People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Saad Dahleb Blida1
Institute of Aeronautics and Space Studies
Aeronautical Construction Department

Manuscript

Toward the attainment of the Master's degree in Aeronautics

Option: **Avionics**

Design of an Inertially Stabilized Airborne Camera for UAV-based Active Object Tracking

Startup project presented as part of the "1275" ministerial decree, provided by Institute of Aeronautics and Space Studies.

Submitted by:

- **Ms. KADDOURI Hind Yasmine**

Supervised by:

- **Dr. KRIM Mohamed**
- **Mr. MEHAYA Mohammed Mehdi**

Economic Partner:

- **Mr. BELARBI Adel**

Trainer:

- **Dr. LEBSIR Abdelkader**

Jury Members:

Dr. DILMI Smain	President
Dr. AZMEDROUB Boussad	Examiner
Dr. HADJI Ahmed	Incubator Representative

Defended on September 18th, 2024

Design of an Inertially Stabilized Airborne Camera for UAV-based Active Object Tracking

Hind Yasmine KADDOURI

A thesis presented in partial fulfillment of the requirements for the degree of Master in Avionics in collaboration with TAER R&D.

Aeronautical Construction Department
Institute of Aeronautics and Space Studies
University of Saad Dahleb Blida1

I dedicate this work to:

My grandparents, whose prayers and love have always provided me with guidance and warmth.

My parents, whose tireless support, wisdom, and unwavering belief in me have been my greatest strength.

My brother and sister, for their humor, inspiration, and constant motivation to strive for better.

My family, who stood by my side through every challenge, offering love and encouragement.

My friends, whose support and encouragement have made this journey all the more fulfilling.

My professors and mentors, for their invaluable guidance, shaping both my academic and personal growth.

Acknowledgement

In the name of Allah, the Most Gracious, the Most Merciful. All praise is due to Allah for granting me the strength, patience, and guidance to complete this modest work. Without His blessings, none of this would have been possible.

I would like to express my sincere gratitude to my supervisors, **Dr. Krim Mohamed** and **Mr. Mehaya Mohammed Mehdi**, for their unwavering support, guidance, and encouragement throughout this project. Their feedback was instrumental in shaping the direction of this work and in helping me grow both academically and personally.

I am also deeply grateful to **Mr. Adel Belarbi**, Director of **TAER R&D**, and the entire team, particularly **Mr. Boutemedjet A.**, whose partnership and feedback significantly contributed to the success of this study.

A special thanks to the engineers at **URAT** (Unité de Rénovation des Aéronefs de Transport – Boufarik), **ERMAéro** (Etablissement de Rénovation des Matériels Aéronautiques), **GAPC** (Groupement Aérien de la Protection Civile), and **Air Algérie**, whose expertise and insights during my internships played a crucial role in broadening my knowledge and gaining a deeper insight into the field.

I would like to express my deepest gratitude to my family, whose love, patience, sacrifices and constant encouragement have been my source of strength.

Finally, my heartfelt thanks go to my dear friends and colleagues for their motivation, shared knowledge, and steadfast support during the most challenging times of this endeavor. Your presence made this experience all the more meaningful, and I will always cherish the memories we created together.

Abstract

This study presents the design, development, and evaluation of an inertially stabilized airborne camera control system utilizing a 3-axis gimbal for UAV-based active object tracking.

The proposed system is designed to control three servomotors, which manage the yaw, pitch, and roll movements of the camera, using measurements from an Inertial Measurement Unit (IMU) with six degrees of freedom model that reflect the vehicle's position in 3-axis motion. The primary objective of the study is to achieve stabilization against disturbances induced by vehicle motion, turbulence, and other external factors that could cause image distortion and impact computer vision tasks. IMU angle measurements were processed using a complementary filter to enhance accuracy, and three Proportional-Integral-Derivative (PID) controllers were implemented to independently manage each servomotor. While the system demonstrated effective stabilization on the roll and pitch axes, limitations were observed on the yaw axis, primarily due to gyroscope drift and the absence of a magnetometer. This issue has been identified as an area for future improvement.

In addition to stabilization, object detection based on image processing capabilities has been developed to achieve the active tracking. Due to the computational limitations of the Raspberry Pi, which served as the system's core processing unit, neural network-based models were found unsuitable for real-time operations. Consequently, a lightweight color-based detection algorithm was employed, achieving a higher frame rate. The system exhibited reliable tracking performance, with a small Root Mean Square Error (RMSE), following the implementation of Proportional-Derivative (PD) controllers. In the final phase, a prototype named Aero Vision 0.1 was constructed and tested, utilizing the control system for both stabilization and tracking processes. Despite some limitations, the overall performance of the gimbal was deemed satisfactory, providing a solid foundation for further development.

Keywords: Inertial Stabilization, 3-Axis Gimbal, UAV, Object Detection,Active Object Tracking, IMU, PID Controller, Visual Servoing, Complementary Filter.

Résumé

Cette étude présente la conception, le développement et l'évaluation d'un système de contrôle de caméra aéroportée stabilisée par inertie utilisant un cardan à trois axes pour le suivi actif d'objets par drone.

Le système proposé est conçu pour contrôler trois servomoteurs qui gèrent les mouvements de lacet, de tangage et de roulis de la caméra, en utilisant les mesures d'une unité de mesure inertielle (IMU) avec un modèle à six degrés de liberté qui reflète la position du véhicule dans un mouvement à trois axes. L'objectif principal de l'étude est de parvenir à une stabilisation contre les perturbations induites par les mouvements du véhicule, les turbulences et d'autres facteurs externes susceptibles de provoquer une distorsion de l'image et d'affecter les tâches de vision par ordinateur. Les mesures d'angle de l'IMU ont été traitées à l'aide d'un filtre complémentaire pour améliorer la précision, et trois contrôleurs proportionnels-intégraux-dérivés (PID) ont été mis en œuvre pour gérer indépendamment chaque servomoteur. Alors que le système a démontré une stabilisation efficace sur les axes de roulis et de tangage, des limitations ont été observées sur l'axe de lacet, principalement en raison de la dérive du gyroscope et de l'absence d'un magnétomètre. Cette question a été identifiée comme un domaine à améliorer à l'avenir.

Outre la stabilisation, la détection d'objets basée sur les capacités de traitement d'images a été développée pour réaliser le suivi actif. En raison des limites de calcul du Raspberry Pi, qui a servi d'unité centrale de traitement du système, les modèles basés sur les réseaux neuronaux ont été jugés inadaptés aux opérations en temps réel. Par conséquent, un algorithme de détection léger basé sur les couleurs a été utilisé, permettant d'atteindre une fréquence d'images élevée. Le système a présenté des performances de suivi fiables, avec une faible erreur quadratique moyenne (RMSE), suite à la mise en œuvre de contrôleurs proportionnels-dérivés (PD). Dans la phase finale, un prototype nommé Aero Vision 0.1 a été construit et testé, utilisant le système de contrôle pour les processus de stabilisation et de suivi. Malgré certaines limitations, les performances globales du cardan ont été jugées satisfaisantes, ce qui constitue une base solide pour la poursuite du développement.

Keywords: Stabilisation Inertielle, Nacelle à 3 axes, Drone, Détection d'Objets, Suivi Actif d'Objets, IMU, Contrôleur PID, Asservissement Visuel, Filtre Complémentaire.

ملخص

تقدم هذه الدراسة تصميم وتطوير نظام تحكم في الكاميرا المحمولة جواً بالقصور الذاتي باستخدام حامل ثلاثي المحاور لتتبع الأجسام بواسطة الطائرات بدون طيار.

صُمم النظام المقترح للتحكم في ثلاثة محركات مؤازرة تتحكم في حركات انحراف الكاميرا وميلها ولفها باستخدام قياسات من وحدة قياس القصور الذاتي ذات ست درجات من الحرية التي تعكس موقع المركبة في حركة ثلاثية المحاور. الهدف الرئيسي من الدراسة هو تحقيق الاستقرار ضد الاضطرابات الناجمة عن حركة المركبة والاضطرابات والعوامل الخارجية الأخرى التي يمكن أن تشوه الصورة وتؤثر على مهام الرؤية الحاسوبية. تم معالجة قياسات الزوايا لوحدة القياس بالقصور الذاتي باستخدام مرشح تكميلي لتعزيز الدقة، وتم تنفيذ ثلاث وحدات تحكم تناسبية تكاملية مشتقة للتحكم بشكل مستقل في كل محرك. في حين أظهر النظام استقرارًا فعالاً على محوري الدوران والميل، لوحظت قيود على محور الانعراج، ويرجع ذلك أساساً إلى انحراف الجيروسكوب وغياب البوصلة الالكترونية. وقد تم تحديد ذلك كمجال للتحسين في المستقبل. بالإضافة إلى التثبيت، تم تطوير و تجريب نظام لاكتشاف الأجسام استناداً إلى قدرات معالجة الصور للتتبع. ونظراً للقيود الحسابية الخاصة بوحدة المعالجة المركزية للنظام، اعتُبرت النماذج القائمة على الشبكات العصبية غير مناسبة لعمليات الوقت الحقيقي. ونتيجة لذلك، تم استخدام خوارزمية كشف خفيفة قائمة على كشف الألوان، مما أتاح تحقيق معدل إطارات يبلغ ٦٠ إطاراً في الثانية. أظهر النظام أداءً موثوقًا في التتبع، مع انخفاض جذر متوسط مربع الخط، بعد تنفيذ وحدات التحكم التناسبيةالمشتقة. في المرحلة النهائية، تم بناء واختبار نموذج أولي باستخدام نظام التحكم لعمليات التثبيت والتتبع. وعلى الرغم من بعض القيود، فقد وُجد أن الأداء العام للجهاز مُرضٍ، مما يوفر أساسًا متينًا لمزيد من التطوير.

الكلمات المفتاحية: التثبيت بالقصور الذاتي، حامل ثلاثي المحاور، طائرة بدون طيار، الكشف، التتبع النشط للأجسام، وحدة قياس القصور الذاتي، وحدة تحكم ، التوجيه البصري، المرشح التكميلي

Contents

Acknowledgement **iv**

Abstract **v**

Contents **x**

List of Figures **xiii**

List of Tables **xiv**

List of Abbreviations **xiv**

General Introduction **1**

I Review of Integrated Systems for Stabilization, Object Detection and Tracking in Dynamic Aerial Environments **4**

I.1 Introduction 4

I.2 Biological Vision 5

I.2.1 The Visual System 5

I.2.2 The Vestibular System 6

I.3 Image Stabilization Techniques 8

I.3.1 Inertially Stabilized Camera 10

I.3.2 Principle of Operation of a Gimbal 11

I.3.3 Related Works 12

I.4 Image Processing 14

I.4.1 Digital Image 14

I.4.2 Color Spaces 15

I.5 Object Detection and Tracking 19

I.5.1 Object Detection 19

I.5.2 Object Tracking 22

I.5.3 Related Works 23

I.6 Conclusion 24

II Theoretical Foundations of Data Acquisition, Filtering, and Control Systems 25

II.1 Introduction 25

II.2 Data Aquisition and Filtering 26

II.2.1 The MPU6050 26

II.2.2 Complementary Filter theory 32

II.3 Visual servoing 34

II.3.1 Position-Based Visual Servoing (PBVS) 34

II.3.2 Image-Based Visual Servoing (IBVS) 34

II.4 Control theory 36

II.4.1 PID controller overview 36

II.4.2 Components of a PID controller 37

II.4.3 Tuning a PID Controller 38

II.5 Conclusion 41

III System Design and Implementation 42

III.1 Introduction 42

III.2 Methodology 43

III.3 System Architecture Overview 44

III.3.1 Mechanical Design 44

III.3.2 Hardware Components Selection 46

III.3.3 Software and Development Tools 50

III.4 Prototype 51

III.5 The Camera Stabilization 52

III.5.1 IMU data Processing and Calibration 52
III.5.2 Implementing the Complementary Filter for IMU Data Fusion . . . 58
III.5.3 Integration of Filtered Data into Control Algorithm 59
III.6 Object Detection 62
III.6.1 Integration of Visual Data into Control Algorithm 65
III.7 Graphical User Interface (GUI) for Gimbal Control 68
III.8 Discussion 69
III.9 Conclusion 71

General Conclusion **72**

Appendix **79**

List of Figures

I.1 Human visual system 5
I.2 Schematic diagram of the biological vision system [57]. 6
I.3 The vestibulo-ocular reflex. 7
I.4 Image stabilization systems 8
I.5 Schematics of a 3-axes gimbal system [56]. 11
I.6 Image processing steps. 14
I.7 Digital image representation in grayscale by pixels [10]. 14
I.8 RGB color space. 15
I.9 Grayscale color space. 16
I.10 HSV color space. 16
I.11 lab color space. 17
I.12 Relationship between CV, AI, ML and DL. 17
I.13 Machine learning and deep learning performance in general with the amount of data [44]. 18
I.14 Classification of object detection methods [38]. 19
I.15 Deep learning-based object detection stages. 20
I.16 YOLO versions over time. 21

II.1 The MPU6050 3-axis. 26
II.2 The MEMS accelerometer. 27
II.3 The MEMS gyroscope 29
II.4 The coriolis force. 29

II.5 Capacitance variation due to Coriolis Forces in oscillating masses. 30
II.6 The Complementary Filter [37] 32
II.7 Representation of Pan and Tilt Errors in Object Localization. 35
II.8 Block Diagram of a PID Controller System. 37
II.9 Transient response parameters. 39

III.1 Design methodology flowchart 43
III.2 Gimbal exploded and isometric views. 44
III.3 Gimbal side and front views. 45
III.4 Gimbal holder. 45
III.5 Raspberry Pi 4 model B. 46
III.6 Raspberry Pi Camera V2. 46
III.7 MG996R Servomotor. 46
III.8 PCA9685. 47
III.9 Raspberry Pi Pico W. 47
III.10MPU6050. 47
III.11System Architecture. 48
III.12The prototype Aero Vision 0.1 51
III.13MPU6050 Sensor with Raspberry Pi Pico W. 52
III.14Gyroscope raw data readings from MPU6050 before calibration. 52
III.15Visualization of gyroscope signals before calibration. 53
III.16Accelerometer raw data readings from MPU6050 before calibration. 53
III.17Visualization of the accelerometer signals before calibration. 53
III.18Impact of vibrations on accelerometer readings. 54
III.19Final gyroscope calibration offsets. 56
III.20Gyroscope raw data readings from MPU6050 after calibration. 56
III.21Visualization of gyroscope signals after calibration. 56
III.22Gyroscope Drift over time. 57
III.23Accelerometer Noise over time. 57
III.24Comparison of angle estimations: Gyroscope vs Accelerometer. 57
III.25The filtered Roll angle signal using complementary filter. 58

III.26The filtered Pitch angle signal using complementary filter. 58

III.27Stability enhancement of Roll and Pitch angles using the complementary filter. 59

III.28Drift correction of Roll and Pitch angles using the complementary filter. . . 59

III.29Flowchart of the Stabilization Algorithm. 60

III.30Results of stabilization control. 61

III.31Detection results using YOLOv8n model at a resolution of 300 × 300 pixels. 62

III.32Detection results using MobileNet SSD at a resolution of 300 × 300 pixels. 63

III.33Detection results using MobileNet SSD at a resolution of 160 × 160 pixels. 63

III.34Detection of Various Colors Based on the Defined HSV Ranges. 64

III.35Flowchart of the Tracking Algorithm. 66

III.36Results of tracking control. 67

III.37Center of the object over time. 67

III.38Graphical User Interface (GUI) for Gimbal Control. 69

List of Tables

I.1 The different types of gimbals. 11

I.2 Comparison between Complementary filter and Kalman filter [11]. 13

II.1 Advantages and Limitations of the Complementary Filter 33

II.2 Advantages and Limitations of PBVS and IBVS 34

II.3 Effects of Increasing Controller Parameters on System Performance. 38

II.4 Summary of Optimization Methods for PID Tuning 40

III.1 Gimbal Electronics 46

III.2 Software and Development Tools Used. 50

List of Abbreviations

AI	**A**rtificial **I**ntelligence
ANN	**A**rtificial **N**eural **N**etwork
CV	**C**omputer **V**ision
CNN	**C**onvolutional **N**eural **N**etworks
DL	**D**eep **L**earning
DS	**D**igital **S**tabilization
ES	**E**lectronic **S**tabilization
FOV	**F**ield **O**f **V**iew
FPS	**F**rame **P**er **S**econd
HS	**H**ybrid **S**tabilization
HSV	**H**ue, **S**aturation, **V**alue
ISP	**I**nertially **S**tabilized **P**latform
IMU	**I**nertial **M**easurement **U**nits
I2C	**I**nter-**I**ntegrated **C**ircuit
LOS	**L**ine **O**f **S**ight
ML	**M**achine **L**earning
MOT	**M**ultiple **O**bject **T**racking
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystems
OS	**O**ptical **S**tabilization
PWM	**P**ulse **W**idth **M**odulation
RMSE	**R**oot **M**ean **S**quare **E**rror
SOT	**S**ingle **O**bject **T**racking
UAV	**U**nmanned **A**erial **V**ehicles
UART	**U**niversal **A**synchronous **R**eceiver/**T**ransmitter

General Introduction

In recent years, Unmanned Aerial Vehicle (UAV) technology has seen considerable advancements, with various organizations, including private companies and research centers, developing sophisticated systems for both military and civilian applications [36] [32].

Military UAVs often employ airborne radars or guided seekers for target acquisition and localization. However, these systems are typically heavy and expensive, making them impractical for civilian UAVs. Among civilian platforms, quadrotors have gained significant popularity due to their stability and lower manufacturing costs compared to helicopters. Despite this, most quadrotors, with a takeoff weight of under 15 kg, have limited payload capacity and battery endurance [30]. These constraints present challenges in accurately and reliably tracking moving targets, particularly because of the limited computational capabilities and low-cost sensors onboard.

