes Neuronales Artificiales | by Ricardo Mendoza

- | Medium,” May 16, 2019. <https://medium.com/@ricardojmv85/entendiendo-las-redes-neuronales-artificiales-1e0f4523eae6> (accessed Mar. 01, 2022).
- [43] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay,” Mar. 2018, Accessed: Mar. 01, 2022. [Online]. Available: <https://arxiv.org/abs/1803.09820v2>.
- [44] V. Rodríguez, “Conceptos básicos sobre redes neuronales,” Oct. 30, 2018. <https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales> (accessed Mar. 01, 2022).
- [45] D. M. Kline and V. L. Berardi, “Revisiting squared-error and cross-entropy functions for training neural network classifiers,” *Neural Comput. Appl.*, vol. 14, no. 4, pp. 310–318, Dec. 2005, doi: 10.1007/S00521-005-0467-Y/TABLES/2.
- [46] J. Brownlee, “What is the Difference Between a Batch and an Epoch in a Neural Network?,” 2018, Accessed: Mar. 01, 2022. [Online]. Available: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.
- [47] S. Sornapudi *et al.*, “Deep Learning Nuclei Detection in Digitized Histology Images by Superpixels,” *J. Pathol. Inform.*, vol. 9, no. 1, Jan. 2018, doi: 10.4103/JPI.JPI_74_17.
- [48] H. Zulkifli, “Understanding Learning Rates and How It Improves Performance in Deep Learning | by Hafidz Zulkifli | Towards Data Science,” Jan. 21, 2018. <https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10> (accessed Mar. 01, 2022).
- [49] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-January, pp. 1–6, Mar. 2018, doi: 10.1109/ICENGTECHNOL.2017.8308186.
- [50] “Convolutional Neural Network. Learn Convolutional Neural Network from... | by dshahid380 | Towards Data Science.” <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529> (accessed Mar. 01, 2022).
- [51] R. Dass, R. Dass, and S. Devi, “Image Segmentation Techniques 1,” Accessed: Mar. 01, 2022. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.227.6638>.
- [52] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation.” [Online]. Available: <http://mb.informatik.uni->

freiburg.de/.

- [53] “Implementing UNet in Pytorch,” Jul. 15, 2020. <https://becominghuman.ai/implementing-unet-in-pytorch-8c7e05a121b4> (accessed Mar. 01, 2022).
- [54] C. Riba, *Diseño concurrente*. Ediciones UPC, 2002.
- [55] “GoPro | La cámara más versátil del mundo | Compra ahora y ahorra.” <https://gopro.com/es/es/> (accessed Mar. 09, 2022).
- [56] “Camera Module V2 for Raspberry Pi.” <https://www.generationrobots.com/en/402667-camera-module-v2-raspberry-pi.html> (accessed Mar. 09, 2022).
- [57] “OpenCR1.0 controller.” <https://www.generationrobots.com/en/402881-opencr10.html> (accessed Mar. 09, 2022).
- [58] “NVIDIA Jetson Nano 2 GB Developer Kit (with WiFi).” <https://www.generationrobots.com/en/403744-nvidia-jetson-nano-2-gb-developer-kit-with-wifi.html> (accessed Mar. 10, 2022).
- [59] “Raspberry Pi Model A+.” <https://www.generationrobots.com/en/402065-raspberry-pi-model-a.html> (accessed Mar. 10, 2022).
- [60] ROBOTIS, “Dynamixel XL430-W250-T Servo.” <https://www.generationrobots.com/en/402823-dynamixel-xl430-w250-t-servomotor.html> (accessed Mar. 10, 2022).
- [61] “GBM4114 Gimbal Brushless Motor 100T (Camera Range: 3200g) | Rctimer.” <http://rctimer.com/gbm4114-gimbal-brushless-motor-100t-camera-range-3200g-p0458.html> (accessed Mar. 10, 2022).
- [62] N. (NASA) Hall, “Shape Effects on Drag,” 2021. <https://www.grc.nasa.gov/www/k-12/airplane/shaped.html> (accessed Jun. 10, 2021).
- [63] H. C. Park, S. Chakir, Y. B. Kim, and T. Huynh, “A nonlinear backstepping controller design for high-precision tracking applications with input-delay gimbal systems,” *J. Mar. Sci. Eng.*, vol. 9, no. 5, 2021, doi: 10.3390/JMSE9050530.
- [64] R. P. Mihail, S. Workman, Z. Bessinger, and N. Jacobs, “Sky segmentation in the wild: An empirical study,” *2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016*, May 2016, doi: 10.1109/WACV.2016.7477637.

- [65] “GitHub - milesial/Pytorch-UNet: PyTorch implementation of the U-Net for image semantic segmentation with high quality images.” <https://github.com/milesial/Pytorch-UNet?fbclid=IwAR0FD1opZTWUSr71rFvB3MALblq-wfubfxuw3HOpnQG0EaSQ4xgqaHblkx0> (accessed Mar. 13, 2022).
- [66] ROBOTIS, “How to choose the right Dynamixel servomotor?,” 2019. <https://www.generationrobots.com/blog/en/how-to-choose-the-right-dynamixel-servomotor/> (accessed Mar. 16, 2022).
- [67] ROBOTIS, “XL430-W250,” 2018. http://support.robotis.com/en/product/actuator/dynamixel_x/xl_series/xl430-w250.htm (accessed Mar. 16, 2022).
- [68] ROBOTIS, “XL-320,” 2018. http://support.robotis.com/en/product/actuator/dynamixel_x/xl_series/xl-320.htm (accessed Mar. 19, 2022).
- [69] S.-J. Kim, S. Lee, and T.-U. Kim, “Structural safety factor for small unmanned aircraft,” *J. Korean Soc. Aviat. Aeronaut.*, vol. 25, no. 2, pp. 12–17, 2017, doi: 10.12985/KSAA.2017.25.2.012.
- [70] Flying Tech, “Mini Vibration Damping Ball (65g) | Flying Tech,” 2021. <https://www.flyingtech.co.uk/frames-props-parts-accessories/rubber-vibration-damping-balls-100-300g?sku=DIY-VDB-100> (accessed Mar. 22, 2022).
- [71] R. G. Budynas and J. K. Nisbett, *Shigley’s Mechanical Engineering Design*, 10th ed. McGraw-Hill Higher Education, 2015.
- [72] MINISTERIO DE ELECTRICIDAD Y ENERGIA RENOVABLE (MEER), “ATLAS EOLICO DEL ECUADOR: CON FINES DE GENERACION ELECTRICA | ARIAE,” Mar. 2013. Accessed: Mar. 18, 2022. [Online]. Available: <https://www.ariae.org/servicio-documental/atlas-eolico-del-ecuador-con-fines-de-generacion-electrica>.
- [73] F. P. (Ferdinand P. Beer, *Mechanics of materials*. McGraw-Hill, 2011.
- [74] Ultimaker, “Ultimaker ABS TDS – Ultimaker Support,” Apr. 19, 2020. <https://support.ultimaker.com/hc/en-us/articles/360012759139-Ultimaker-ABS-TDS> (accessed Mar. 21, 2022).