Cameras, being affordable, lightweight, and capable of passively capturing environmental data, have emerged as promising tools for UAV-based target tracking. These features are particularly advantageous for advanced aerial imagery applications, which have prompted substantial research into enhancing camera-based tracking methods for UAVs [30]. Such advancements enable the rapid detection of targets over extensive areas, followed by continuous surveillance of selected targets during the tracking phase.

Various approaches can be used to actively track a target object with a UAV. These include adjusting the UAV's position, altering the camera's orientation using a pan[1] and tilt[2] mechanism, or deploying multiple cameras.

This study focuses on active object tracking using a gimbal system, which is a key component of inertially stabilized platforms (ISPs) used in various mobile carriers, such as manned and unmanned aerial vehicles (MAVs/UAVs) and helicopters [36]. These systems, typically comprising two or three axes, stabilize the sensor against disturbances from the vehicle, turbulence, and other external factors that can induce image distortion [31].

[1] **Pan:** The horizontal movement of the camera, allowing it to rotate left or right.

[2] **Tilt:** The vertical movement of the camera, enabling it to rotate up or down.

Gimbals are employed in a wide range applications to keep the target centered within the image frame including: traffic surveillance, reconnaissance, and target acquisition (RSTA) [30], as well as search and rescue (SAR), where they aid in locating missing individuals or assets in unsafe geographical areas that are otherwise inaccessible to search and rescue teams [35] [51].

Additionally, these systems are utilized for visual inspections of structures and components in remote or difficult-to-access locations, where such tasks are typically expensive, dangerous, and time-consuming. Examples include inspections of power lines [45], wind turbines, transportation infrastructure, buildings, industrial sites, and mines [17], all while maintaining the object of interest within the camera's field of view (FOV). Further applications extend to aerial photography, missile tracking, autonomous navigation, and agricultural seeding and monitoring.

This technology is not only pivotal for fostering innovation but also plays a significant role in economic development, particularly within emerging markets such as Algeria. However, Algeria currently lacks domestic manufacturers capable of producing imaging systems tailored for UAV projects. This gap is especially crucial for military UAV systems, where the domestic production of all components is strategically significant. Furthermore, for civilian applications, local production of these imaging systems would bring substantial economic benefits, particularly when integrated with domestically manufactured UAVs.

This study aims to develop a prototype of a stabilized camera control system with a 3 xis-gimbal, achieving the following objectives:

- Stabilization of the roll, pitch, and yaw axes against external perturbations.
- Object tracking functionality, ensuring that the camera's line-of-sight (LOS) is directed at the target and remains centered within the camera's field of view (FOV).

The project is structured into several tasks to meet the outlined objectives. First, it requires a thorough understanding of control theory to manage the three motors of the gimbal. Additionally, studying methods for obtaining visual data and integrating them into the control system is essential to achieve the tracking functionality. Furthermore, a comprehensive understanding of IMU data processing is required, along with the integration of these data into the control system to ensure stabilization, all while considering the available hardware resources.

This thesis explores the theoretical foundations, design methodologies, and practical implementations of the system under study, and it is structured into three main chapters:

Chapter I reviews the current state-of-the-art in image stabilization techniques, including the control theory, as well as the object detection and tracking systems. By examining various technological approaches and algorithms, this chapter lays the foundation for the practical development of a real-time object tracking system with high precision. Key factors such as material properties and available computational resources are considered to ensure optimal performance, particularly in maintaining a high frames-per-second (FPS) rate for real-time applications.

Chpter II provides an in-depth analysis of sensor data integration using the IMU sensor and complementary filters for accurate data processing. Visual servoing techniques are explored to enable vision-based adjustments in the system's motion control. The theory of PID controllers in stabilizing and fine-tuning system responses is emphasized, laying the groundwork for practical implementation.

Chapter III focuses on the practical application of the concepts discussed in earlier chapters through prototype development and testing. The implementation phase involves the successful processing of IMU data, calibration, and data fusion using a complementary filter to ensure system stability. Detection was done by YOLOv8n model, MobileNet SSD and color-based detection then integrate the visual data into control algorithm for effective object tracking. The chapter concludes with an assessment of the system's performance, identifying future opportunities for improvement.

At the end, the results obtained in the study and future perspectives are summarized in the conclusion.

The appendix is dedicated to the presentation of the Aero Vision startup project along with a technical sheet.

Chapter I

Review of Integrated Systems for Stabilization, Object Detection and Tracking in Dynamic Aerial Environments

I.1 Introduction

In recent years, there has been a growing demand for high-quality imagery in various fields, particularly in aerial applications. Achieving stable and accurate images in dynamic environments, such as during flight, is a significant challenge. This challenge has driven the development of stabilization and object tracking technologies, which are critical for applications such as surveillance, navigation, and search and rescue. This chapter begins with an overview of biological vision, which serves as inspiration for understanding computer vision. It then explores the state-of-the-art in image stabilization, object detection, and object tracking, examining the various techniques and systems developed to ensure stable, high-quality images on airborne platforms.

I.2 Biological Vision

I.2.1 The Visual System

The human eye is an extraordinary optical organ with the ability to focus light and adjust to different lighting conditions. However, it is just one component of the intricate process of vision.

The visual system serves as the physiological foundation for visual perception, allowing individuals to detect and process light. This system plays a crucial role in capturing, transducing, and interpreting information related to light within the visible spectrum. By doing so, it constructs an image and creates a mental representation (or a mental model) of the surrounding environment [2].

As shown in figure I.1, the visual system encompasses both the **optical system**, which includes components like the cornea and lens, and the **neural system**, which comprises structures such as the retina and visual cortex. These two systems work together harmoniously to facilitate the complex process of vision, ultimately enabling individuals to perceive and make sense of their surroundings [2].

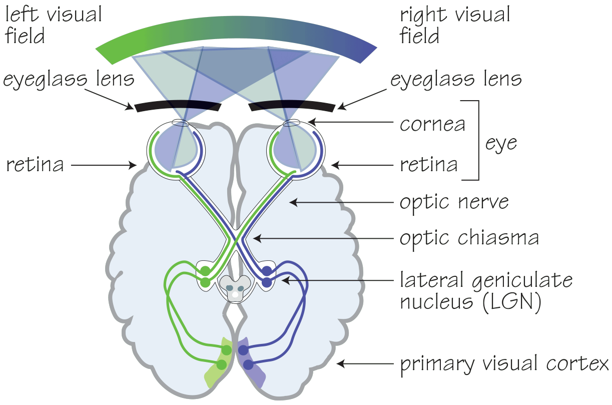


Figure I.1: Human visual system

Vision is achieved through a coordinated effort involving three distinct stages: optic, retinal, and neuronal, so it is the incredible result of highly coordinated teamwork [2].

The eye forms an image of the world on the retina during the optic stage. Specialized cells in the retina convert light energy into electrical signals during the retinal stage. Finally, the neuronal stage involves the brain's processing and interpretation of neural signals, resulting in the conscious experience of vision. This collaborative effort between the eye's intricate design and the brain's processing power highlights the true complexity of human vision[2].

To round up the concept, figure I.2 depicts the process:

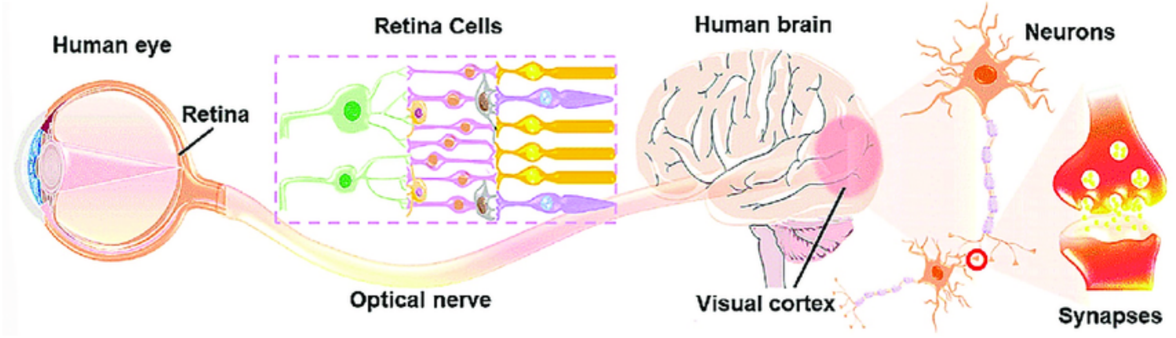


Figure I.2: Schematic diagram of the biological vision system [57].

I.2.2 The Vestibular System

The vestibular system, located within the inner ear, serves as a biological equivalent to the accelerometers employed in camera image stabilization systems.This system enables visual stability in humans and numerous animal species by maintaining gaze fixation during head movements, which involve rotations and translations [1].

The vestibular system consists of two components: the semicircular canals [1], which detect rotational movements, and the otoliths (ear-stones) [2], which detect linear accelerations. The signals sent by the vestibular system mainly go to the neural structures that control eye movement, forming the anatomical basis of the vestibulo-ocular reflex, which is crucial for clear vision [1].

The VOR, or vestibular-ocular reflex, is a reflexive eye movement that serves to stabilize images on the retina during head movement. This is achieved by producing an eye movement in the opposite direction of head movement, thereby preserving the image on the center of the visual field. For instance, when the head moves to the right, the eyes move to the left, and vice versa. Given that head movements are consistently present, the VOR plays a critical role in maintaining visual stability: patients who experience impaired VOR often struggle with reading, as they are unable to stabilize their eyes during minor head tremors. Notably, the VOR reflex is not contingent upon visual input and operates effectively total darkness or when the eyes are closed [1].

A closer look at figure I.3 reveals that upon detection of head rotation, the vestibular system transmits inhibitory signals to the extraocular muscles on one side and excitatory signals to the muscles on the opposite side.

[1] The semicircular canals are three semicircular linked tubes found in the deepest region of each ear. There are three semicircular canals: lateral, anterior, and posterior which provide the sensation of angular acceleration.

[2] These organs are a small calcium carbonate structures in the saccule or utricle of the inner ear that allows an organism, including humans, to perceive linear acceleration, both horizontally and vertically (gravity).

This coordinated neural response elicits compensatory eye movements, counteracting head motion and preserving a stable image on the retina. Notably, these compensatory eye movements exhibit minimal latency, typically occurring within 10 milliseconds of head movement, highlighting the remarkable efficiency of this biological stabilization mechanism[1].

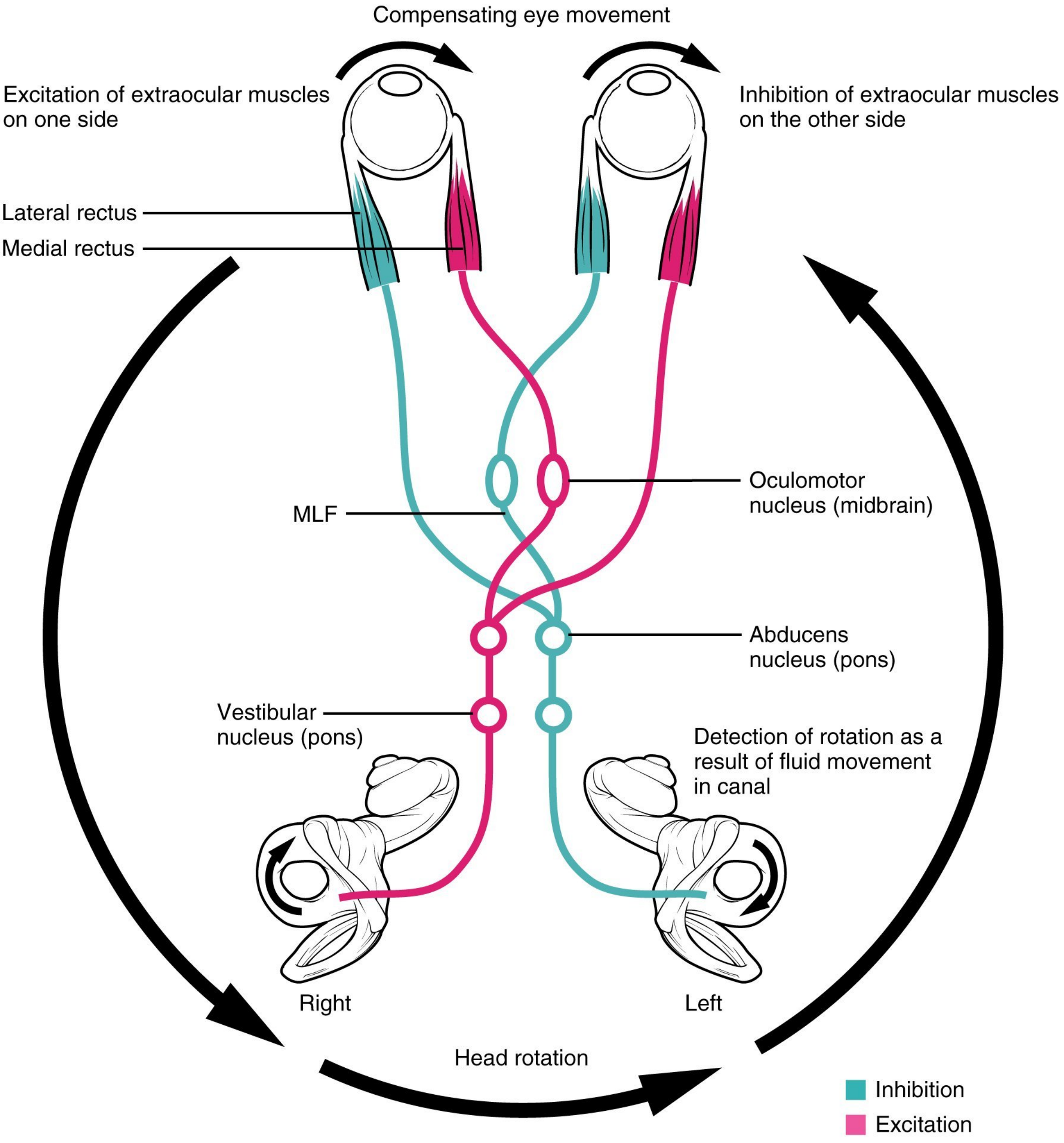

Compensating eye movement
Excitation of extraocular muscles on one side
Inhibition of extraocular muscles on the other side
Lateral rectus
Medial rectus
MLF
Oculomotor nucleus (midbrain)
Abducens nucleus (pons)
Vestibular nucleus (pons)
Detection of rotation as a result of fluid movement in canal
Right
Left
Head rotation
🟦 Inhibition
🟥 Excitation

Figure I.3: The vestibulo-ocular reflex.

This process of adaptation and rapid image processing by the biological brain inspires us in the design of artificial vision systems, from the image processing stage to the stabilization actions that combine gyroscope and accelerometer data.

I.3 Image Stabilization Techniques

Image stabilization plays a crucial role in enhancing the accuracy and reliability of object detection and tracking. Stabilization systems has gone through several stages: mechanical, optical, digital and hybrid stabilization, each tailored to address varying degrees of instability.

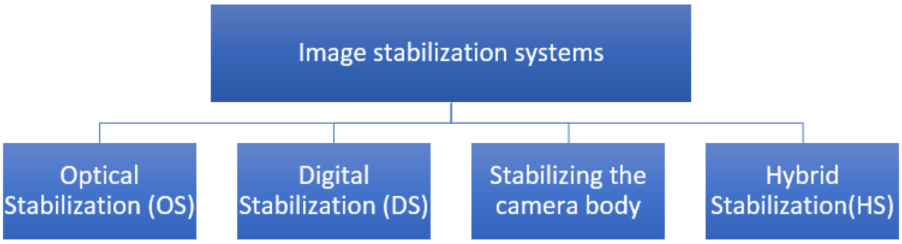


Image stabilization systems
Optical Stabilization (OS)
Digital Stabilization (DS)
Stabilizing the camera body
Hybrid Stabilization(HS)

Figure I.4: Image stabilization systems

The majority of cameras in use today are equipped with either ES (Electronic Stabilization) or OS (Optical Stabilization) as their built-in stabilization systems [52]. Each tackles camera shake differently, offering distinct advantages and limitations.

Optical Stabilization (OS)

1. **In-Lens Stabilization**

   This mechanism employs a movable element within the lens itself, typically a group of lenses or a prism to adjust the optical path to the sensor. Gyroscopes detect camera shake, and the system adjusts the position of this element to counteract the movement, effectively keeping the image sensor stable[54]. This system offers excellent image quality, as it is compatible with all lenses attached to the camera and has minimal impact on power consumption. Major shocks or sudden movements are not absorbed by the optical elements. Compared to digital alternatives, it may be a pricier option and might not perform as effectively for very rapid movements.

2. **Sensor-Shift Stabilization or In Body Image stabilization (IBIS)**

   This approach stabilizes the image sensor itself rather than a lens element. By physically shifting the sensor in the opposite direction of camera shake, a clear image is captured. IBIS, commonly found in mirrorless cameras, offers a wider range of effectiveness against camera shake compared to in-lens stabilization. However, it may introduce slight image cropping due to the physical movement of the sensor and potentially consume more battery power.

The primary function of all optical stabilization systems is to stabilize the image projected on the sensor prior to the sensor's conversion of the image into digital information.

Digital Stabilization (DS)

Real-time digital image stabilization, also known as Electronic Image Stabilization (EIS), This software-based technique analyzes consecutive video frames to detect shake. The software then adjusts each frame slightly to compensate for the movement, essentially cropping and repositioning the image data electronically [54].

This method requires the resolution of the image sensor to exceed the resolution of the recorded video. It's generally less expensive than optical solutions, it can be applied to both photos and videos and functions with any camera. However, it's important to consider potential drawbacks, DS may reduce image resolution due to cropping and, in some situations, introduce artifacts like blurring or ghosting.

Additionally, it's often less effective than optical stabilization for compensating for larger camera movements, a minor movement or inaccuracy in the image will become highly apparent after amplification.

So DS can be applied after the image is taken, but it is impractical for large amplification images. And here come the Camera body stabilization system that may solve this issue[49].

As a result, these internal systems have a limited ability to generate entirely blur-free images or videos because their stabilization elements can resist small vibrations.

Camera Body Stabilization

Unlike internal methods, this technique stabilizes the entire camera body through an external device. An IMU [3](Inertial Measurement Unit)[36] detects camera shake, the information is then relayed to a motorized gimbal system (will be define in I.3.1) that physically counteracts the movement.

Hybrid Stabilization

As the name suggests, it means that the camera combines two or more stabilization technologies simultaneously.

[3] **IMU :** Inertial Measurement Unit is a critical component in navigation and control systems, providing essential data for determining an object's orientation and motion. It typically integrates the following types of sensors : Accelerometers, Gyroscopes and on occasion Magnetometers and Barometers.

I.3.1 Inertially Stabilized Camera

Inertially stabilized camera is a technology that utilizes IMU measurement to counteract unwanted rotations and ensure stability. This is particularly beneficial for objects like vehicles and camera systems [22]. This principle can be implemented in a device called a **gimbal**.

Gimbals are mechanical devices classified as a type of active stabilizer, commonly used in aerial vehicles to reduce vibrations and stabilize the camera in response to angular motion of the vehicle and other disturbances through inertial stabilization [8], particularly with the advancement of MEMS [4] technology in sensors [30], which offer high precision, low power consumption and low cost. This enhances stability and performance across various fields, including aerospace and military applications.

Furthermore, gimbals enhance the camera's Field of View (FOV) through angular motion, which can either compensate for or be integrated with translational motion to achieve the desired objective. This ability to address FOV limitations is especially important when tracking moving targets [8]. Active tracking using a gimbal device is, indeed, a highly compelling functionality in computer vision applications.

Gimbals come in various configurations depending on the number of axes they stabilize,here are the different types that exist for airborne cameras:

1. **Single-Axis Gimbal**

   This is the most basic and lightweight form. It utilizes a motor on a single axis, typically designed to counteract either pitch or roll movements. This approach is often found in smartphone gimbals or budget-friendly drone camera systems.

2. **Two-Axis Gimbal**

   This system represents a more advanced form. It employs motors on two axes to counteract pitch and roll movements. This approach is effective for mitigating mild vibrations and turbulence encountered during stable flight conditions. However, it may struggle in scenarios involving extreme maneuvers or strong winds.

3. **Three-Axis Gimbal**

   This system (The aim of this study) is the most versatile and commonly used option for airborne cameras. It incorporates motors on three axes, providing correction for pitch, roll, and yaw. This additional axis ensures exceptional stability even during rapid maneuvers or high winds. With a three-axis system, the camera maintains a level orientation, keeping the horizon straight and the footage smooth, even in challenging conditions. The figure I.5 below shows the three axes and rotations of a camera.

[4] **MEMS :** Micro-Electro-Mechanical Systems refer to a technology that encompasses miniaturized mechanical and electromechanical components fabricated using microfabrication techniques, typically ranging from a few micrometers to a few millimeters.

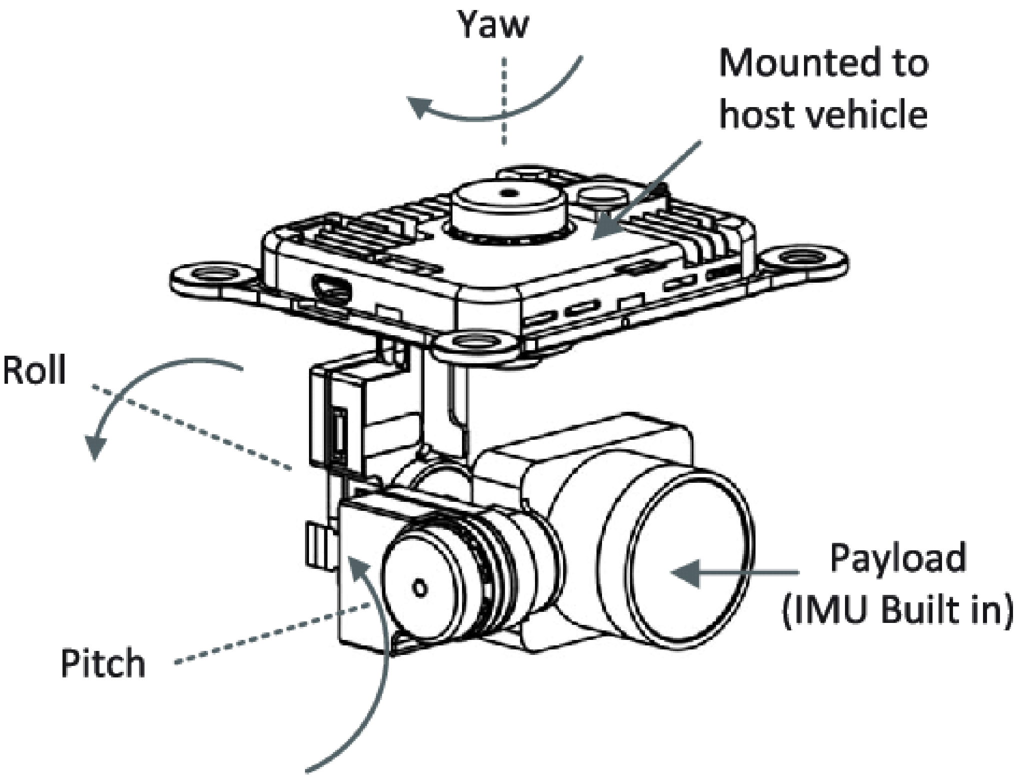


Figure I.5: Schematics of a 3-axes gimbal system [56].

The following table summarize the different types of gimbal systems for airborne camera, their advantages and limitations:

Table I.1: The different types of gimbals.

System Type	Axes Corrected	Advantages	Limitations
Single-Axis	Pitch or Roll	Simple design, lightweight, potentially lower cost	Limited effectiveness for combined movements
Two-Axis	Pitch & Roll	Simpler design than 3-axis, potentially lower cost	Limited effectiveness in extreme conditions
Three-Axis	Pitch, Roll & Yaw	Most versatile, excellent for various flight conditions	More complex design, potentially higher cost

I.3.2 Principle of Operation of a Gimbal

The fundamental concept of this system is to generate a counteracting motion in the direction opposite to the vibration. This is accomplished through a combination of sensors, processing units, and actuators, all functioning within a closed-loop control system, effectively reducing the impact of platform disturbances on the camera.

The inertial IMU sensor is responsible for providing these counteracting motions by measuring data hundreds of times per second. It detects movements, records angular velocities, accelerations, and magnetic forces, and then transmits this information to a microcontroller. The system uses three servo motors (or brushless motors), each operating independently on the three gimbal axes, to make real-time adjustments that counteract shakes by moving an equal amount in the opposite direction generated by a microcontroller.

The sensor continually monitors the camera's relative position to the ground. By comparing this with a predetermined optimal position, the system assesses how much the camera's position deviates with each movement. The primary goal is to maintain this optimal position.

I.3.3 Related Works

Many aspects of visual servoing and gimbal motion control have been studied for decades [24] including [20] and [22] as a part of Inertial Stabilized Platforms (ISPs), and those reviewed in the two-part tutorials series by Chaumette and Hutchinson [6] [7]. Numerous control algorithms have been explored in the literature [42]. Some of the well-known techniques include PID [36], Linear Quadratic Regulator (LQR), Sliding Mode Control (SMC) [28], Backstepping [24], H∞ [3], Fuzzy Logic [47], and Artificial Neural Networks. Each of these algorithms has its own distinct advantages and limitations [59].

After conducting a comprehensive analysis of various algorithms, the author in [59] concluded that the PID control algorithm is the most effective in terms of accuracy, simplicity, and feasibility. This conclusion is supported by its widespread use in industry, as noted in [36].

The data acquired from the IMU are processed and filtered using various techniques, including the Kalman filter and the Complementary filter [15] [11]. These filtering methods are employed to enhance the accuracy and reliability of the sensor data by reducing noise and improving the estimation of orientation and motion.

In [11], the author concludes that the choice between the complementary filter and the kalman filter depends on the specific application and available resources. The Complementary filter is ideal for scenarios requiring simplicity and real-time performance, such as basic robotics and motion tracking. In contrast, the Kalman filter is better suited for high-precision applications like aerospace and autonomous vehicles due to its higher accuracy and robustness to noise and sudden changes. His findings demonstrated that the complementary method using less computational resources delivers performance close to that of the more expensive Kalman filter without the aid of auxiliary sensors such as a digital compass.

Table I.2: Comparison between Complementary filter and Kalman filter [11].

	Complementary filter	**Kalman filter**
Algorithm complexity	Simple, easy to implement	More complex, requires more computational resources and expertise to implement
Accuracy	Low accuracy, particularly in noisy environments	High accuracy
Sensor fusion ability	Limited to combining data from two sensors, typically an accelerometer and a gyroscope	Can fuse data from multiple sensors, such as accelerometer, gyroscope, magnetometer, and GPS, allowing for more accurate tracking and compensation for various external factors
Real-time performance	Real-time processing is possible	Real-time processing is possible but can be limited by the complexity of the algorithm and the available computational resources
Robustness	Less robust to sudden changes and noise	More robust to sudden changes and noise due to its ability to estimate the noise and uncertainty in the system

The previous background serves to achieve our first objective. The second objective focuses on detecting the presence of the target in optical imagery within the camera's field of view (FOV) and subsequently actively tracking its position in the image plane, frame by frame, using the gimbal mechanism. Achieving this requires image processing, the integration of computer vision algorithms for object detection and tracking, and real-time control mechanisms to stabilize and continuously center the target.

I.4 Image Processing

Many researchers are currently focused on developing computer vision systems that simulate biological abilities, such as scene understanding, object detection. Computer vision, a key field within artificial intelligence (AI), enables computers to interpret and understand visual inputs like digital images and videos. By leveraging machine learning (ML) and deep learning (DL), these systems extract meaningful information, driving innovations across various applications.

Image processing forms the foundation of CV tasks, such as object detection and tracking. Before CV systems can interpret visual data, image processing steps are necessary to enhance image quality and isolate relevant features. This process, which involves importing, analyzing, manipulating, and extracting information from images , as shown in Figure I.6, is crucial for successful object detection and tracking.

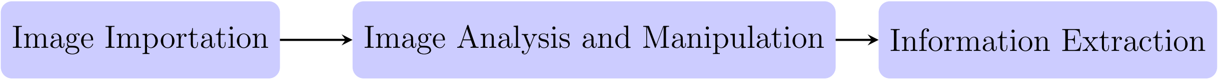
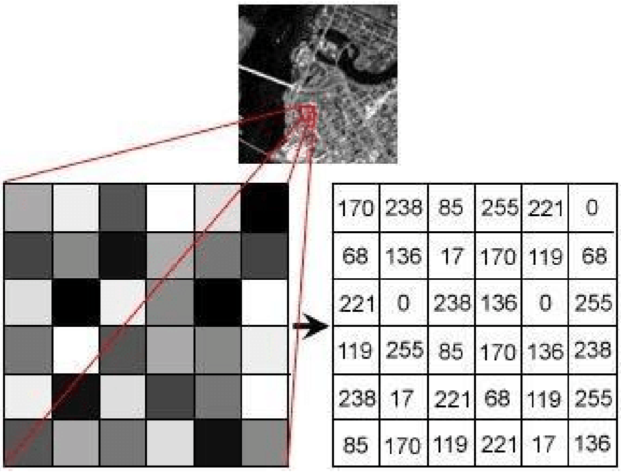


Figure I.6: Image processing steps.

I.4.1 Digital Image

An image is a two-dimensional representation of a visual object or scene, captured and stored as a digital or analog entity. In digital form, an image is typically composed of a grid of pixels [5] organized into rows (y) and columns (x) , where each pixel contains information about the color and intensity at that specific point.



170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

Figure I.7: Digital image representation in grayscale by pixels [10].

The resolution of an image, defined by the number of pixels along its width and height, determines the level of detail it can represent. Resolution is also related to pixel density,

[5] **Pixel :** is the smallest unit of a digital image, representing a single point in the image's grid.

typically measured in pixels per inch (PPI) for displays or dots per inch (DPI) for printed images, where 1 inch is equivalent to 2.54 centimeters. High PPI/DPI values indicate that pixels are densely packed within a given physical area of a display or printed image, resulting in greater image clarity and finer detail.

I.4.2 Color Spaces

The representation of images and the selection of appropriate color spaces are foundational components of computer vision systems, exerting a profound influence on a wide array of applications, including image processing and machine learning algorithms [55].

Color spaces are mathematical models that define how colors are expressed as sets of numerical values, usually comprising three or four components enabling precise specification of color values within an image.

The most widely recognized color spaces include grayscale, RGB, HSV and Lab. For instance, RGB is commonly used in display devices, whereas Lab is preferred for color analysis and perceptual uniformity [55].

RGB Color Space

The RGB (Red, Green, Blue) is an additive color space where colors are created by combining varying intensities of red, green, and blue light. Each component is typically represented by a value ranging from 0 to 255 [19].

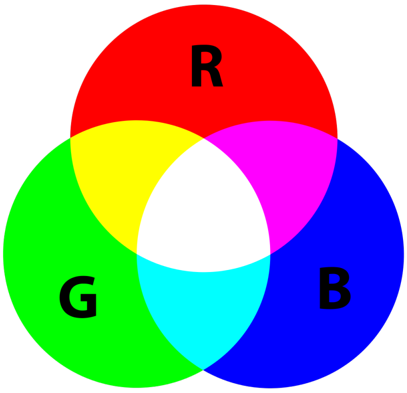


Figure I.8: RGB color space.

Grayscale Color Space

The grayscale color space is represented by varying shades of gray, from black to white, using a single channel. Each pixel's intensity value corresponds to its brightness, with no color information. In an 8-bit grayscale image, these values range from 0 to 255., where:

- 0 represents black.

- 255 represents white.

- Intermediate values represent various shades of gray.

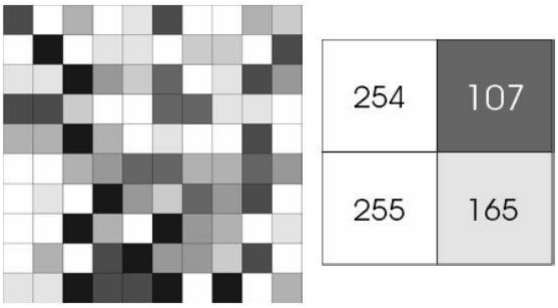


Figure I.9: Grayscale color space.

HSV Color Space

The HSV (Hue, Saturation, Value) color space is cylindrical-coordinate representation of points in an RGB color model. It is designed to be more intuitive for human perception and is often used in applications where color adjustments and selection are required [19].

- Hue represents the type of color.

- Saturation describes the intensity or purity of the color.

- Value represents the brightness or luminance.

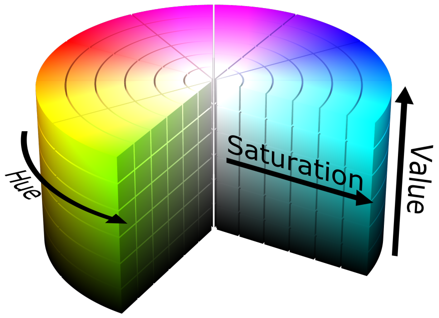


Figure I.10: HSV color space.

Lab Color Space

The Lab color space is a perceptually uniform color model widely used in color science, particularly for tasks requiring precise color analysis and differentiation. It ensures that numerical changes in color values correspond closely to visually perceived changes, making it a preferred choice for accurate color representation.[19]. The Lab color space comprises three components:

- L: Lightness, representing the brightness of the color.
- a: The component representing the green-red axis.
- b: The component representing the blue-yellow axis.

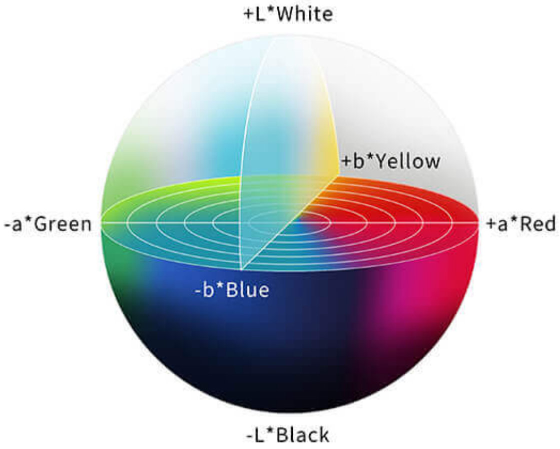


Figure I.11: lab color space.

Before proceeding, let's establish a clear understanding of some key terminology. Artificial intelligence, Machine learning, and Deep learning are commonplace terms in today's society, frequently used and often interchanged by companies attempting to market their products. However, these three terminologies are not interchangeable, as there are fundamental differences between them. The next figure illustrates the key distinctions and give an overview, how CV, AI, ML, and DL are different from each other or how these three terminologies are related to each other:

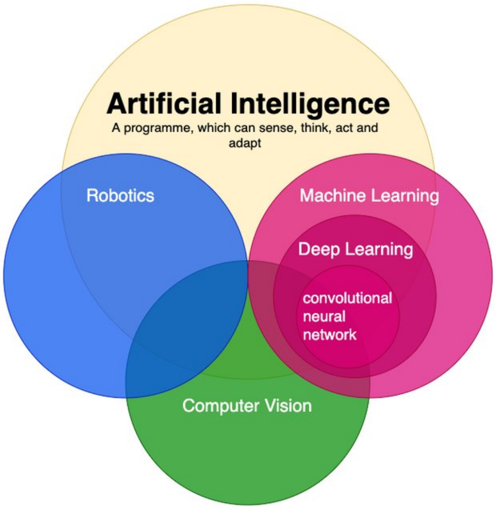

Artificial Intelligence
A programme, which can sense, think, act and adapt
Robotics
Machine Learning
Deep Learning
convolutional neural network
Computer Vision

Figure I.12: Relationship between CV, AI, ML and DL.