APPENDICES

APPENDIX I. Sift, Surf and Orb feature detection code

```
import cv2
import numpy as np
import glob

filenames=glob.glob("Path/*.png")

for file in filenames:

    img = cv2.imread(file, cv2.IMREAD_GRAYSCALE)

    sift = cv2.xfeatures2d.SIFT_create()
    surf = cv2.xfeatures2d.SURF_create()
    orb = cv2.ORB_create(nfeatures=1500)

    keypoints_sift, descriptors = sift.detectAndCompute(img, None)
    keypoints_surf, descriptors = surf.detectAndCompute(img, None)
    keypoints_orb, descriptors = orb.detectAndCompute(img, None)

    img1 = cv2.drawKeypoints(img, keypoints_sift, None)
    img2 = cv2.drawKeypoints(img, keypoints_surf, None)
    img3 = cv2.drawKeypoints(img, keypoints_orb, None)

    cv2.imshow("Image", img1)
    #cv2.imshow("Image", img2)
    #cv2.imshow("Image", img3)

    cv2.waitKey(0)
    cv2.destroyAllWindows(1)
```

APPENDIX II. Canny Edge Detection code

```
import numpy as np
import cv2 as cv
import csv
import time
from matplotlib import pyplot as plt
import glob
from os.path import isfile, join
from scipy import ndimage

filenames=glob.glob("Path/*.jpg")

for file in filenames:
    ts=time.time()
    #img = cv.imread(file,0)
    imgraw = cv.imread(file)
    imgraw=cv.cvtColor(imgraw,cv.COLOR_BGR2RGB)
    img=cv.cvtColor(imgraw,cv.COLOR_RGB2HSV)
    Raw_edges = cv.Canny(img,100,255,apertureSize=3,L2gradient=True)

#FILTERS
    Blur_Filter = cv.blur(src=img, ksize=(5,5))
    Edges_Blur =
cv.Canny(Blur_Filter,100,255,apertureSize=3,L2gradient=True)

    Gaussian_Blur_Filter = cv.GaussianBlur(img,(5,5),sigmaX=0, sigmaY=0)
    Edges_Gaussian =
cv.Canny(Gaussian_Blur_Filter,100,255,apertureSize=3,L2gradient=True)

    Median_Blur_Filter = cv.medianBlur(src=img, ksize=11)
    Median_Bilateral=cv.bilateralFilter(src=Median_Blur_Filter, d=15,
sigmaColor=75, sigmaSpace=75)
    Edges_Median =
cv.Canny(Median_Blur_Filter,100,255,apertureSize=3,L2gradient=True)

    Bilateral_Filter = cv.bilateralFilter(src=img, d=15, sigmaColor=75,
sigmaSpace=75)
    Edges_Bilateral =
cv.Canny(Bilateral_Filter,100,255,apertureSize=3,L2gradient=True)

    te=time.time()
    print('--- %s seconds ---'%(te-ts))
    print("-----")

    plt.subplot(521),plt.imshow(img,cmap = 'gray')
    plt.title('Raw Image'), plt.xticks([]), plt.yticks([])
```



```
plt.subplot(522),plt.imshow(Raw_edges,cmap = 'gray')
plt.title('Raw_Edges'), plt.xticks([]), plt.yticks([])
plt.subplot(523),plt.imshow(Blur_Filter,cmap = 'gray')
plt.title('Blur_Filter'), plt.xticks([]), plt.yticks([])
plt.subplot(524),plt.imshow(Edges_Blur,cmap = 'gray')
plt.title('Edges_Blur'), plt.xticks([]), plt.yticks([])

plt.subplot(525),plt.imshow(Gaussian_Blur_Filter,cmap = 'gray')
plt.title('Gaussian_Blur_Filter'), plt.xticks([]), plt.yticks([])
plt.subplot(526),plt.imshow(Edges_Gaussian,cmap = 'gray')
plt.title('Edges_Gaussian'), plt.xticks([]), plt.yticks([])

plt.subplot(527),plt.imshow(Median_Blur_Filter,cmap = 'gray')
plt.title('Median_Blur_Filter'), plt.xticks([]), plt.yticks([])

plt.subplot(528),plt.imshow(Edges_Median,cmap = 'gray')
plt.title('Edges_Median'), plt.xticks([]), plt.yticks([])

plt.subplot(529),plt.imshow(Bilateral_Filter,cmap = 'gray')
plt.title('Bilateral_Filter'), plt.xticks([]), plt.yticks([])
plt.subplot(5,2,10),plt.imshow(Edges_Bilateral,cmap = 'gray')
plt.title('Edges_Bilateral'), plt.xticks([]), plt.yticks([])

plt.show()
```

APPENDIX III. U-Net Image Segmentation CNN Data Loading Code

```
import logging
import os
from os import listdir
from os.path import splitext
from pathlib import Path

import numpy as np
import torch
from PIL import Image
from torch.utils.data import Dataset
from glob import glob

class BasicDataset(Dataset):
    def __init__(self, images_dir: str, masks_dir: str, scale: float = 1.0,
mask_suffix: str = ''):
        self.images_dir = Path(images_dir)
        #i=str(self.images_dir)
        self.masks_dir = Path(masks_dir)
        assert 0 < scale <= 1, 'Scale must be between 0 and 1'
        self.scale = scale
        self.mask_suffix = mask_suffix

        self.ids = [splitext(file)[0] for file in listdir(masks_dir) if not
file.startswith('.')]
        if not self.ids:
            raise RuntimeError('No input file found in {}, make sure you put
your images there'.format(masks_dir))
        logging.info('Creating dataset with {}
examples'.format(len(self.ids)))

    def __len__(self):
        return len(self.ids)

    @classmethod
    def preprocess(cls, pil_img, scale, is_mask):
        w, h = pil_img.size
        newW, newH = int(scale * w), int(scale * h)
        assert newW > 0 and newH > 0, 'Scale is too small, resized images
would have no pixel'
        pil_img = pil_img.resize((newW, newH))
        img_ndarray = np.asarray(pil_img)

        if img_ndarray.ndim == 2 and not is_mask:
            img_ndarray = img_ndarray[np.newaxis, ...]
        elif not is_mask:
            img_ndarray = img_ndarray.transpose((2, 0, 1))
```

```

        if not is_mask:
            img_ndarray = img_ndarray / 255

        return img_ndarray

    @classmethod
    def load(cls, filename):
        ext = splitext(filename)[1]
        if ext in ['.npz', '.npy']:
            return Image.fromarray(np.load(filename))
        elif ext in ['.pt', '.pth']:
            return Image.fromarray(torch.load(filename).numpy())
        else:
            return Image.open(filename)

    def __getitem__(self, idx):
        name = self.ids[idx]

        for i in self.images_dir.glob(name):
            for k in self.masks_dir.glob(name+ self.mask_suffix + '.*'):
                mask = self.load(k)

                for j in i.glob('*/*'):
                    img = self.load(j)
                    assert img.size == mask.size, \
                        'Image and mask {} should be the same size, but are {}
and {}'.format(name,img.size,mask.size)

                    img = self.preprocess(img, self.scale, is_mask=False)
                    mask = self.preprocess(mask, self.scale, is_mask=True)
                    return {
                        'image':
torch.as_tensor(img.copy()).float().contiguous(),
                        'mask': torch.as_tensor(mask.copy()).long().contiguous()
                    }

class SkyDataset(BasicDataset):
    def __init__(self, images_dir, masks_dir, scale=1):
        #super().__init__(images_dir, masks_dir, scale, mask_suffix='_mask')
        super().__init__(images_dir, masks_dir, scale, mask_suffix='')

```

(Source: [65])

APPENDIX IV. Sky line motion compensation algorithm

```
import numpy as np
import cv2 as cv
import imutils
import time
from matplotlib import pyplot as plt
import glob

def fit(x, y):

    xbar = sum(x)/len(x)
    ybar = sum(y)/len(y)
    n = len(x) # or len(y)

    numer = sum([xi*yi for xi,yi in zip(x, y)]) - n * xbar * ybar
    denom = sum([xi**2 for xi in x]) - n * xbar**2

    a = numer / denom
    b = ybar - a * xbar
    return a, b

def boundary_removal(img):
    for i in range (1,13):
        img[:,-i]=img[:,-15]

    for i in range (0,12):
        img[:,i]=img[:,15]

    for i in range (0,12):
        img[i,:]=img[15,:]

    return img

filenames=glob.glob("Path/*.jpg")

last_last_straightline_params = None
last_straightline_params = None
image_counter = 0

for file in filenames:
    ts=time.time()
    image_counter +=1
    if image_counter == 1:
```

```

img = cv.imread(file,0)
img = boundary_removal(img)

x_position = []
y_position = []
for i in range (0,len(img[0,:]),5):
#for i in range (len(img[0,:])):
    for j in range (0,len(img[:,0]),5):
        #for j in range (len(img[:,0])):
            if img[j,i]==0:
                x_position.append(i)
                y_position.append(j)
                break
a, b = fit(x_position, y_position)
angle_a=np.arctan(a)
last_last_angle=angle_a
last_last_b=b

#Compensate Movement
img_reference=img
reference_angle=angle_a

elif image_counter == 2:
img = cv.imread(file,0)
img = boundary_removal(img)

x_position = []
y_position = []
for i in range (0,len(img[0,:]),5):
#for i in range (len(img[0,:])):
    for j in range (0,len(img[:,0]),5):
        #for j in range (len(img[:,0])):
            if img[j,i]==0:
                x_position.append(i)
                y_position.append(j)
                break
a, b = fit(x_position, y_position)
angle_a=np.arctan(a)
last_angle=angle_a
last_b=b
theta_inc=last_angle-last_last_angle
b_inc=last_b-last_last_b

#Compensate Movement
angle_2=angle_a
angle_compensate=(np.rad2deg(angle_2-reference_angle))
rotate_image=img
align_image=imutils.rotate(rotate_image,angle_compensate)

```

```

elif image_counter > 2:
    img = cv.imread(file,0)
    img = boundary_removal(img)

    theta_pred=theta_inc+last_angle
    b_pred=last_b+b_inc
    a_p=np.tan(theta_pred)
    b_p=b_pred

    last_last_angle=last_angle
    last_last_b=last_b

    x_position_predict = [i for i in range(0,len(img[0,:]),5)]
    #x_position_predict = [i for i in range(len(img[0,:]))]
    y_position_predict = []
    for x in x_position_predict:
        y=a_p*x +b_p
        y_position_predict.append(y)

    x_position = x_position_predict
    y_position = []
    for i,j in zip(x_position,y_position_predict):
        j=int(j)
        if img[j,i] == 0:
            while (img[j,i] == 0):
                j -=1
            y_position.append(j)
        else:
            while (img[j,i] == 127):
                j +=1
            y_position.append(j)

    a, b = fit(x_position, y_position)
    x_line = np.arange(min(x_position), max(x_position), 1)
    y_line=a*x_line +b
    angle_a=np.arctan(a)
    last_angle=angle_a
    last_b=b
    theta_inc=last_angle-last_last_angle
    b_inc=last_b-last_last_b

    #Compensate Movement
    angle_mov=angle_a
    angle_compensate=(np.rad2deg(angle_mov-reference_angle))
    rotate_image=img

```

```

align_image=imutils.rotate(rotate_image,angle_compensate)

te=time.time()
print('--- %s seconds ---'%(te-ts))
print("-----")

plt.subplot(211),plt.imshow(img,cmap = 'gray')
plt.title('Image'), plt.xticks([]), plt.yticks([])

plt.subplot(212),plt.imshow(img,cmap = 'gray')
plt.title('Motion Prediction'), plt.xticks([]), plt.yticks([])
#plt.scatter(x_position, y_position,c='y',s=1)
plt.scatter(x_line, y_line,s=0.1, c='r')
plt.scatter(x_position_predict, y_position_predict,s=0.1, c='g')

plt.show()

#Compensate Movement
plt.subplot(121),plt.imshow(img_reference,cmap = 'gray')
plt.title('Reference Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(align_image,cmap = 'gray')
plt.title('Compensated Image'), plt.xticks([]), plt.yticks([])
plt.show()

```

APPENDIX V. Mechanical properties of steel and aluminum

Table A-22

Results of Tensile Tests of Some Metals* Source: J. Datsko, "Solid Materials," chap. 32 in Joseph E. Shigley, Charles R. Mischke, and Thomas H. Brown, Jr. (eds.-in-chief), *Standard Handbook of Machine Design*, 3rd ed., McGraw-Hill, New York, 2004, pp. 32.49–32.52.

Number	Material	Condition	Strength (Tensile)					Strain Strength, Exponent m	Fracture Strain ϵ_f
			Yield S_{yy} MPa (kpsi)	Ultimate S_u MPa (kpsi)	Fracture, σ_{ff} MPa (kpsi)	Coefficient σ_{0r} MPa (kpsi)			
1018	Steel	Annealed	220 (32.0)	341 (49.5)	628 (91.1) [†]	620 (90.0)	0.25	1.05	
1144	Steel	Annealed	358 (52.0)	646 (93.7)	898 (130) [†]	992 (144)	0.14	0.49	
1212	Steel	HR	193 (28.0)	424 (61.5)	729 (106) [†]	758 (110)	0.24	0.85	
1045	Steel	Q&T 600°F	1520 (220)	1580 (230)	2380 (345)	1880 (273) [†]	0.041	0.81	
4142	Steel	Q&T 600°F	1720 (250)	1930 (280)	2340 (340)	1760 (255) [†]	0.048	0.43	
303	Stainless steel	Annealed	241 (35.0)	601 (87.3)	1520 (221) [†]	1410 (205)	0.51	1.16	
304	Stainless steel	Annealed	276 (40.0)	568 (82.4)	1600 (233) [†]	1270 (185)	0.45	1.67	
2011	Aluminum alloy	T6	169 (24.5)	324 (47.0)	325 (47.2) [†]	620 (90)	0.28	0.10	
2024	Aluminum alloy	T4	296 (43.0)	446 (64.8)	533 (77.3) [†]	689 (100)	0.15	0.18	
7075	Aluminum alloy	T6	542 (78.6)	593 (86.0)	706 (102) [†]	882 (128)	0.13	0.18	

*Values from one or two heats and believed to be attainable using proper purchase specifications. The fracture strain may vary as much as 100 percent.

[†]Derived value.

(Source: [71])