Artificial Intelligence (AI)

It is a branch of computer science focused on creating intelligent machines. These machines can learn, reason, and self-correct to perform tasks typically requiring human-like intelligence. AI can refer to either machine learning-based programs or even explicitly programmed computer programs. AI encompasses two primary approaches:

1. **Machine Learning (ML):** is a subfield of AI [58] where algorithms learn from data without explicit programming. This data can be in various formats like images, text, or numbers. By analyzing vast datasets, ML algorithms identify patterns and relationships within the data. This newfound knowledge empowers them to make predictions or informed decisions on new, unseen data. Machine learning fuels functions of computer vision such as recognition and tracking.

2. **Deep Learning (DL):** is a subfield of machine learning [58] inspired by the structure of the human brain. It utilizes multi-layered **Artificial Neural Networks (ANNs)**. These networks process information progressively through interconnected layers, mimicking how the brain learns. Deep learning models excel at handling complex data like images, speech, and natural language.

In simpler terms, all deep learning is a form of machine learning, but not all machine learning techniques involve deep learning models [26].

Besides, Machine Learning can often function with smaller datasets, especially when combined with effective feature engineering, unlike Deep Learning which typically requires large amounts of data to train complex models effectively.This fact is depicted in Figure I.13.

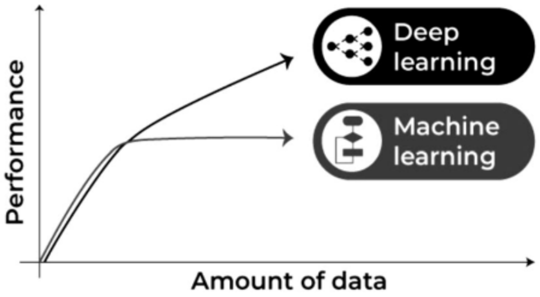


Figure I.13: Machine learning and deep learning performance in general with the amount of data [44].

I.5 Object Detection and Tracking

The rapid advancements in deep learning (DL) and the increasing computational power of GPUs have significantly enhanced the performance of state of the art object detectors and trackers.

I.5.1 Object Detection

Object detection, a prerequisite for initiating the tracking process, involves identifying and localizing objects within a single image or video frame, treating each frame independently without retaining memory of previous frames. Consequently, it can only detect objects that are visible in the current frame and may fail if the object is obscured or occluded.

Object Detection Approaches

According to the literature, object detection methods are generally classified into two categories: traditional techniques (also known as classical methods) and modern deep learning-based techniques.

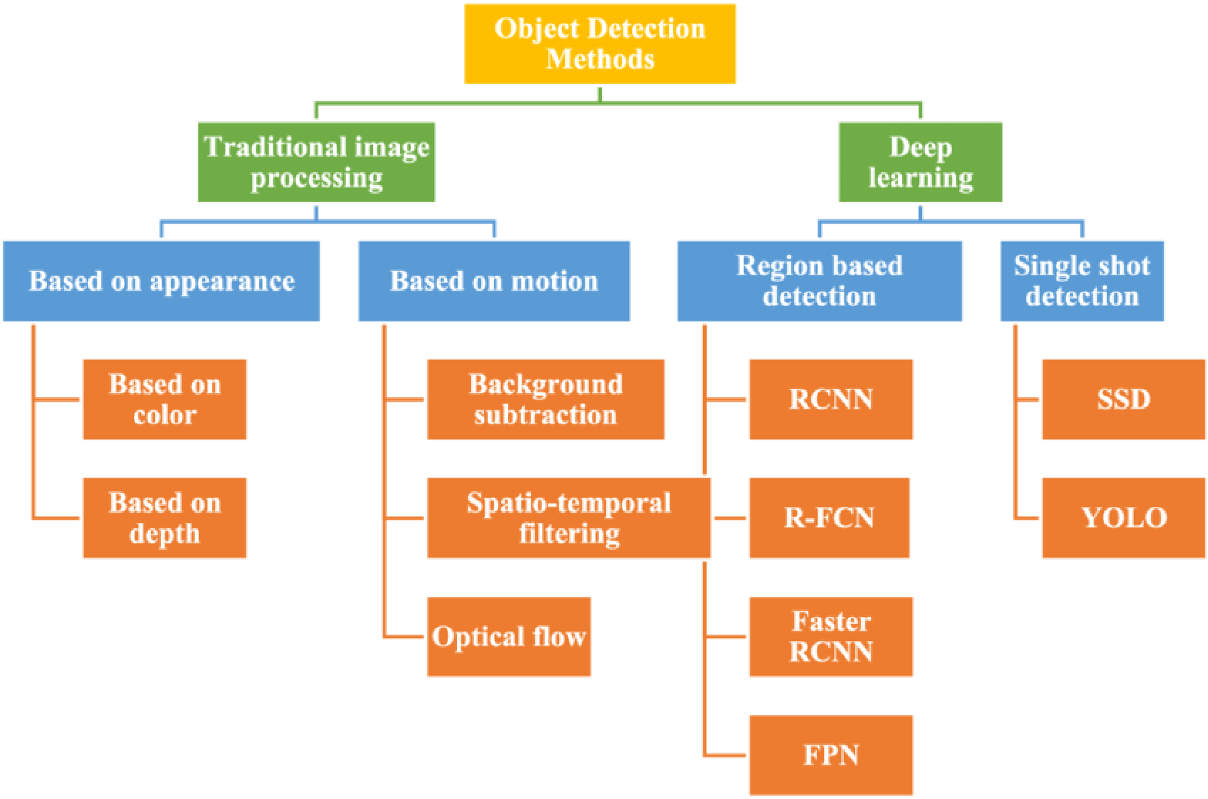


Figure I.14: Classification of object detection methods [38].

Classical Methods: These methods often rely on predefined features such as color, motion, or texture. For instance, color-based detection involves identifying an object based on its unique color signature relative to the background. Color thresholding is the core technique used in color-based detection. It involves defining a specific range of color values (threshold) that correspond to the object of interest. Pixels within this color range are considered part of the object, while those outside the range are ignored or classified as background. Motion-based detection works by identifying areas of movement between frames.

Although computationally less demanding, these methods are limited in complex environments where lighting changes or occlusion may occur.

Deep Learning-Based Methods: Deep learning has revolutionized object detection with the introduction of Convolutional Neural Networks (CNNs). A significant milestone was achieved with the R-CNN (Region-based Convolutional Neural Network) family of detectors, which use region proposals to localize objects and then apply CNNs for classification and bounding box refinement [16]. While models like Faster R-CNN [41] and Mask R-CNN have achieved state-of-the-art accuracy.However, their multi-stage process increases computational complexity, which limits their suitability for real-time performance requirements on constrained systems.

In contrast, futher advancements were made with the one-stage detectors such as YOLO (You Only Look Once) [39] and SSD (Single Shot MultiBox Detector) [29], prioritize speed over precision by bypassing the region proposal step, predicting object locations and classes simultaneously in a single pass through the network. These models provide a more practical solution for real-time applications, though they still require considerable computational resources and may sacrifice some accuracy compared to their two-stage counterparts.

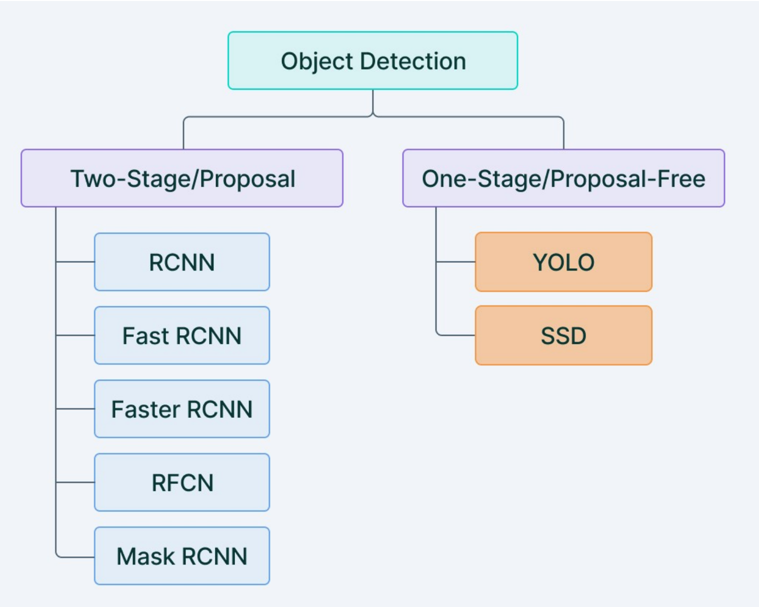


Figure I.15: Deep learning-based object detection stages.

Single-stage object detectors directly predict bounding boxes and class probabilities in a single pass through the network, simultaneously performing both tasks in one step.

Two-stage object detectors follow a region proposal-based approach, where the detection process is divided into two steps: region proposal generation and object classification and localization. In the first stage, a separate algorithm (Selective Search or a Region Proposal Network) is used to generate region proposals or regions of interest (ROIs) that may contain objects. In the second stage, these proposed regions are passed through a classifier, often a CNN, to predict the class and refine the bounding box coordinates. Two-stage detectors are generally used when accuracy is prioritized over speed.

YOLO Model

Over time, the YOLO model has evolved, with newer versions significantly improving accuracy while maintaining high speed. These improvements involve better feature extraction, more precise bounding box predictions, and a higher capacity to handle complex object detection tasks.

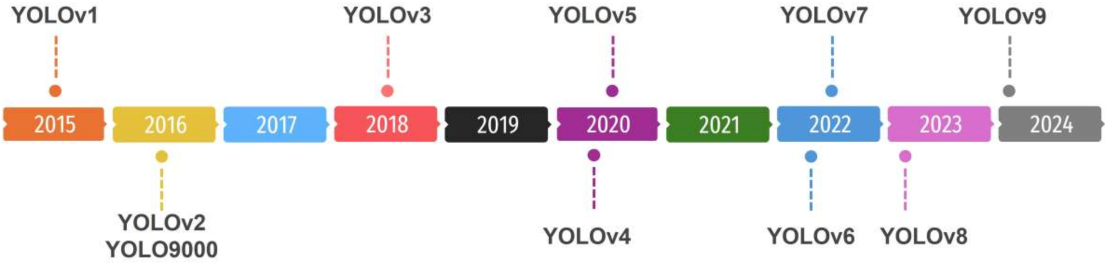


Figure I.16: YOLO versions over time.

The network divides the input image into a grid, and for each grid cell, it predicts bounding boxes, confidence scores, and class probabilities. Its biggest strength lies in its speed, as it can process images in real-time with high frame rate but sometimes struggles with small objects or overlapping objects due to the fixed-size grid, which can result in lower accuracy compared to two-stage detectors like Faster R-CNN [39].

A compact variant, known as Tiny YOLO is available and operates on the same principles of object detection, but with a focus on speed at the expense of precision [33]. Additionally, a Nano model exists which aims to balance speed and accuracy, providing improved performance over Tiny YOLO while maintaining suitability for deployment in resource-constrained environments.

SSD models

Like YOLO, SSD performs object detection in a single forward pass, but it differs in how it handles object sizes. SSD applies multiple convolutional filters at different scales to detect objects of varying sizes. This multi-scale feature extraction allows SSD to handle both small and large objects more effectively than YOLO [29].

When the model must operate under limited computational and energy constraints, MobileNet SSD was introduced as a lightweight and efficient version of the SSD model [21], specifically designed for mobile and embedded devices where computational resources are limited. The core of this model is based on MobileNet, a family of efficient CNNs developed by Google, used to extract features from input images. These features are then passed through the SSD layers, which predict bounding boxes and class probabilities. This architecture reduces the number of parameters and computational costs compared to traditional convolutional networks [21]. While its accuracy may not match that of the original SSD or YOLO, it performs well enough for practical real-time applications in resource-constrained environments.

[33] reported that the combination of SSD with MobileNet v1 provides an excellent balance between speed and accuracy, achieving a processing rate of 22 frames per second (FPS) on embedded AI hardware, specifically the NVIDIA Jetson Tegra X2.

I.5.2 Object Tracking

Object tracking is recognized as one of the most challenging tasks in computer vision, involving the continuous localization of a moving object (or multiple objects) over time. This process entails analyzing a sequence of frames, where information from previous frames is utilized to interpret the current frame. By retaining memory of past frames, the system can effectively track and follow the object's trajectory, even in the presence of occlusions or other obstructions. It is further divided into two levels: Single Object Tracking (SOT) and Multiple Object Tracking (MOT).

SOT: this is the most fundamental level of object tracking, where the objective is to track a single object of interest throughout a video sequence. The object tracking algorithm begins by defining the bounding box around the target object in the initial frame and subsequently locates the same object in the remaining frames.

MOT: developed by Zenon Pylyshyn in 1988, was originally designed to study the human visual system's ability to track multiple moving objects. MOT aims to simultaneously track multiple objects across different classes in a video, making it more challenging than single-object tracking. The algorithm identifies objects in each frame, draws bounding boxes, assigns unique coordinates, and tracks their movement across consecutive frames, even when objects occlude or appear similar, until they exit the frame.

The Difficulties Of Object Tracking

Object tracking, while a widely studied and applied technique in computer vision, still faces several difficulties and challenges. These include occlusion, illumination variations, background clutter, object deformation, scale and viewpoint changes, object interaction and occlusion, real-time performance constraints, long-term tracking, lack of labeled data, and sensor limitations [50].

Occlusion by other objects or background elements can make it difficult to track an object, while illumination variations (Changes in lighting conditions) can alter its appearance. Background clutter (Complex or dynamic backgrounds) can lead to false detections or lost tracks. Objects that change shape or appearance over time are more difficult to track compared to rigid objects. Real-time performance is crucial for applications like surveillance or autonomous systems.

Object tracking faces also other challenges such as scale and viewpoint changes that can affect tracking accuracy. As well as sensor limitations, such as low resolution, noise, or limited field of view, can impact performance.

Object Tracking Approaches

Several literature surveys have been published, examining state-of-the-art methods, their limitations, and their applications in object tracking. However, many of these methods are computationally intensive and depend on parallel processing units, making them unsuitable for the limited on-board computational resources [17].

In parallel to detection, CNNs have also been instrumental in the development of object tracking algorithms. The Siamese Network architecture, as introduced in SiamFC (Fully Convolutional Siamese Networks) [5], has been a foundational model in this domain. SiamFC leverages a fully convolutional network to compare a target image with search regions, enabling efficient and robust tracking. Subsequent models, such as SiamRPN [27] and SiamMask [53], improved the precision and robustness of tracking by integrating region proposal networks and segmentation masks, respectively.

Other approaches that have emerged include Correlation Filters, which were widely used before the rise of deep learning-based methods. These filters, such as KCF (Kernelized Correlation Filters) [18], are lightweight and computationally efficient, making them suitable for real-time tracking, though they may lack the robustness of CNN-based methods in complex scenarios.

In this study, we will focus on active object tracking with a gimbal, where the camera's orientation is continuously adjusted to keep the object of interest centered in the frame, even as the UAV changes its position. The control system will rely on the detection information to achieve this.

I.5.3 Related Works

Active object detection and tracking have been extensively studied in recent years, particularly in the context of UAV applications.

Cunha et al. in [9] propose a vision-based control system for object tracking using a 3-axis gimbal. Their system uses convolutional neural networks (CNNs) to track the target and ensures the target remains at the center of the image by controlling the gimbal's orientation. The approach is tested on a human face, demonstrating its effectiveness for autonomous cinematography and other vision-enabled robotic applications.

In , Liu et al. [30] present a real-time visual tracking system implemented on a low-cost UAV, using a Kernelized Correlation Filter (KCF)-based algorithm for object detection and tracking. The system includes a 3-axis stabilized gimbal, which autonomously aims at the target while the UAV tracks it during flight. To achieve real-time image processing, a NVIDIA Jetson TX2 module was used to implement the tracking algorithm. The experimental results show that this approach allows effective tracking of moving targets with minimal boresight error. The study provides valuable insights for using low-cost platforms in reconnaissance and target acquisition tasks.

A research by Regner et al. in [40] discusses an object-tracking control system using a gimbal mechanism for RPAS (Remotly Piloted Aircraft System) in industrial inspection environments. It leverages the YOLOv3 deep learning model for real-time target detection and a combination of non-linear inverse kinematics and PI controllers to ensure the target remains within the camera's field of view. The experiments in both simulations performed Gazebo environment and real environments validated the system's ability to maintain target tracking under various conditions using the onboard computer Jetson TX2 to process data.

Hansen and Figueiredo in [17] propose a method that leverages a three degree of freedom (DoF) gimbal system mounted on UAVs to actively track spherical and planar objects in real time. Their work emphasizes the importance of using active gimbal systems to reduce motion blur and image distortion, especially in dynamic environments where fast-moving objects must be detected and tracked reliably. Through experiments conducted in Gazebo simulation environments, they demonstrate significant improvements in pose estimation accuracy when compared to traditional passive tracking methods. Their findings suggest that active gimbal-based tracking can enhance the spatial estimation of objects in challenging conditions, making it highly relevant for autonomous UAV applications like surveillance and inspection tasks.

I.6 Conclusion

This chapter has provided a comprehensive overview of the current state of the art in image stabilization techniques, and object detection and tracking. The integration of these technologies is crucial for achieving stable, accurate imagery in airborne applications. By exploring the technological approaches, we can better understand the methods that ensure optimal image quality in dynamic environments. This knowledge sets the foundation for the subsequent development that will be discussed in the following chapters. The choice of algorithms will be guided by a careful assessment of specific material properties and available resources, with the goal of achieving real-time object detection and active tracking with high precision and an optimal frames per second (FPS) rate.