APPENDIX VI. Metric Threads

Table 8-1

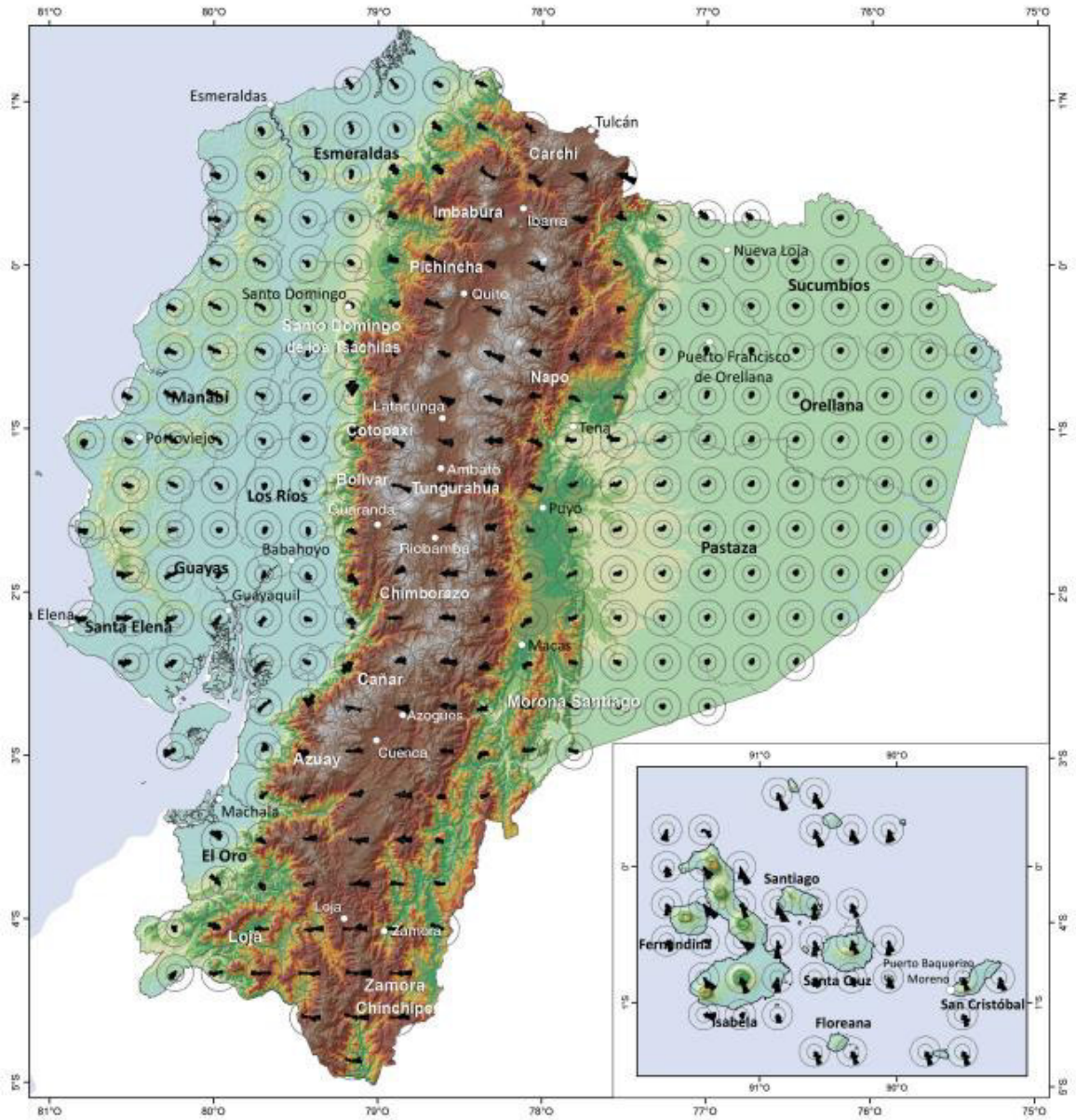
Diameters and Areas of Coarse-Pitch and Fine-Pitch Metric Threads.*

Nominal Major Diameter d mm	Coarse-Pitch Series			Fine-Pitch Series		
	Pitch p mm	Tensile-Stress Area A_t mm ²	Minor-Diameter Area A_m mm ²	Pitch p mm	Tensile-Stress Area A_t mm ²	Minor-Diameter Area A_m mm ²
1.6	0.35	1.27	1.07			
2	0.40	2.07	1.79			
2.5	0.45	3.39	2.98			
3	0.5	5.03	4.47			
3.5	0.6	6.78	6.00			
4	0.7	8.78	7.75			
5	0.8	14.2	12.7			
6	1	20.1	17.9			
8	1.25	36.6	32.8	1	39.2	36.0
10	1.5	58.0	52.3	1.25	61.2	56.3
12	1.75	84.3	76.3	1.25	92.1	86.0
14	2	115	104	1.5	125	116
16	2	157	144	1.5	167	157
20	2.5	245	225	1.5	272	259
24	3	353	324	2	384	365
30	3.5	561	519	2	621	596
36	4	817	759	2	915	884
42	4.5	1120	1050	2	1260	1230
48	5	1470	1380	2	1670	1630
56	5.5	2030	1910	2	2300	2250
64	6	2680	2520	2	3030	2980
72	6	3460	3280	2	3860	3800
80	6	4340	4140	1.5	4850	4800
90	6	5590	5360	2	6100	6020
100	6	6990	6740	2	7560	7470
110				2	9180	9080

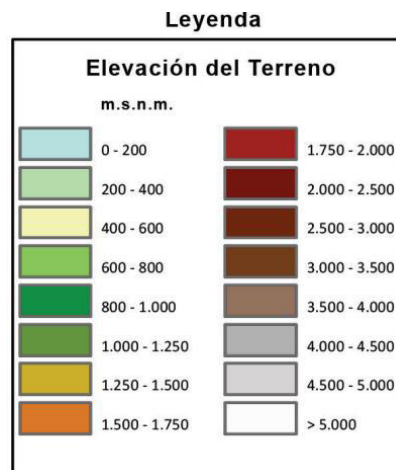
*The equations and data used to develop this table have been obtained from ANSI B1.1-1974 and B18.3.1-1978. The minor diameter was found from the equation $d_r = d - 1.226869p$, and the pitch diameter from $d_p = d - 0.649519p$. The mean of the pitch diameter and the minor diameter was used to compute the tensile-stress area.

(Source: [71])

APPENDIX VII. Wind Atlas of Ecuador



Altitud (m)	Temperatura (°C)	Presión (hPa)	Densidad (kg/m ³)
0	15.0	1013.2	1.225
100	14.3	1001.3	1.213
200	13.7	989.5	1.202
300	13.0	977.73	1.190
400	12.4	966.11	1.179
500	11.7	954.6	1.167
1000	8.5	898.7	1.112
2000	2.0	794.9	1.007
3000	-4.5	701.1	0.909
4000	-11.0	616.4	0.819
5000	-17.5	540.2	0.736

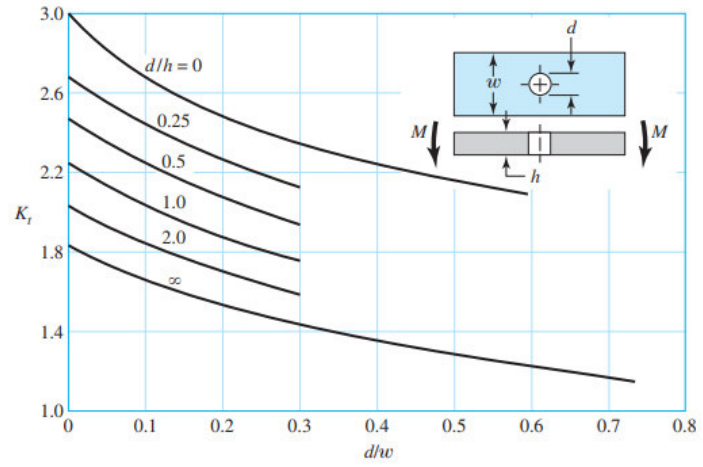


(Source: [72])

APPENDIX VIII. Stress-Concentration Factors K_t

Figure A-15-2

Rectangular bar with a transverse hole in bending.
 $\sigma_0 = Mc/I$, where
 $I = (w - d)h^3/12$.



(Source: [71])

APPENDIX IX. Coefficient for rectangular bars in torsion

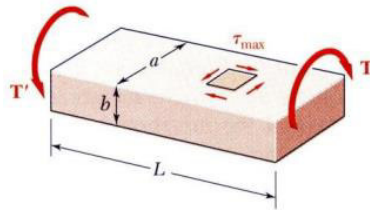


TABLE 3.1. Coefficients for Rectangular Bars in Torsion

a/b	c_1	c_2
1.0	0.208	0.1406
1.2	0.219	0.1661
1.5	0.231	0.1958
2.0	0.246	0.229
2.5	0.258	0.249
3.0	0.267	0.263
4.0	0.282	0.281
5.0	0.291	0.291
10.0	0.312	0.312
∞	0.333	0.333

(Source: [73])

APPENDIX X. ABS Plastic Mechanical Properties

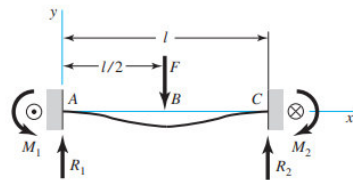
Mechanical properties*

	Injection molding		3D printing	
	Typical value	Test method	Typical value	Test method
Tensile modulus	2,030 MPa	ISO 527 (1 mm/min)	1,618.5 MPa	ISO 527 (1 mm/min)
Tensile stress at yield	43.6 MPa	ISO 527 (50 mm/min)	39 MPa	ISO 527 (50 mm/min)
Tensile stress at break	-	-	33.9 MPa	ISO 527 (50 mm/min)
Elongation at yield	4.8%	ISO 527 (50 mm/min)	3.5%	ISO 527 (50 mm/min)
Elongation at break	34%	ISO 527 (50 mm/min)	4.8%	ISO 527 (50 mm/min)
Flexural strength	-	-	70.5 MPa	ISO 178
Flexural modulus	-	-	2,070 MPa	ISO 178
Izod impact strength, notched (at 23 °C)	-	-	10.5 kJ/m ²	ISO 180
Charpy impact strength (at 23 °C)	58 kJ/m ²	ISO 179	-	
Hardness	-	-	76 (Shore D)	Durometer

(Source: [74])

APPENDIX XI. Beam fixed at both ends equations

Fixed supports—center load



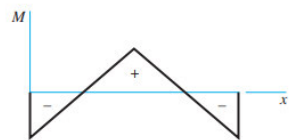
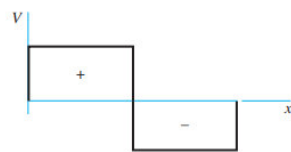
$$R_1 = R_2 = \frac{F}{2} \quad M_1 = M_2 = \frac{Fl}{8}$$

$$V_{AB} = -V_{BC} = \frac{F}{2}$$

$$M_{AB} = \frac{F}{8}(4x - l) \quad M_{BC} = \frac{F}{8}(3l - 4x)$$

$$y_{AB} = \frac{Fx^2}{48EI}(4x - 3l)$$

$$y_{\max} = -\frac{Fl^3}{192EI}$$



(Source: [71])

APPENDIX XII. Assembly and components drawings

List of drawings

GUAV.001. Gimbal assembly drawing

GUAV.101. Protective case

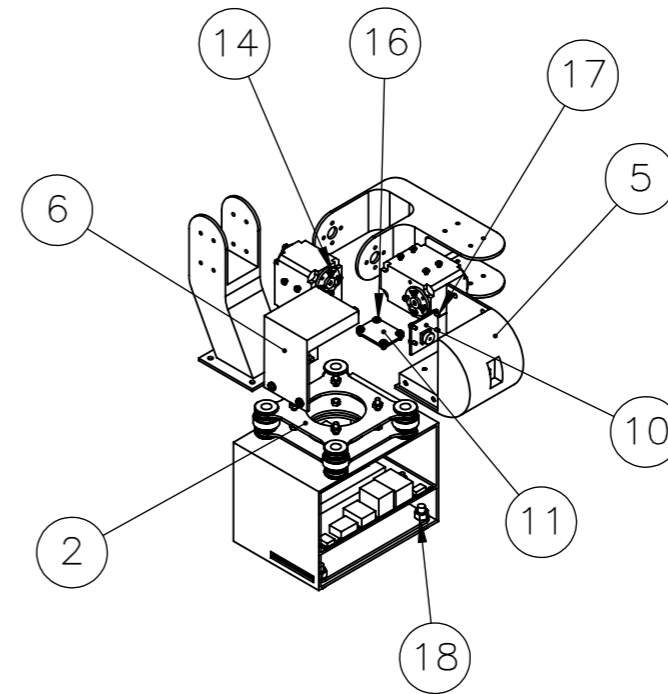
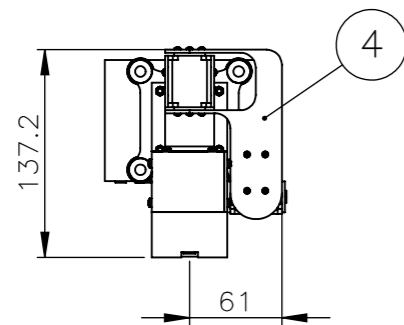
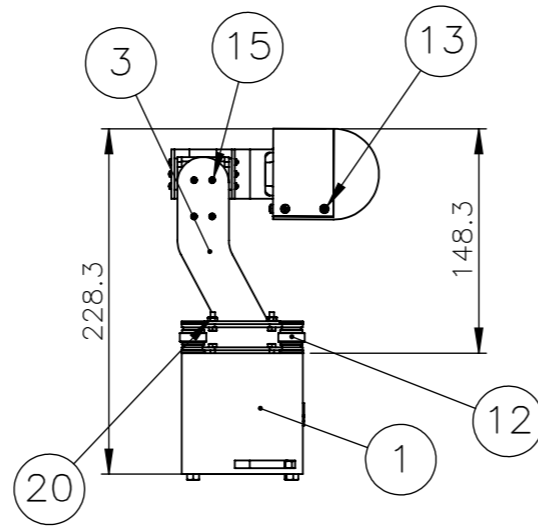
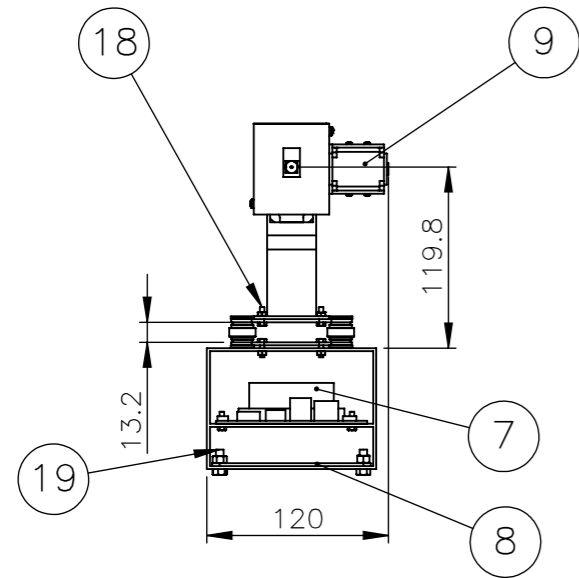
GUAV.102. Anti-vibration plate

GUAV.103. Roll rotation link

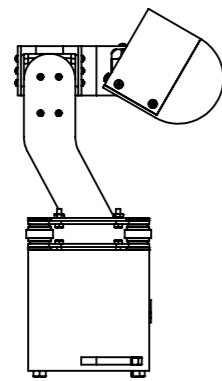
GUAV.104. Pitch rotation link

GUAV.105. Aerodynamic profile link

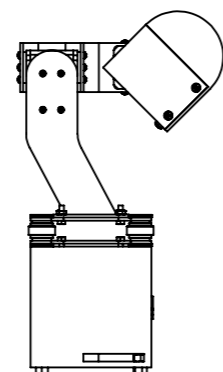
GUAV.106. Aerodynamic profile cover



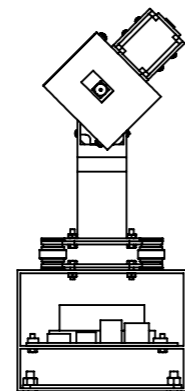
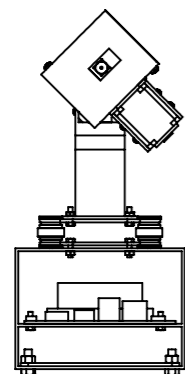
Gimbal maximum positions
Pitch = -30° Pitch = +45°