Chapter II

Theoretical Foundations of Data Acquisition, Filtering, and Control Systems

II.1 Introduction

Chapter II explores the critical concepts necessary for achieving precision in the system under study through the integration of sensor data, visual servoing, and control theory.

The chapter begins by discussing data acquisition from the MPU6050 sensor, along with the application of complementary filters to refine the sensor data. The discussion then transitions to visual servoing methods, which enable dynamic, vision-based control of the gimbal. Finally, the chapter covers PID control theory, including its components and tuning processes, which are essential for maintaining stability and achieving optimal system performance.

" Theory guides, experiment decides..."

II.2 Data Aquisition and Filtering

Data filtering is a fundamental process in signal processing and control systems, used to manipulate or enhance a signal by removing unwanted components or features to achieve desired characteristics (reducing noise, extracting useful information...).

In this project, the complementary filter is used to get a more accurate reading from the IMU sensor which is the MPU6050.

II.2.1 The MPU6050

The MPU6050, developed by InvenSense, is a six-axis IMU sensor based on MEMS[1] technology. It combines a 3-axis gyroscope and a 3-axis accelerometer integrated on a single chip that uses a standard I^{2}C [2] bus for data communication [25] [11]. Its three coordinate systems are defined as follows:

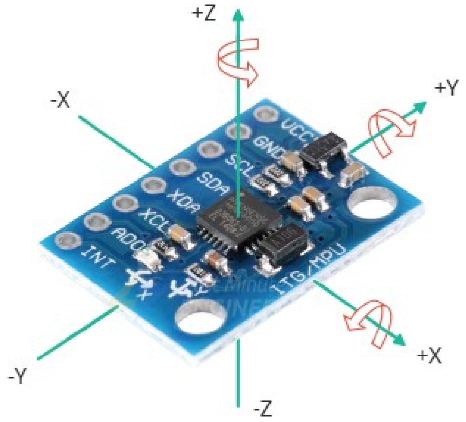


Figure II.1: The MPU6050 3-axis.

The MPU6050 also has an onboard processor, known as the Digital Motion Processor (DMP) [25] [52], which combines data from the accelerometer and gyroscope. This hardware accelerator is capable of performing complex calculations, thereby relieving the microcontroller of these tasks [11].

[1] **MEMS :** Micro-Electro-Mechanical Systems refer to a technology that encompasses miniaturized mechanical and electromechanical components fabricated using microfabrication techniques, typically ranging from a few micrometers to a few millimeters.

[2] **I2C :** stands for Inter-Integrated Circuit, is also known as IIC or I^{2}C. It is a serial communication protocol extensively used for short-distance communication between peripheral devices and microcontrollers. It consists of two signals, Serial Data Line (SDA) and Serial Clock Line (SCL) where the SCL is the clock signal and the SDA is the data signal for bidirectional, synchronous serial bus communication.

The MEMS Accelerometer

The MEMS accelerometer in the MPU6050 consists of a small proof mass suspended by springs within the silicon substrate. It is designed to measure linear acceleration forces, which may arise from static sources, such as gravity, or dynamic sources, such as motion or vibrations.

When acceleration is applied, the proof mass is displaced from its equilibrium position, leading to a change in capacitance between the mass and fixed electrodes. This variation in capacitance is then converted into an electrical signal [11], which is proportional to the acceleration experienced along the respective axis, this principle is explained in figure II.2.

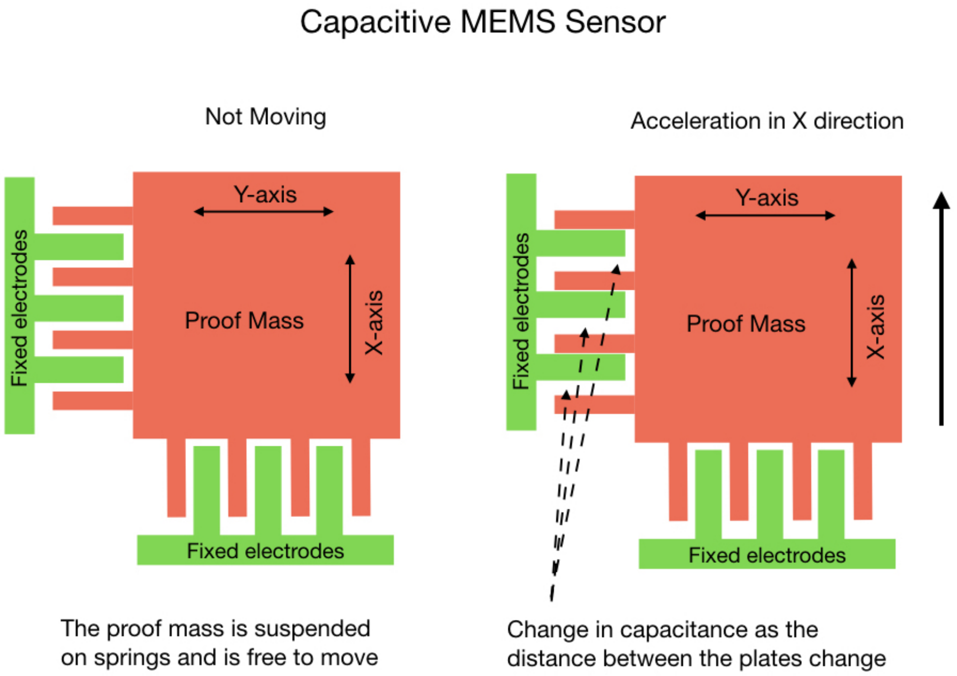

Capacitive MEMS Sensor
Not Moving
Acceleration in X direction
Y-axis
X-axis
Proof Mass
Fixed electrodes
Fixed electrodes
Y-axis
X-axis
Proof Mass
Fixed electrodes
Fixed electrodes
The proof mass is suspended on springs and is free to move
Change in capacitance as the distance between the plates change

Figure II.2: The MEMS accelerometer.

The outputs of the accelerometer are typically expressed in units of acceleration, such as meters per second squared (m/s^2) or in terms of gravitational acceleration (g), where 1g equals approximately 9.81 m/s^2. Accelerometer readings are susceptible to high-frequency noise and vibrations, and can be influenced by external forces [11]. As a result, while accelerometer data is more reliable for long-term orientation estimation, it is less accurate in capturing rapid, short-term movements.

By applying trigonometric calculations, the angle of the sensor's position relative to a reference frame can be determined in radians.

Roll Angle (ϕ)

The roll angle, which represents the rotation around the X-axis, can be calculated as follows:

$$\text{Roll} = \phi = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \tag{II.1}$$

Where:

- a_x, a_y, and a_z are the accelerations along the X, Y, and Z axes, respectively.

Pitch Angle (θ)

The pitch angle, representing the rotation around the Y-axis, can be calculated as:

$$\text{Pitch} = \theta = \arctan\left(\frac{-a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{II.2}$$

Yaw Angle (ψ)

Rotation around the Z-axis does not produce a measurable change in the gravitational force components detected by the accelerometer. This is because, during rotation around the Z-axis, the X and Y components of the gravitational force remain unchanged, as gravity continues to act in a vertically downward direction [11].

To calculate the yaw angle, gyroscopes are employed, as they measure angular velocity around all three axes (X, Y, and Z), without filtering [11]. Alternatively, a magnetometer, which is often integrated into IMUs, can be used to determine the sensor's orientation relative to the earth's magnetic field, thereby providing a reference for the yaw angle [11].

The MEMS Gyroscope

The MEMS gyroscope in the MPU6050 is composed of a small, silicon-based vibrating structure (containing 4 proof mass as shown in II.3 in a continuous oscillating movement) designed to measure angular velocity [11], defined as the rate of change of angular position over time, along the X, Y, and Z axes.

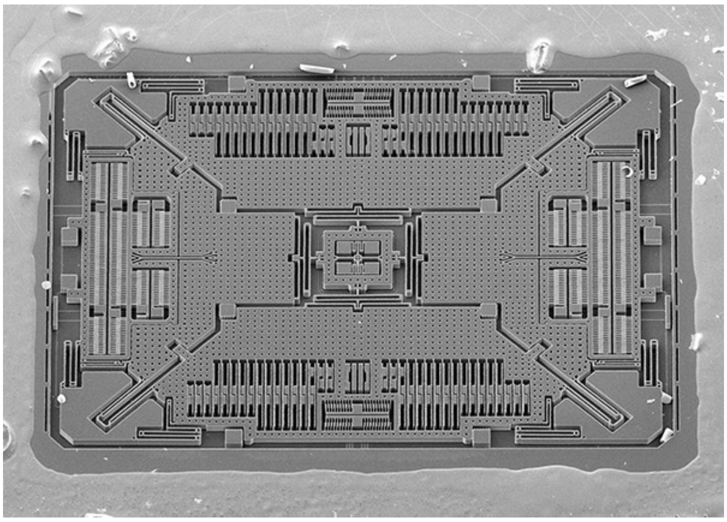


Figure II.3: The MEMS gyroscope

This gyroscope operates based on the Coriolis effect [11]. The Coriolis effect states that when a mass moves in a given direction with a certain speed and an external angular motion is applied to it, a force is generated that causes the mass to move perpendicularly [11]. The force generated is called the Coriolis force which is directly proportional to the angular velocity of the rotation : $F_c = 2m(\mathbf{v} \times \mathbf{\Omega})$ where Fc is the Coriolis force, m is the mass of the object, v is the velocity of the object relative to the rotating frame, ω is the angular velocity of the rotating frame and × denotes the cross product between the velocity and angular velocity vectors.

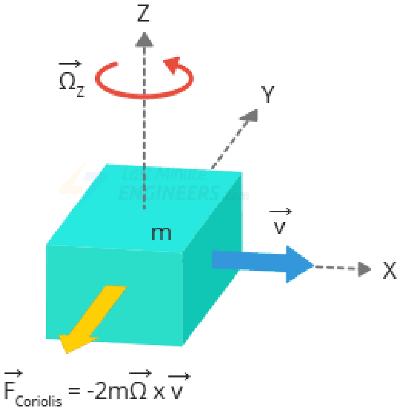


Figure II.4: The coriolis force.

Consider two masses that oscillate continuously in opposite directions. When subjected to an angular rate, the Coriolis effect acts on each mass in opposite directions. This differential effect induces a variation in the capacitance between the masses depending on the axis of the angular movement, which can be detected and measured.

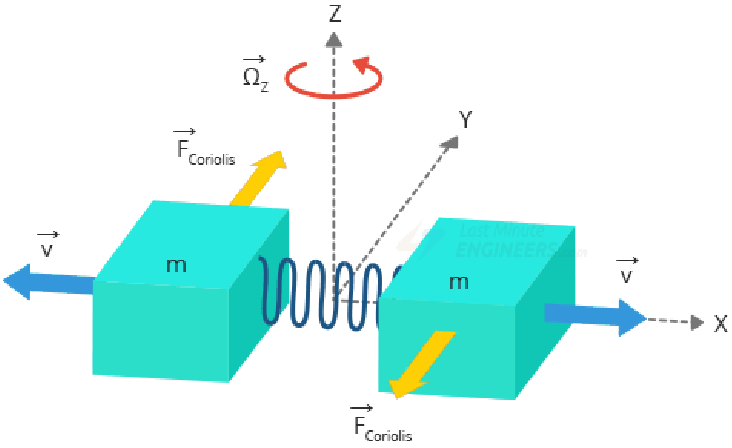


Figure II.5: Capacitance variation due to Coriolis Forces in oscillating masses.

Thus, when the device undergoes rotational motion, the Coriolis force causes the vibrating structure to shift from its original position. This displacement is detected by capacitive sensors, which convert the mechanical displacement into an electrical signal [11]. The signal is subsequently processed and digitized to represent the angular velocity in degrees per second (°/s) for each of the three axes (X, Y, and Z).

Although gyroscopes generally produce less noisy data compared to accelerometers, they are prone to drift over time due to inherent bias and noise [11]. This drift can result in accumulated errors in the measurements. To accurately determine angular position, the angular velocity provided by the gyroscope must be integrated over time.

Roll Angle (ϕ)

The roll angle is obtained by integrating the angular velocity around the X-axis :

$$\text{Roll} = \phi(t) = \phi(t_0) + \int_{t_0}^{t} \omega_x(\tau)\, d\tau \tag{II.3}$$

Where:

- ω_x is the angular velocity around the X-axis.
- $\phi(t_0)$ is the initial roll angle at time t_0.
- $\phi(t)$ is the roll angle at time t.

Pitch Angle (θ)

The pitch angle is obtained by integrating the angular velocity around the Y-axis :

$$\text{Pitch} = \theta(t) = \theta(t_0) + \int_{t_0}^{t} \omega_y(\tau)\, d\tau \tag{II.4}$$

Where:

- ω_y is the angular velocity around the Y-axis.
- $\theta(t_0)$ is the initial pitch angle at time t_0.
- $\theta(t)$ is the pitch angle at time t.

Yaw Angle (ψ)

The yaw angle is obtained by integrating the angular velocity around the Z-axis :

$$\text{Yaw} = \psi(t) = \psi(t_0) + \int_{t_0}^{t} \omega_z(\tau)\, d\tau \tag{II.5}$$

Where:

- ω_z is the angular velocity around the Z-axis.
- $\psi(t_0)$ is the initial yaw angle at time t_0.
- $\psi(t)$ is the yaw angle at time t.

To achieve highly accurate orientation information, data from both the accelerometer and gyroscope are often combined, or fused, using a complementary filter (II.2.2). This fusion process leverages the strengths of both sensors: the accelerometer provides stable long-term orientation data, while the gyroscope offers precise short-term movement detection [11].

II.2.2 Complementary Filter theory

A complementary filter is a computational technique commonly employed in signal processing, particularly in systems involving sensor fusion [48], where multiple sensor measurements are integrated to produce a more accurate and reliable estimate of a given quantity. This approach exploits the strengths of both sensors to mitigate their individual weaknesses [11].

The primary objective of a complementary filter is to combine the high-frequency response of the gyroscope with the low-frequency response of the accelerometer [11]. The underlying concept involves applying a low-pass filter to the accelerometer data and a high-pass filter to the gyroscope data, followed by the combination of the filtered signals [48] [15].

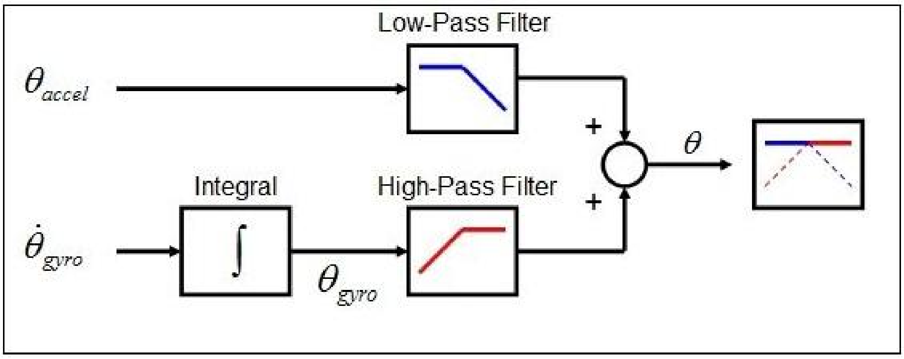


Figure II.6: The Complementary Filter [37]

This approach allows the accelerometer to correct the long-term drift of the gyroscope, while the gyroscope contributes to smooth short-term changes. The result is a stable and robust solution to generate an estimate for orientations [48].

Mathematical Formulation

Let θ be the estimated angle. The complementary filter can be mathematically expressed as [15]:

$$\theta_{\text{est}}(t) = \alpha(\theta_{\text{est}}(t-1) + \omega_{\text{gyro}} \cdot \Delta t) + (1-\alpha)\theta_{\text{acc}} \tag{II.6}$$

where:

- $\theta_{\text{est}}(t)$ is the estimated angle at time t.
- $\theta_{\text{est}}(t-1)$ is the previous estimated angle.
- ω_{gyro} is the angular velocity measured by the gyroscope.
- Δt is the time interval between measurements.

- θ_{acc} is the angle estimated from the accelerometer data.
- α is the filter coefficient, a tuning parameter that determines the balance between the accelerometer and gyroscope data. (typically $0 < \alpha < 1$ [11]).

Gyroscope Contribution : The term $\phi_{\text{est}}(t-1) + \omega_{\text{gyro}} \cdot \Delta t$ represents the predicted orientation based on the gyroscope's angular velocity. This prediction is subject to drift over time.

Accelerometer Contribution : The term $(1-\alpha)\phi_{\text{acc}}$ introduces the accelerometer data, which provides an absolute reference to correct the gyroscope's drift.

Filter Coefficient (α) : The value of α controls the blending of the gyroscope and accelerometer data. A higher α gives more weight to the gyroscope data, making the filter more responsive to short-term changes but potentially more susceptible to drift. A lower α relies more on the accelerometer, reducing drift but increasing the effect of high-frequency noise.

Advantages and Limitations

Table II.1: Advantages and Limitations of the Complementary Filter

Aspect	Advantages	Limitations
Simplicity	Computationally efficient and easy to implement.	Limited to linear systems; may not perform well in complex scenarios.
Real-time Capability	Provides real-time orientation estimation, essential for practical applications.	Tuning sensitivity; performance heavily depends on correct tuning of the filter coefficient.
Stability	Balances short-term accuracy of the gyroscope with long-term stability of the accelerometer.	Not suitable for all scenarios, especially in environments with significant external accelerations.

II.3 Visual servoing

Visual servoing, or vision-based robotic control, is a fundamental technique in robotics that uses visual data to control motion. It integrates methods from image processing, computer vision, and control theory [24][6].

By using computer vision data as feedback, the robot can adjust its position and orientation in real-time, reducing the difference between the desired and observed visual features to achieve accurate positioning. Visual servoing is classified into two main types based on the features utilized in the feedback loop [6], as follows:

II.3.1 Position-Based Visual Servoing (PBVS)

In PBVS, visual data is processed to determine the 3D pose position and orientation of the robot or target object within the workspace. This pose information, obtained through image processing and camera calibration, is then used by the control system to guide the robot to the desired position [6][24].