Roll = -45°



Roll = +45°

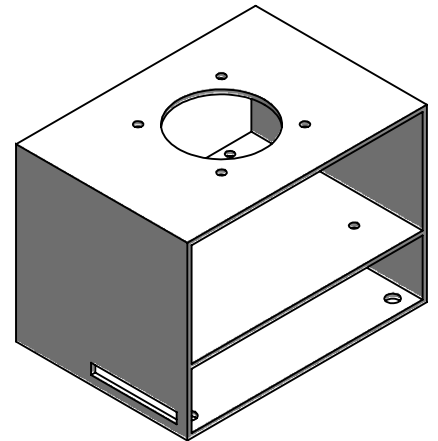
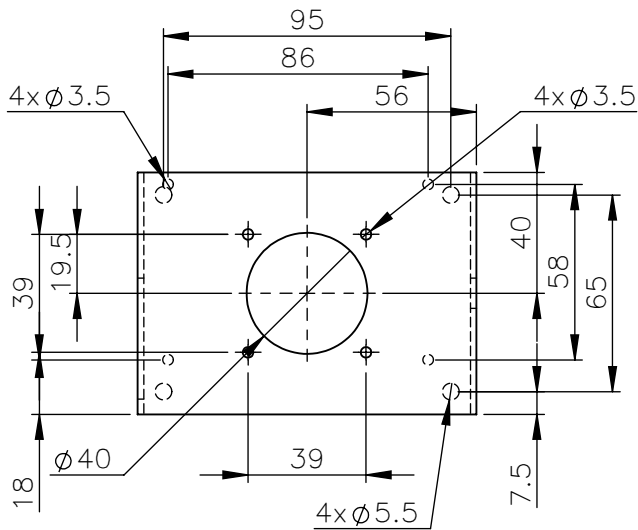
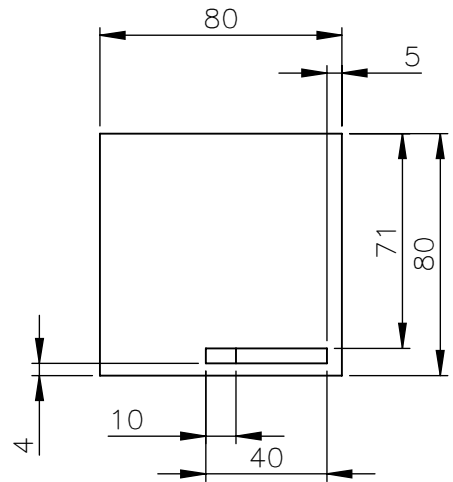
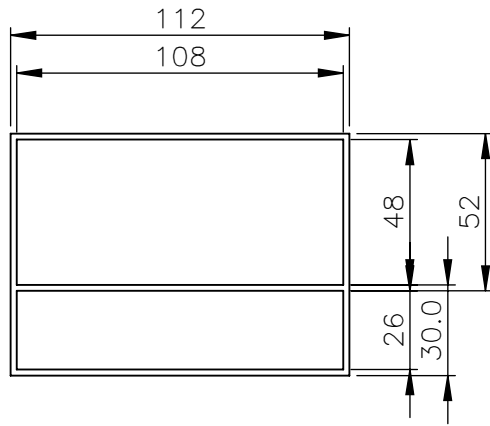


STANDARD ELEMENTS						
N°	ZONE	DESCRIPTION	DRAWING AND/OR STANDARD	QTY.	MATERIAL	OBSERVATION
21	C-6	HEX NUT M5	ANSI B18.2.4.1M	4	ALUMINUM	
20	B-2	HEX NUT M3	ANSI B18.2.4.1M	12	ALUMINUM	
19	B-1	BOLT M5x13	ANSI B18.6.7M	4	ALUMINUM	
18	A-1	BOLT M3x10	ANSI B18.6.7M	12	ALUMINUM	
17	A-7	SCREW M2x4	ANSI B18.6.7M	4	ALUMINUM	
16	A-6	SCREW M2.5x4	ANSI B18.6.7M	4	ALUMINUM	
15	A-4	SCREW M2x6	ANSI B18.6.7M	16	ALUMINUM	
14	A-6	SCREW M2x5	ANSI B18.6.7M	12	ALUMINUM	
13	A-4	SCREW M3x5	ANSI B18.6.7M	6	ALUMINUM	
12	B-4	VIBRATION DAMPENER BALL (100g)		4	RUBBER	D14/100g
11	B-6	ADAFRUIT BNO055 IMU SENSOR		1		
10	B-7	RASPBERRY PI CAMERA V2		1		
9	A-2	DYNAMIXEL XL430-W250-T SERVO		2		
8	B-2	OPENCV1.0 CONTROLLER		1		
7	B-2	NVIDIA JETSON NANO		1		2GB

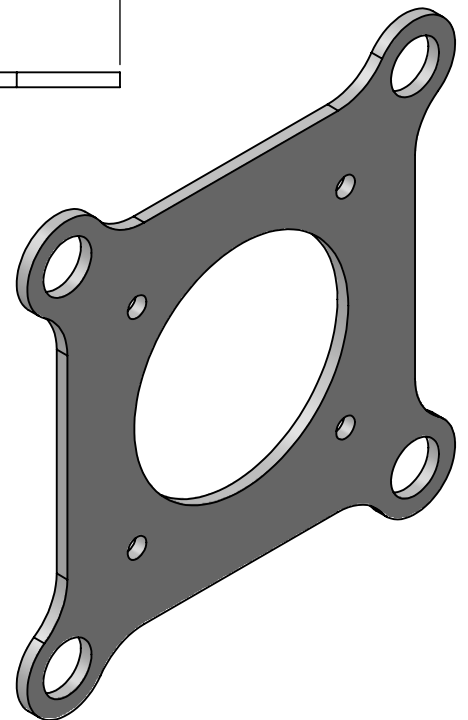
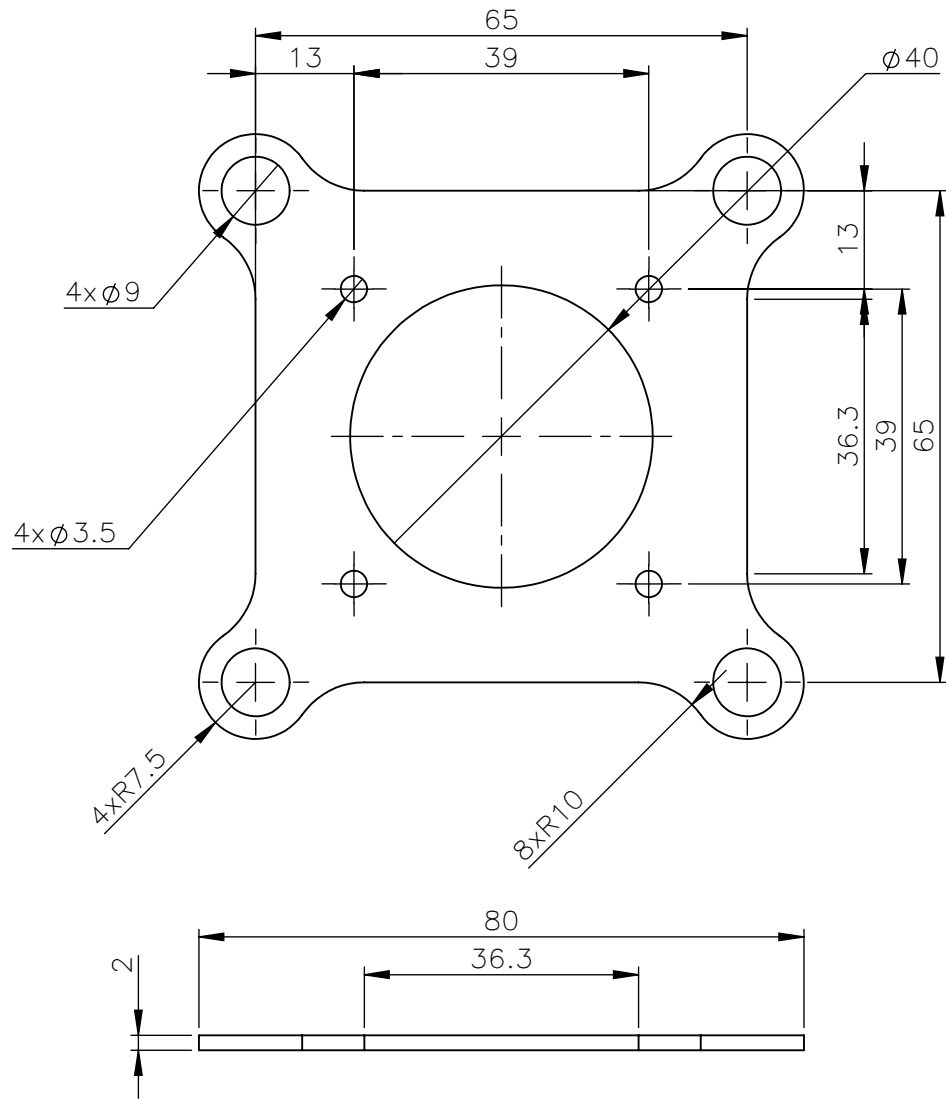
SHOP DRAWINGS						
N°	ZONE	DESCRIPTION	DRAWING AND/OR STANDARD	QTY.	MATERIAL	OBSERVATION
6	A-5	AERODYNAMIC PROFILE COVER	.106	1	ABS PLASTIC	
5	A-7	AERODYNAMIC PROFILE LINK	.105	1	ABS PLASTIC	
4	C-2	PITCH ROTATION LINK	.104	1	ABS PLASTIC	
3	A-3	ROLL ROTATION LINK	.103	1	ABS PLASTIC	
2	B-5	ANTI-VIBRATION PLATE	.102	2	ABS PLASTIC	
1	B-4	PROTECTIVE CASE	.101	1	ABS PLASTIC	