II.3.2 Image-Based Visual Servoing (IBVS)

In IBVS, the control system directly utilizes visual features extracted from the image plane, such as pixel coordinates of points, lines, or contours, to control the robot. Unlike PBVS, IBVS operates within the 2D image space, using image measurements without reconstructing the 3D pose [6][24]. A notable study by Hurák et al. introduced an image-based pointing controller, developed using the IBVS approach, where the control inputs were the camera's angular rates [22][23].

Table II.2: Advantages and Limitations of PBVS and IBVS

Aspect	**Point-Based Visual Servoing (PBVS)**	**Image-Based Visual Servoing (IBVS)**
Advantages	- Provides precise control of the robot's position in 3D space. - Allows for better handling of occlusions and depth variations. - Useful for tasks requiring precise spatial alignment.	- Simpler and more intuitive as it deals directly with image data. - Can be more robust to changes in the environment or occlusions. - Easier to implement for certain visual tasks.
Limitations	- Requires accurate 3D information and calibration. - Computationally intensive and may need complex algorithms for feature extraction. - Sensitive to errors in depth perception.	- Can be less accurate in terms of 3D positioning. - Performance may degrade if there are significant changes in the scene or if features become indistinguishable. - May suffer from scale ambiguity.

Extracting visual data for control feedback

Figure II.7 illustrates the pan and tilt errors that are crucial to the visual servoing system. Pan refers to the horizontal rotation of the camera, while tilt refers to the vertical rotation. The figure provides a visual explanation of how the system calculates and corrects these pan and tilt errors to maintain the object of interest at the center of the camera frame.

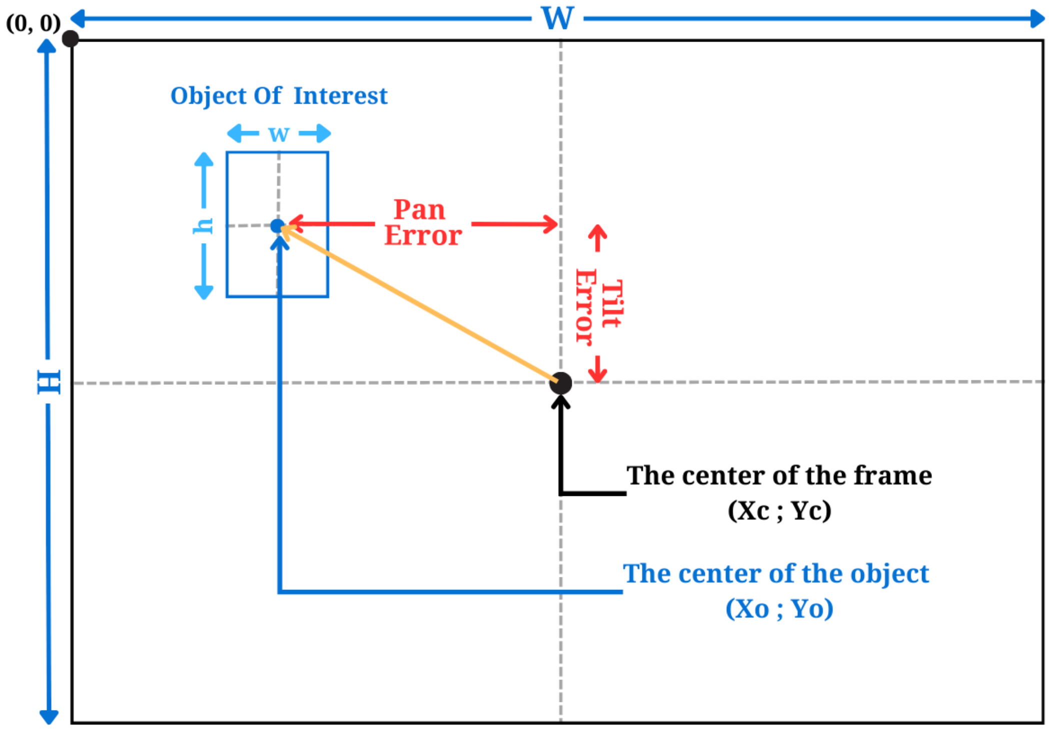

(0, 0)
W
H
Object Of Interest
w
h
Pan Error
Tilt Error
The center of the frame (Xc ; Yc)
The center of the object (Xo ; Yo)

Figure II.7: Representation of Pan and Tilt Errors in Object Localization.

The object of interest is represented as a rectangle located within the 2D coordinate system of the camera's field of view. The object's dimensions are w (width) and h (height). The center of the object is (Xo, Yo).

The camera frame is characterized by its width W and height H. The center of the frame, located at (Xc, Yc), is the target alignment point for the object. The pan error is the horizontal deviation between the object's center and the frame's center, while the tilt error is the vertical deviation. Mathematically, the pan and tilt errors can be expressed as:

$$\text{Pan Error} = Xo - Xc$$

$$\text{Tilt Error} = Yo - Yc$$

These errors are critical parameters for the servoing control system, which adjusts the camera's orientation to minimize them, thereby ensuring the object remains centered in the frame.

This representation is pivotal in understanding how visual feedback from the camera is processed and used to control the gimbal's servomotors, ensuring consistent object tracking during airborne operations.

In visual servoing, several control strategies are commonly used, each tailored to different aspects of the system's dynamics and task requirements. The most prevalent strategies include Proportional-Derivative (PD) control, Proportional-Integral-Derivative (PID) control, and other advanced approaches.

II.4 Control theory

Process control systems are critical for the design of safe and efficient industrial plants. Among the various control methodologies available, the Proportional-Integral-Derivative (PID) controller stands out as one of the simplest yet most effective solutions [4] .

In the context of our study, achieving stabilization and precise control of the gimbal is paramount for maintaining steady and accurate orientation, as well as for ensuring effective object tracking. Therefore, we will implement a PID controller, widely recognized as a robust and efficient control algorithm capable of delivering optimal system performance.

II.4.1 PID controller overview

A Proportional-Integral-Derivative (PID) controller is a control loop feedback mechanism widely used in industrial control systems [4]. The PID controller continuously calculates an error value as the difference between a desired setpoint and a measured process variable and then performing a corrective action to adjust the process accordingly based on three fundamental parameters : Proportional term(P) which depends on the present error, Integral term(I) which depends on past errors, and Derivative term(D) which depends on anticipated future errors. These parameters can be weighted, or tuned, to adjust their effect on the process.

Consider the exemplary scheme depicted in Figure II.8. Initially, the system output $y(t)$ is fed back and compared with the setpoint $r(t)$ by the comparator, producing the time-dependent error signal $e(t) = r(t) - y(t)$. This error signal is subsequently minimized by the loop filter and used to generate the control signal $u(t)$, which drives the system's output, thus initiating closed-loop operation.

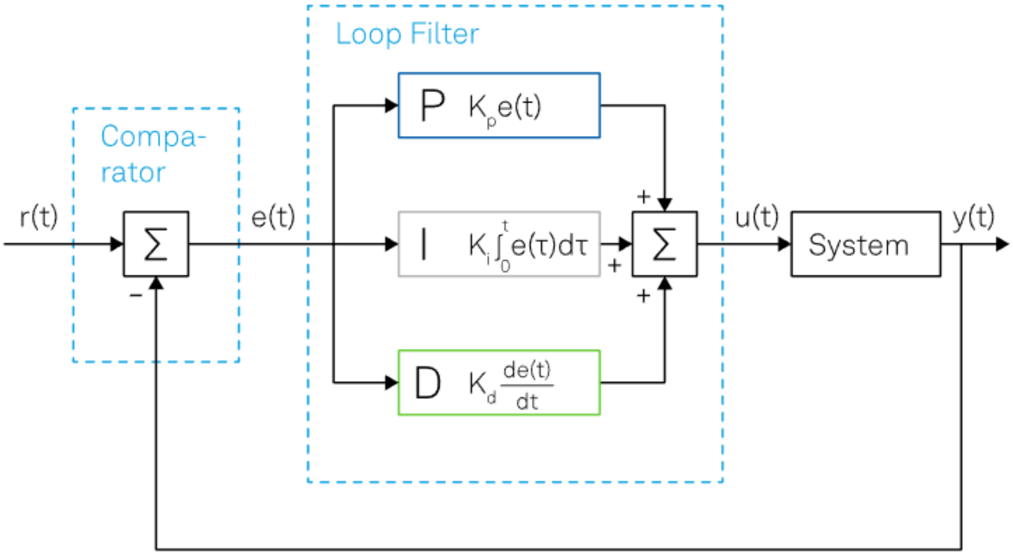


Figure II.8: Block Diagram of a PID Controller System.

II.4.2 Components of a PID controller

Proportional Term (P)

The proportional term produces an output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant K_p, known as the proportional gain. If the error is large, the proportional response is large, which helps to reduce the error quickly. However, relying solely on the proportional term can lead to steady-state error. The proportional control action is given by:

$$P(t) = K_p e(t) \tag{II.7}$$

Integral Term (I)

The integral term (also known as Reset Control) sums the error over time, ensuring that even small errors accumulate and are corrected. This term helps eliminate steady-state errors, ensuring the system eventually reaches the setpoint. The integral response is adjusted by the integral gain K_i. The integral control action is given by:

$$I(t) = K_i \int_0^t e(\tau)\, d\tau \tag{II.8}$$

Derivative Term (D)

The derivative term predicts future error by calculating the rate of change of the error. It adds a damping effect, reducing the tendency for the system to oscillate and improving the stability of the response. The derivative response can be adjusted by the derivative gain K_d. The derivative control action is given by:

$$D(t) = K_d \frac{de(t)}{dt} \tag{II.9}$$

PID Control Equation

The overall control action is the sum of the proportional, integral, and derivative terms:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{II.10}$$

where:

- $u(t)$ is the control output.
- $e(t)$ is the error at time t.
- K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

II.4.3 Tuning a PID Controller

Tuning a PID controller involves setting the optimal gains K_p, K_i, and K_d to achieve the desired performance. However, prior to tuning the PID gains, it is essential to understand their effects on system performance, as shown in Table II.3:

Table II.3: Effects of Increasing Controller Parameters on System Performance.

Controller Parameter	**Effect on Rise Time**	**Effect on Overshoot**	**Effect on Settling Time**	**Effect on Steady-State Error**	**Effect on Stability**
K_p	Decrease	Increase	Minor change	Decrease	Degrade
K_i	Decrease	Increase	Increase	Eliminate	Degrade
K_d	Minor change	Decrease	Decrease	No Effect	Improve

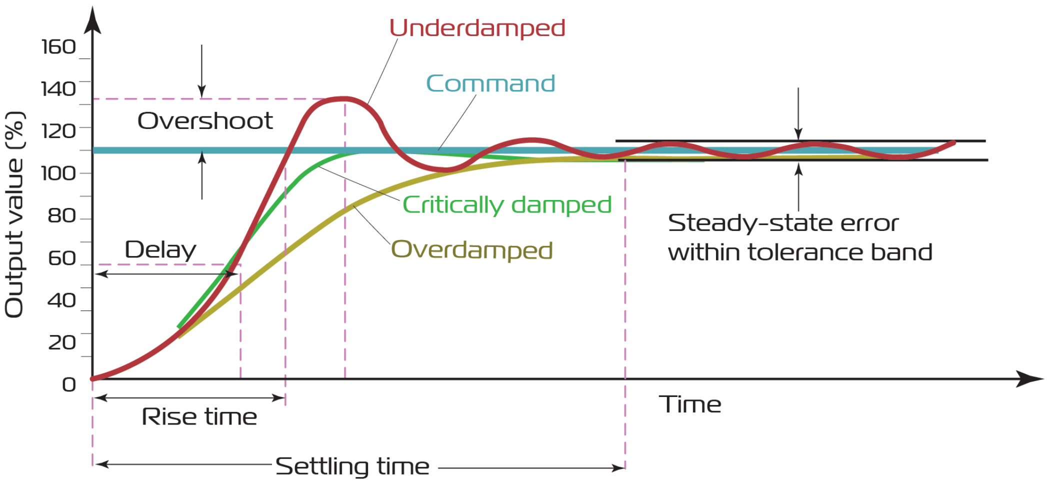


Figure II.9: Transient response parameters.

The proper tuning of a servo system can be assessed based on three key characteristics: response time, settling time, and overshoot. Response time is the duration to reach a specified percentage of the target value, settling time is the duration to stabilize within a certain range of the target value, and overshoot is the extent of exceeding the target value. The aim of servo tuning is to minimize response time, settling time, and overshoot to achieve optimal system performance. Several methods exist for tuning, including:

Manual Tuning

- Ziegler-Nichols Method [4][36] The foundational tuning rules were established by Ziegler and Nichols in the 1940s. Their method entailed executing a basic experiment on the process to derive dynamic features in both the time and frequency domains. For more detailed information, please refer to [4].
- Trial and Error In practice, many PID controllers are tuned using trial and error. Engineers adjust the parameters incrementally while observing the system's response until the desired performance is achieved.
  - Start with K_p at a low value and gradually increase it until the system responds adequately without excessive overshoot.
  - Adjust K_i to eliminate steady-state error while monitoring for any introduction of oscillations.
  - Fine-tune K_d to reduce overshoot and damping.

Automated Tuning

- Software Tools : Many modern control systems use software tools that can automatically tune PID controllers based on system models or response data.
- Optimization Algorithms: Techniques such as Genetic Algorithms (GA) [36], Particle Swarm Optimization (PSO)[36] [42] or gradient descent can be employed to find optimal PID parameters.

Table II.4: Summary of Optimization Methods for PID Tuning

Method	Description	Advantages	Disadvantages
Genetic Algorithms (GA)	Evolutionary algorithm that mimics natural selection by iterating over a population of candidate PID parameters and evaluates each set of PID gains based on performance metrics.	– Can handle nonlinear, complex systems. – Avoids local minima. – Flexible and robust.	– Computationally intensive. – Requires many function evaluations. – May converge slowly.
Particle Swarm Optimization (PSO)	Nature-inspired optimization algorithm based on the behavior of bird flocking or fish schooling. Each ”particle” represents a possible PID parameter set and moves through the search space by adjusting its position based on its own best experience and the best experience of the swarm.	– Simple to implement. – Few parameters to tune. – Faster convergence than GA.	– May converge prematurely to a suboptimal solution. – Sensitive to initial conditions.
Gradient Descent	First-order optimization algorithm that iteratively adjusts PID parameters in the direction of the steepest decrease of a cost function, typically derived from system error. The method can be effective in tuning PID parameters by minimizing a performance index such as Root Mean Square Error (RMSE)	- Direct minimization of the error. - Effective for convex problems. - Fast convergence with well-behaved systems.	– Sensitive to learning rate. – Struggles with non-convex problems. – May get stuck in local minima.

The PID controller's robustness and simplicity make it a preferred choice for many control applications, including our system. By carefully tuning the PID gains, the system achieves optimal performance, balancing response speed, stability, and minimal steady-state error.

II.5 Conclusion

This chapter has provided a concise yet comprehensive overview of the integration of sensor data, visual servoing techniques, and PID control theory.

By leveraging the MPU6050 sensor and complementary filters, systems can achieve the accurate data processing essential for precise motion control. Visual servoing techniques enable responsive, vision-based adjustments, enhancing the interaction between the system and its environment. The exploration of PID controllers underscores their importance in fine-tuning system responses to ensure stability and accuracy. This groundwork prepares for further analysis and optimization, with the next chapter focusing on implementing these approaches.

Chapter III

System Design and Implementation

III.1 Introduction

The implementation phase is a crucial stage in the development process, where theoretical concepts, discussed in chapter II, and designs are transformed into a functional prototype.

This chapter outlines the detailed steps taken to realize the proposed system. Following this, a detailed exploration of the system architecture is presented, covering the mechanical design, the selection of appropriate hardware components, and the software and development tools utilized.

The core of the implementation involves processing and calibrating the Inertial Measurement Unit (IMU) data, applying a complementary filter for IMU data fusion, and integrating the processed data into the control algorithm to achieve effective stabilization.

Furthermore, the chapter delves into detection and tracking functionalities, including the integration of visual data into the control algorithm to center the object of interest.

Each section of this chapter is written to provide a comprehensive understanding of the technical challenges encountered and the solutions employed to ensure the system's performance and reliability.

" Theory without practice is empty;
practice without theory is blind "
Immanuel Kant

III.2 Methodology

The methodology adopted serves as a blueprint for transforming theoretical concepts into a functional prototype, ensuring that all components work harmoniously to meet stabilization and tracking objectives.

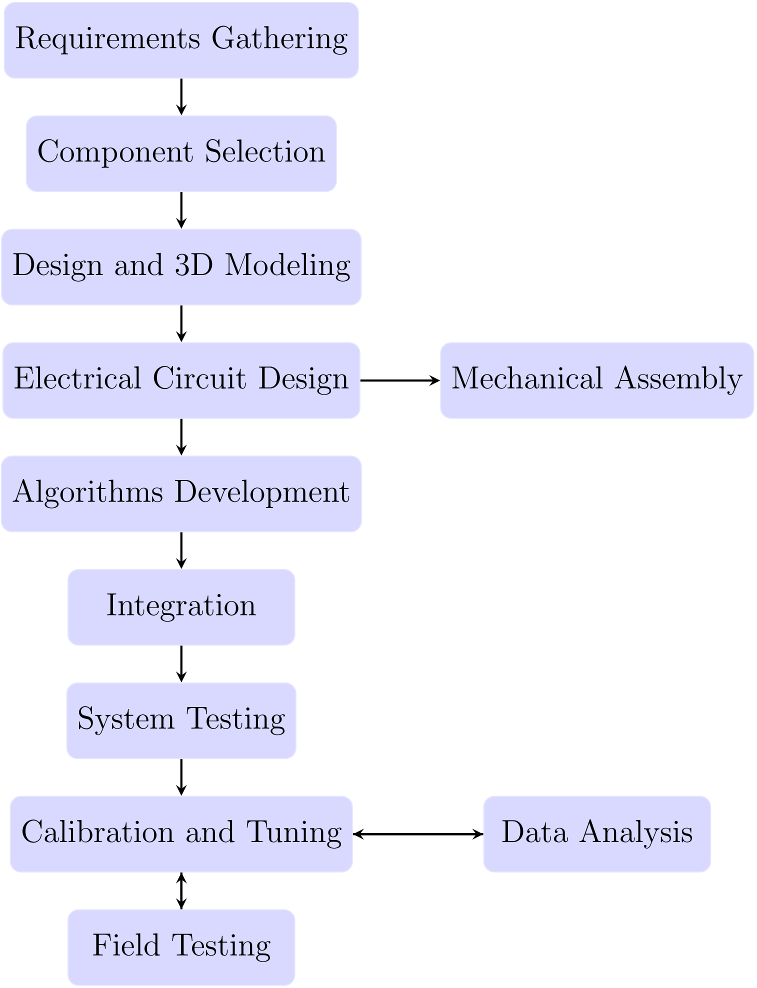


Figure III.1: Design methodology flowchart

The methodology begins with the collection and documentation of system requirements, which guide the selection of hardware and software components. 3D design models are developed to ensure component fit and functionality, followed by the design of electrical circuits and mechanical assembly. Algorithms are developed to control the hardware, ensuring integration between hardware and software. The system undergoes rigorous testing, calibration, and tuning to meet performance standards. Finally, real-world testing validates the system's effectiveness in practical applications.

The successful completion of this project necessitates the manufacturing and procurement of various components. Given the time constraints, the procurement process should be streamlined, focusing on simplicity, short delivery times, and minimal costs.

III.3 System Architecture Overview

The system architecture integrates both hardware and software components to achieve precise control and stabilization of the camera platform. This architecture leverages multiple sensors, actuators, and processing units to maintain camera stability and focus on the target object.

III.3.1 Mechanical Design

The mechanical design of the system is essential for ensuring stability, precision, and reliability. The foundation of this mechanical design is the 3D-printed structure that forms the 3-axis gimbal. This structure is based on an existing design from the site below [1], with several modifications and optimized dimensions to better accomodate the components and meet the specific requirements of the system.

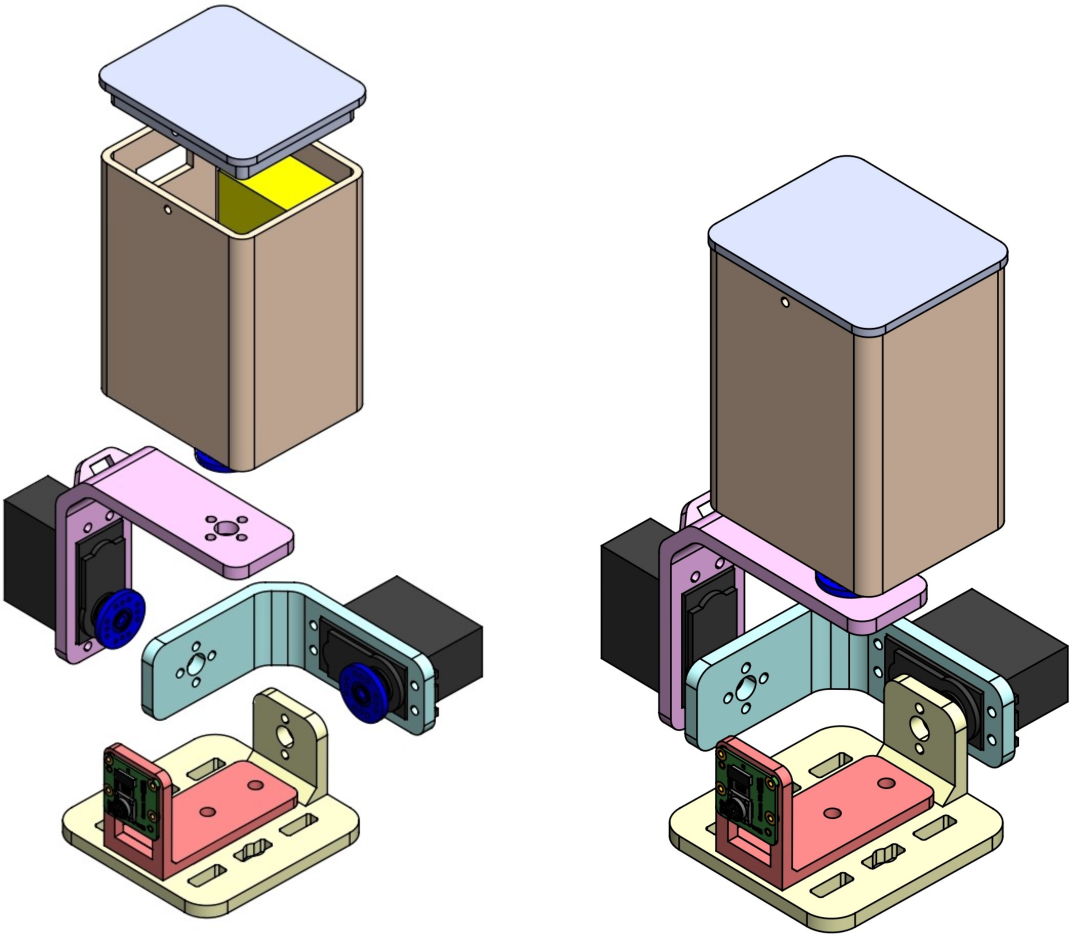


Figure III.2: Gimbal exploded and isometric views.

[1] https://shorturl.at/YOsrd

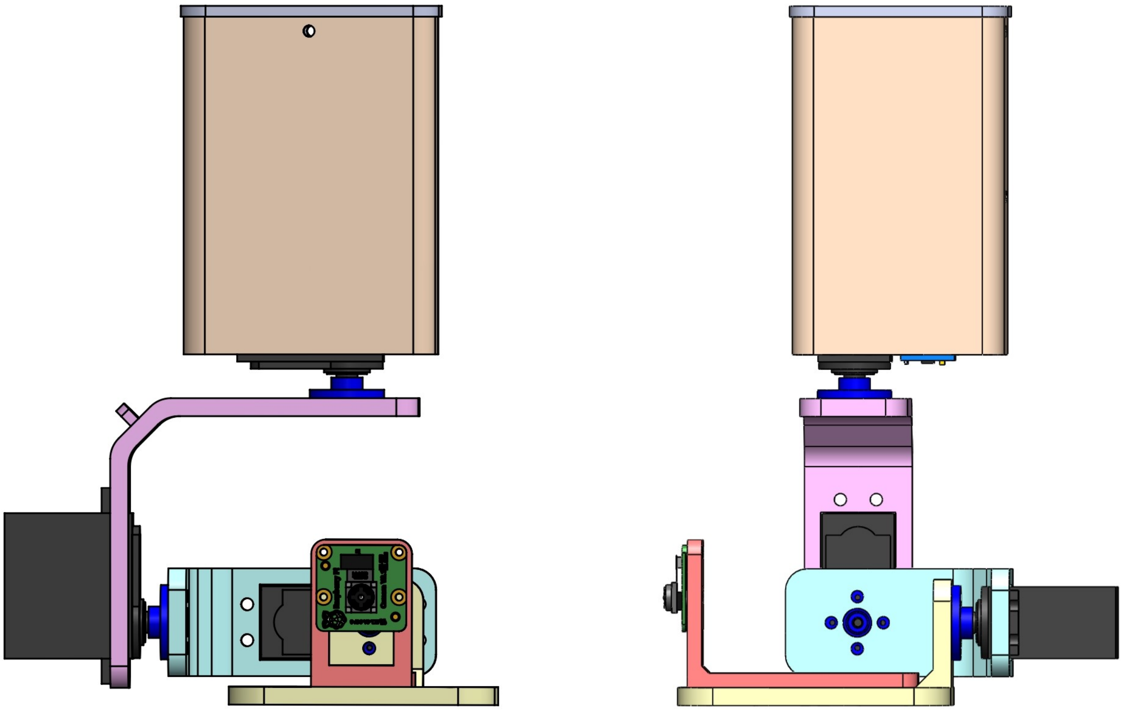


Figure III.3: Gimbal side and front views.

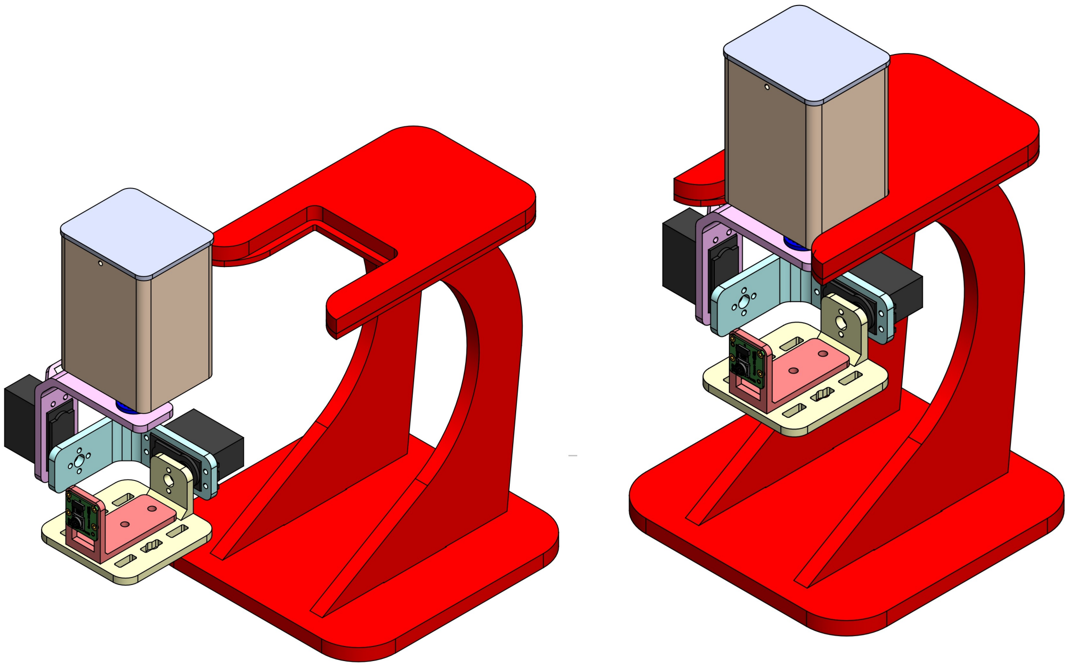
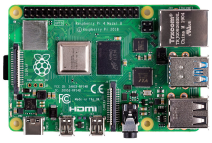


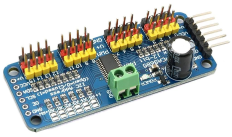
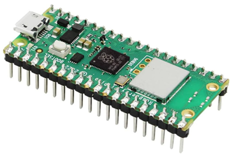
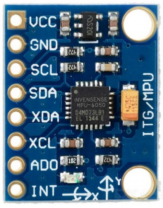
Figure III.4: Gimbal holder.

III.3.2 Hardware Components Selection

To achieve the desired functionality, various components were carefully selected. The selection criteria included compatibility, performance, ease of integration, and cost-effectiveness. The primary electronic components utilized in the system are enumerated in the table III.1:

Table III.1: Gimbal Electronics

Component	Description	Specifications
 Figure III.5: Raspberry Pi 4 model B.	The Raspberry Pi 4 is a powerful single-board computer capable of handling a variety of tasks.	- Quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz - 4GB LPDDR4-3200 SDRAM - Dual-band 802.11ac wireless, Bluetooth 5.0, BLE - 2 × USB 3.0 ports, 2 × USB 2.0 ports - Gigabit Ethernet From the datasheet [13].
Figure III.6: Raspberry Pi Camera V2.	The Raspberry Pi Camera V2 is a high-quality camera module designed for use with Raspberry Pi boards.	- 8 megapixel Sony IMX219 sensor - Fixed focus lens - 1080p30, 720p60 and 640x480p90 video From the datasheet [14].
Figure III.7: MG996R Servomotor.	The MG996R servomotor is a robust and high-torque motor widely used in robotics and control systems.	- Operating Voltage: 4.8V - 7.2V - Stall Torque: 9.4 kg/cm (4.8V), 11 kg/cm (6V) - Speed: 0.19 sec/60° (4.8V), 0.14 sec/60° (6V) - 180° rotation From the datasheet [34].

Component	Description	Specifications
 Figure III.8: PCA9685.	The PCA9685 is a PWM driver board used for controlling multiple servomotors (3 servos are required).	• 16-channel, 12-bit PWM servo driver • I2C-controlled • Adjustable frequency PWM up to about 1.6 KHz • 3.3V or 5V compliant From the datasheet [46].
 Figure III.9: Raspberry Pi Pico W.	The Raspberry Pi Pico W is a microcontroller board based on the Raspberry Pi RP2040 chip.	• Dual-core ARM Cortex M0+ processor • 264KB SRAM, 2MB onboard Flash memory • 2.4GHz 802.11n wireless LAN • 26 multi-function GPIO pins From the datasheet [12].
 Figure III.10: MPU6050.	The MPU6050 is a 6-axis motion tracking device that integrates a 3-axis gyroscope and a 3-axis accelerometer.	• 3-axis gyroscope and 3-axis accelerometer • Digital Motion Processor (DMP) for sensor fusion • Communication: I2C • Operating Voltage: 3.3V - 5V From the datasheet [25].

Component Communication and Integration within the System Architecture

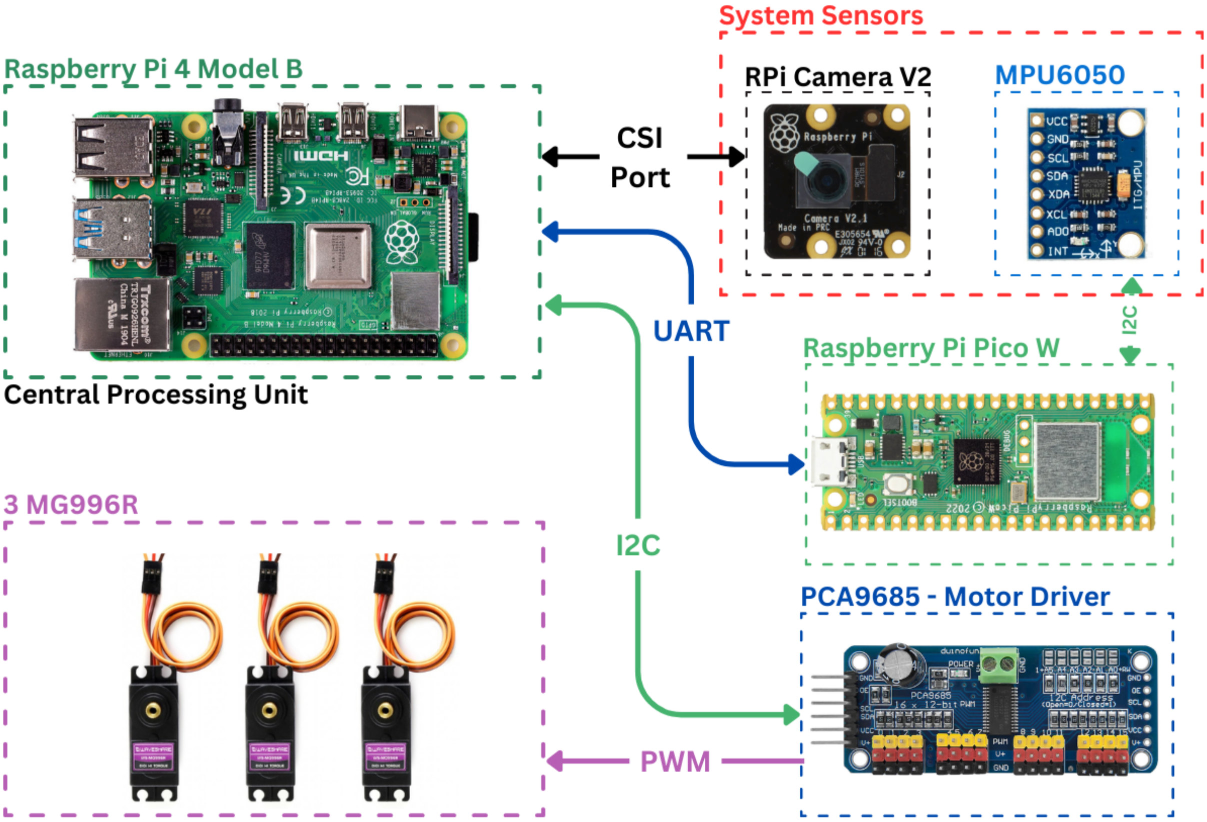


Figure III.11: System Architecture.

The central processing unit of the system is **the Raspberry Pi 4**, which is responsible for executing control algorithms, processing visual data, and managing communication with peripheral devices. This device functions as the primary computational hub, coordinating inputs from sensors and outputs to actuators.

The Raspberry Pi Camera v2 is directly connected to the Raspberry Pi 4 via the Camera Serial Interface (CSI) port. It captures live video feeds, which are integral to both object detection and tracking processes.

The PCA9685 is an I2C [2]-based PWM [3] driver utilized for controlling the MG996R servomotors [46]. The PCA9685 interfaces with the Raspberry Pi 4 to receive control signals and accurately modulate the positions of the servomotors.

[2] **I2C :** stands for Inter-Integrated Circuit, is also known as IIC or I^{2}C. It is a serial communication protocol extensively used for short-distance communication between peripheral devices and microcontrollers. It consists of two signals, Serial Data Line (SDA) and Serial Clock Line (SCL) where the SCL is the clock signal and the SDA the data signal for bidirectional, synchronous serial bus communication.