EPN	FACULTAD DE INGENIERÍA MECÁNICA		DWG	Ariel Herrera	SCALE: 1:5
			DIS.	Ariel Herrera	
	GIMBAL ASSEMBLY		REV.	Esteban Valencia, PhD	DATE: 25-03-22

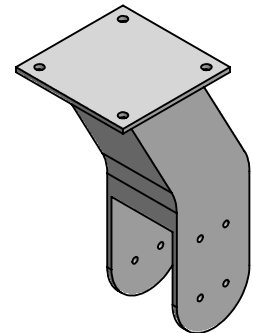
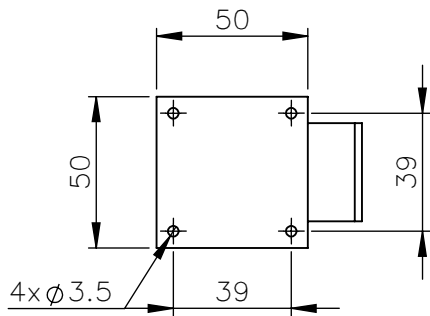
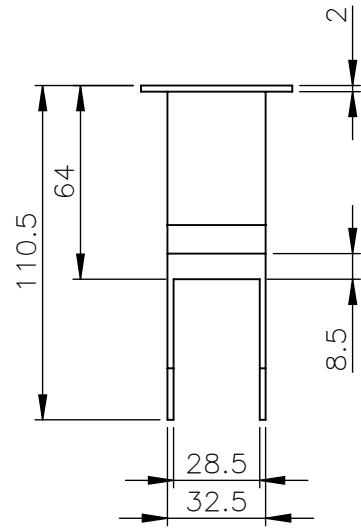
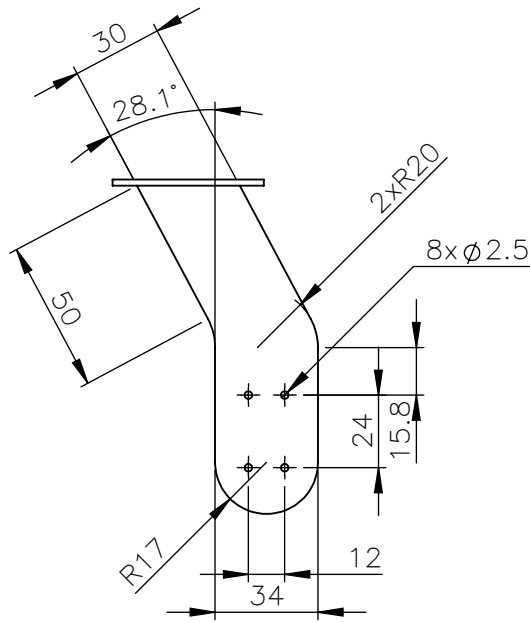
GUAV.001



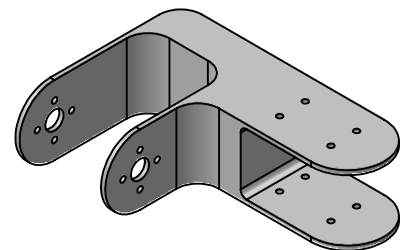
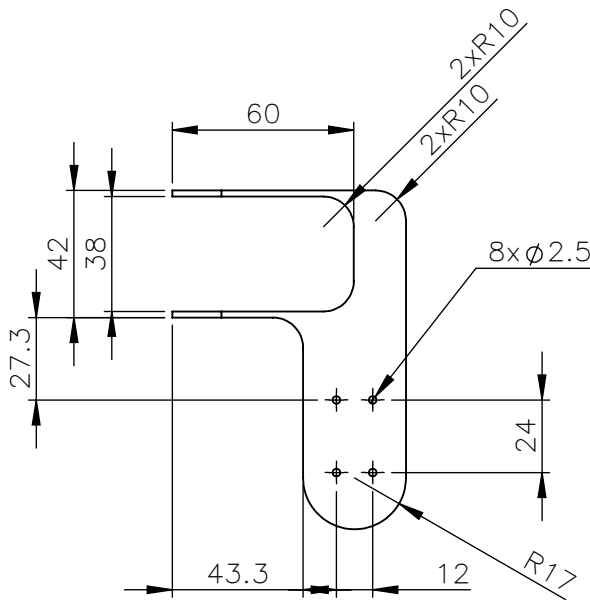
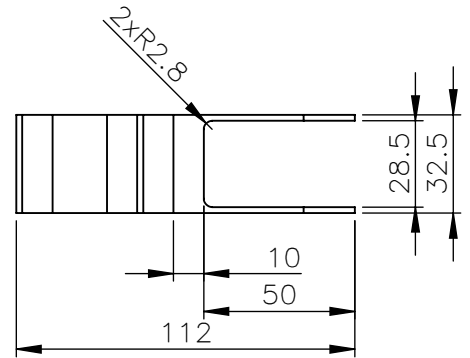
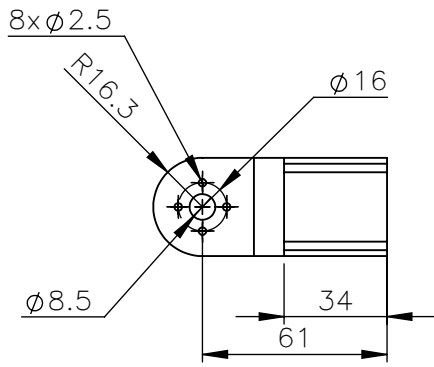
HEAT TREAT.:				EPN		FACULTAD DE INGENIERÍA MECÁNICA	
COATING:						DWG	Ariel Herrera
MATERIAL: ABS Plastic		O/A.Tol.: ±0.1		SCALE: 1:2.5		DIS.	Ariel Herrera
						REV.	Esteban Valencia, PhD
PROTECTIVE CASE				GUAV.101			DATE: 25-03-22



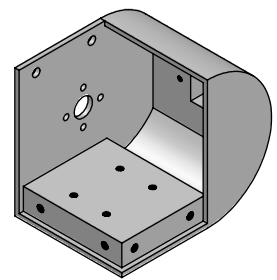
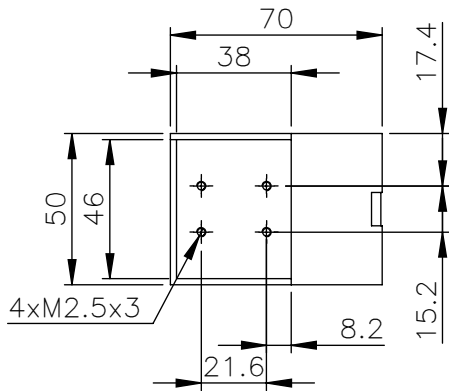
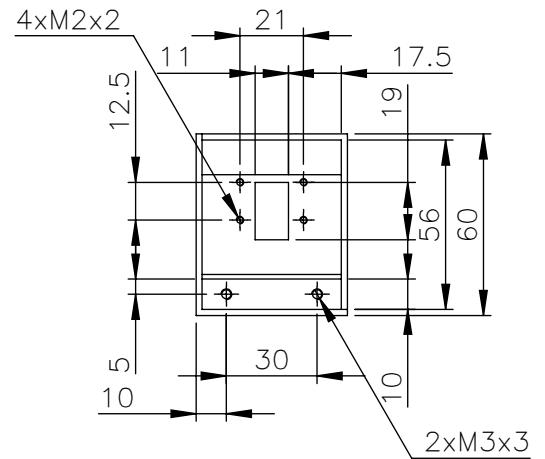
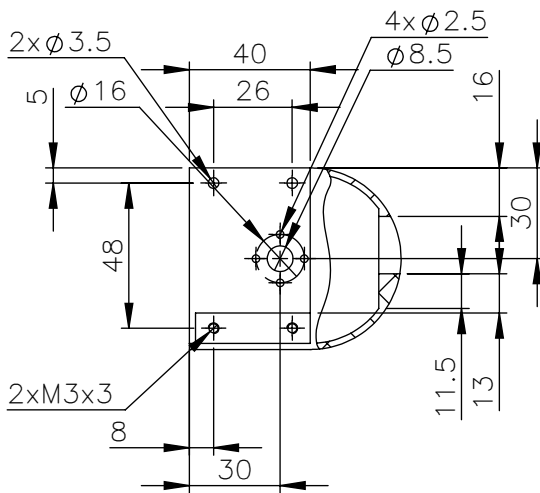
HEAT TREAT.:				EPN	FACULTAD DE INGENIERÍA MECÁNICA	
COATING:					DWG	Ariel Herrera
MATERIAL: ABS Plastic		O/A.Tol.: ±0.1		SCALE: 1:1	DIS.	Ariel Herrera
					REV.	Esteban Valencia, PhD
ANTI-VIBRATION PLATE					GUAV.102	



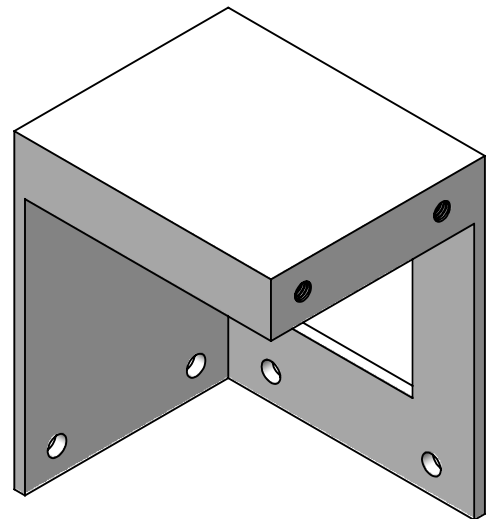
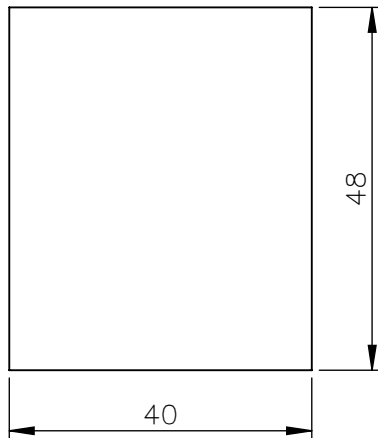
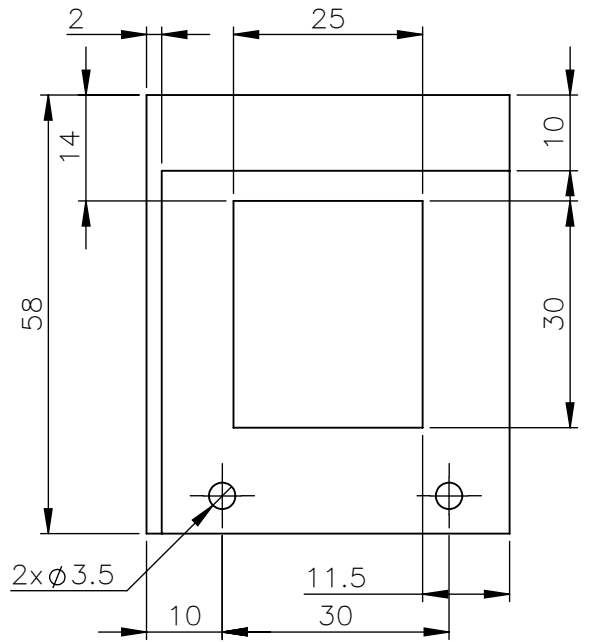
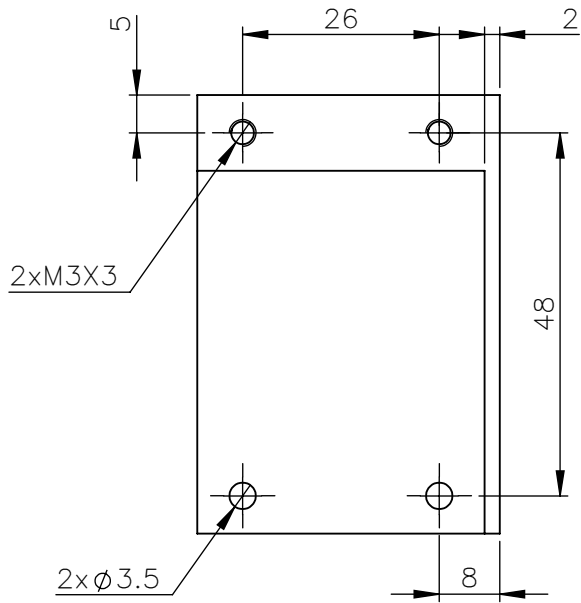
HEAT TREAT.:		EPN	FACULTAD DE INGENIERÍA MECÁNICA	
COATING:			DWG	Ariel Herrera
MATERIAL: ABS Plastic	O/A.Tol.: ±0.1	SCALE: 1:2.5	DIS.	Ariel Herrera
			REV.	Esteban Valencia, PhD
			DATE: 25-03-22	
ROLL ROTATION LINK		GUAV.103		



HEAT TREAT.:		EPN	FACULTAD DE INGENIERÍA MECÁNICA	
COATING:			DWG	Ariel Herrera
MATERIAL: ABS Plastic	O/A.Tol.: ±0.1	SCALE: 1:2.5	DIS.	Ariel Herrera
			REV.	Esteban Valencia, PhD
			DATE: 25-03-22	
PITCH ROTATION LINK		GUAV.104		



HEAT TREAT.:		EPN	FACULTAD DE INGENIERÍA MECÁNICA	
COATING:			DWG	Ariel Herrera
MATERIAL: ABS Plastic		O/A.Tol.: ±0.1	SCALE:	DIS.
			1:2.5	Ariel Herrera
			REV.	Esteban Valencia, PhD
AERODYNAMIC PROFILE LINK			GUAV.105	
			DATE: 25-03-22	



HEAT TREAT.:		EPN	FACULTAD DE INGENIERÍA MECÁNICA	
COATING:			DWG	Ariel Herrera
MATERIAL: ABS Plastic	O/A.Tol.: ±0.1	SCALE: 1:1	DIS.	Ariel Herrera
			REV.	Esteban Valencia, PhD
			AERODYNAMIC PROFILE COVER	
				DATE: 25-03-22