[3] **PWM :** stands for Pulse Width Modulation is a technique used to control servos by varying the width of the pulse in a periodic signal.

MG996R Servomotors are high-torque servos tasked with adjusting the camera's orientation. These servomotors allow the system to counteract movements and vibrations, thereby stabilizing the camera and maintaining a steady focus on the tracked object.

The Raspberry Pi Pico W is a microcontroller that assists in processing additional sensor data, specifically from the MPU6050. It communicates with the Raspberry Pi 4 via UART [12] [4], facilitating efficient data collection and processing.

The MPU6050 (Gyroscope and Accelerometer) provides real-time data on the system's orientation and acceleration. This sensor is essential for detecting and compensating for any disturbances that may affect the camera's stability.

[4] **UART :** stands for Universal Asynchronous Receiver/Transmitter. It is a hardware communication protocol used for asynchronous serial communication between devices. It utilizes two lines, TX (Transmit Line) and RX (Receive Line).

III.3.3 Software and Development Tools

To effectively manage and develop the system, several software and development tools were utilized. In this project, a Raspberry Pi OS Legacy 64-bit, a port of Debian Bullseye, was used.

We established a connection to the Raspberry Pi using the SSH protocol [5]. We used PuTTY, a user-friendly interface that allows remote connections to UNIX, Linux, and Raspberry Pi systems from a Windows computer. To achieve this, we obtained the IP address of our Raspberry Pi on our network. Since the Raspberry Pi does not provide direct output, the project employs the VNC Viewer client to display the output. The VNC Viewer connects to the RPi via its IP address, enabling us to interface with it from our desktop computer [43].

Table III.2 provides a comprehensive overview of the software and development tools used in this project.

Table III.2: Software and Development Tools Used.

Software/Tool	**Description**
Raspberry Pi OS (Legacy-64 bits)	Operating system based on Debian Bullseye, optimized for the Raspberry Pi hardware.
PuTTY	Free and open-source terminal emulator, used for SSH, Telnet, and serial console connections, commonly used for accessing the Raspberry Pi remotely via SSH.
VNC Viewer	VNC stands for Virtual Network Computing. It is a remote desktop access tool, used for managing the Raspberry Pi remotely.
Python	High-level programming language used for developing control algorithms and image processing routines.
MicroPython	Lean and efficient implementation of Python 3 for microcontrollers, used on the Raspberry Pi Pico W.
Thonny	Integrated development environment (IDE) for Python, used for writing and debugging Python and MicroPython code.

[5] **SSH or Secure Shell:** a network protocol that enables a secure and encrypted connection between two computers

III.4 Prototype

Before assembly, each component of the system was individually tested using a breadboard and basic connections to ensure proper functionality. A prototype has been then constructed as shown in Figure III.12.

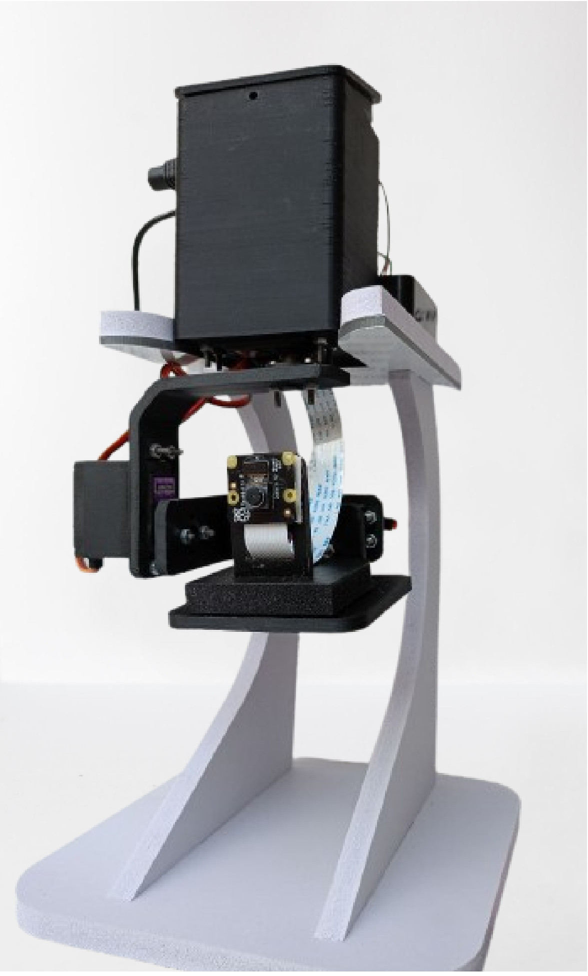


Figure III.12: The prototype Aero Vision 0.1

PLA (Polylactic Acid) was chosen for its ease of printing, strength, and lightweight properties. The structure is printed with a layer height of 0.2 mm and an infill density of 60% to balance strength and weight.

Precise alignment and calibration of the servomotors are crucial for accurate control, and the mounting points are designed to ensure correct positioning.
Pan Axis (Yaw): The top section of the gimbal supports the pan servomotor, enabling horizontal rotation.
Tilt Axis (Pitch): The middle section of the gimbal houses the tilt servomotor, allowing for vertical movement.
Roll Axis: The base of the gimbal contains the roll servomotor, ensuring stable camera orientation.

III.5 The Camera Stabilization

III.5.1 IMU data Processing and Calibration

Before integrating the IMU into a project, it must undergo calibration. Accurate calibration is essential for ensuring precise measurements and reliable performance in gyroscopic and accelerometric applications. This section details the data readings obtained from the MPU6050 sensor and describes the calibration procedures used, with a focus on aligning the sensor's output with known reference values.

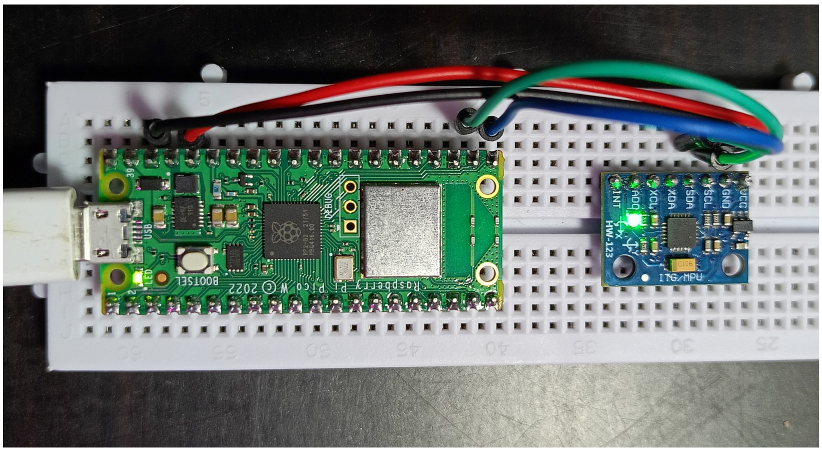
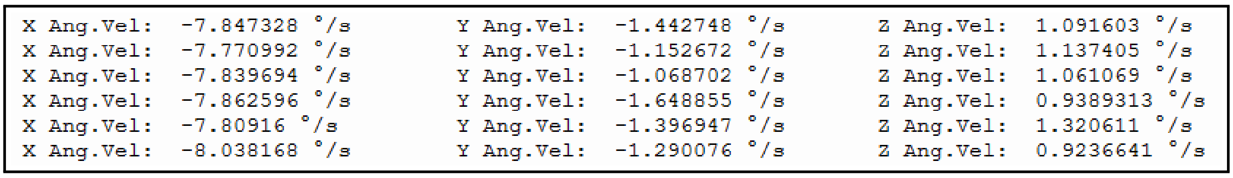


Figure III.13: MPU6050 Sensor with Raspberry Pi Pico W.

The sensor is powered by the Raspberry Pi Pico W, and data is read digitally via the I2C serial communication protocol. The MPU6050 library for MicroPython, from the site below [6], was used to convert the information from bits to raw sensor data. This data can then be utilized to calculate the pitch, roll, and yaw angles.

Gyroscope Readings Before Calibration

The readings obtained from the gyroscope sensor before the calibration process are presented in Figures III.14, III.15. When the IMU is in a steady state, it should ideally measure approximately 0°/s on the x, y, and z-axes.



```
X Ang.Vel:  -7.847328 °/s        Y Ang.Vel:  -1.442748 °/s        Z Ang.Vel:  1.091603 °/s
X Ang.Vel:  -7.770992 °/s        Y Ang.Vel:  -1.152672 °/s        Z Ang.Vel:  1.137405 °/s
X Ang.Vel:  -7.839694 °/s        Y Ang.Vel:  -1.068702 °/s        Z Ang.Vel:  1.061069 °/s
X Ang.Vel:  -7.862596 °/s        Y Ang.Vel:  -1.648855 °/s        Z Ang.Vel:  0.9389313 °/s
X Ang.Vel:  -7.80916 °/s         Y Ang.Vel:  -1.396947 °/s        Z Ang.Vel:  1.320611 °/s
X Ang.Vel:  -8.038168 °/s        Y Ang.Vel:  -1.290076 °/s        Z Ang.Vel:  0.9236641 °/s
```

Figure III.14: Gyroscope raw data readings from MPU6050 before calibration.

[6] https://shorturl.at/sIyV2

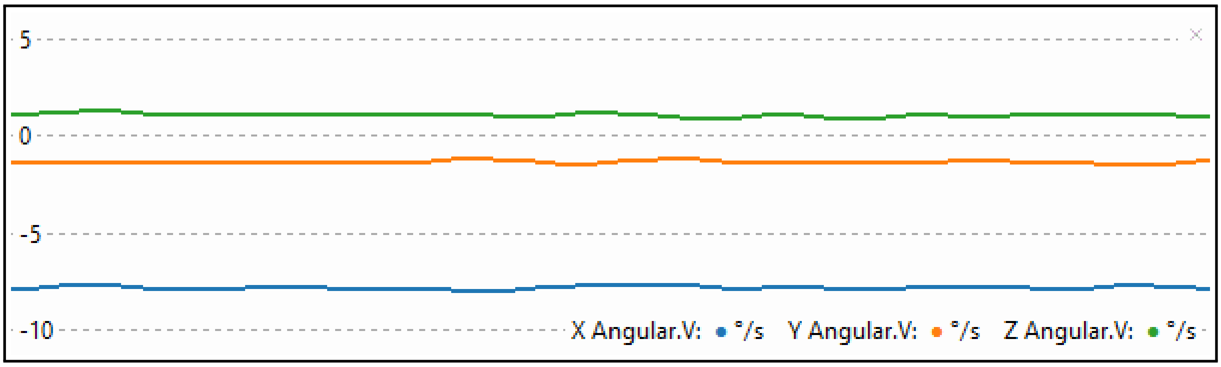
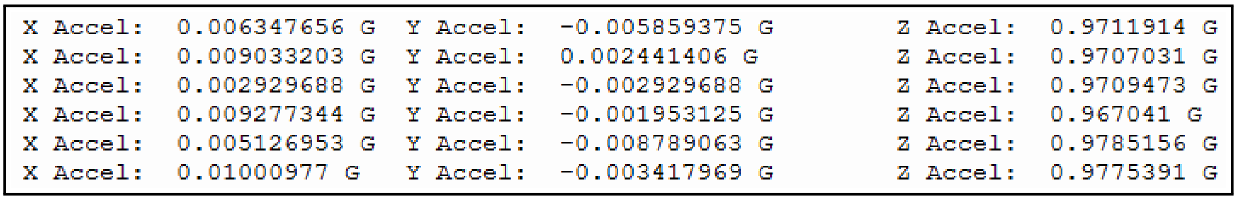


Figure III.15: Visualization of gyroscope signals before calibration.

Accelerometer readings before calibration

The readings obtained from the accelerometer sensor before calibration process are presented in figures III.16, III.17, and III.18. When the IMU is in a steady state, it should ideally measure 9.8 m/s^2 on the z-axis and approximately 0 m/s^2 on the x and y-axes.



```
X Accel:  0.006347656 G  Y Accel:  -0.005859375 G      Z Accel:  0.9711914 G
X Accel:  0.009033203 G  Y Accel:  0.002441406 G       Z Accel:  0.9707031 G
X Accel:  0.002929688 G  Y Accel:  -0.002929688 G      Z Accel:  0.9709473 G
X Accel:  0.009277344 G  Y Accel:  -0.001953125 G      Z Accel:  0.967041 G
X Accel:  0.005126953 G  Y Accel:  -0.008789063 G      Z Accel:  0.9785156 G
X Accel:  0.01000977 G   Y Accel:  -0.003417969 G      Z Accel:  0.9775391 G
```

Figure III.16: Accelerometer raw data readings from MPU6050 before calibration.

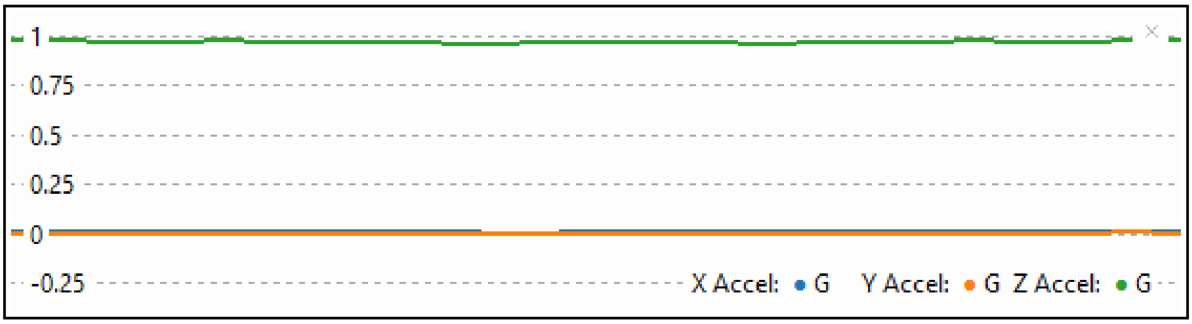


Figure III.17: Visualization of the accelerometer signals before calibration.

Figures III.16 and III.17 demonstrate that accelerometer calibration was not required, as the raw data outputs and signals confirm that the accelerometer is correctly aligned.

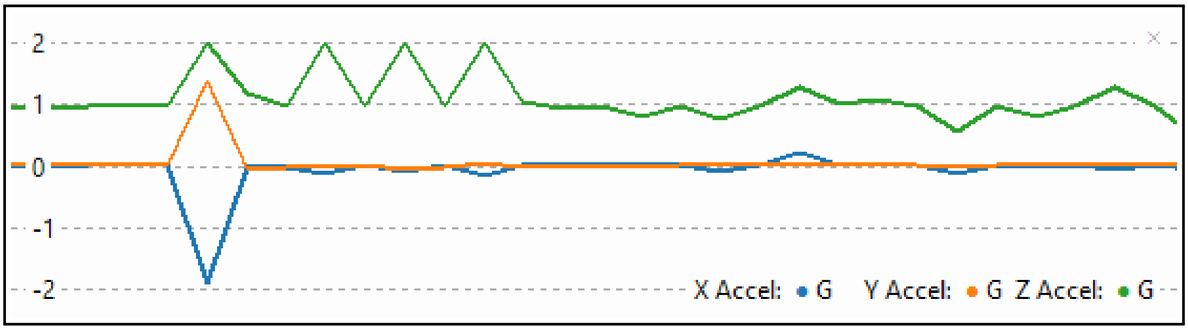


Figure III.18: Impact of vibrations on accelerometer readings.

Following the application of vibration to the accelerometer, Figure III.18 illustrates the sensor's sensitivity to these vibrations, as discussed in Chapter II (II.2.1).

Given that the IMU may not be perfectly aligned with the ground when mounted on the gimbal, it is necessary to collect a series of measurements from both the accelerometer and gyroscope to determine the required offsets. These offsets are crucial for converting data from the Body Frame (the IMU's position on the gimbal) to the Inertial Frame (the reference system used for calculating angles and velocities).

Gyroscope Calibration

Several methods can be employed to calibrate the gyroscopic values. The following algorithm will describe the calibration procedure.

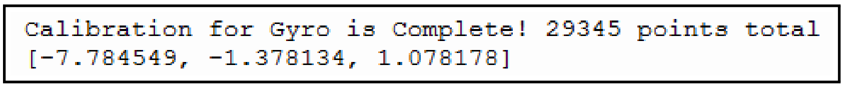
Algorithm 1 Gyroscope Calibration Procedure

```
Input: Calibration time calibration_time in seconds
Output: Offset values offsets for gyro calibration

Start
Initialize MPU6050 sensor with I2C communication
Delay 4 seconds to allow MPU6050 to settle

procedure DEFINE_GET_GYRO
    Function get_gyro
    gx, gy, gz $\leftarrow$ Read gyro values from MPU6050
    Return gx, gy, gz
end procedure

procedure DEFINE_GYRO_CALIBRATION(calibration_time)
    Function gyro_calibration
    offsets $\leftarrow$ [0, 0, 0]
    num_of_points $\leftarrow$ 0
    end_loop_time $\leftarrow$ current_time + calibration_time
    while current_time $<$ end_loop_time do
        num_of_points $\leftarrow$ num_of_points + 1
        (gx, gy, gz) $\leftarrow$ get_gyro()
        offsets[0] $\leftarrow$ offsets[0] + gx
        offsets[1] $\leftarrow$ offsets[1] + gy
        offsets[2] $\leftarrow$ offsets[2] + gz
        if num_of_points % 100 == 0 then
            Print: “Still Calibrating Gyro... num_of_points points so far”
        end if
    end while
    Print: “Calibration for Gyro is Complete! num_of_points points total”
    Compute mean of offsets: offsets $\leftarrow$ offsets / num_of_points
    Return offsets
end procedure

Call gyro_calibration(calibration_time) and print the result
End
```

The code used for this study is available online on GitHub. For detailed code, please refer to the following link [7]. The aim is to minimize the offsets as much as possible.

[7] https://shorturl.at/u52J8

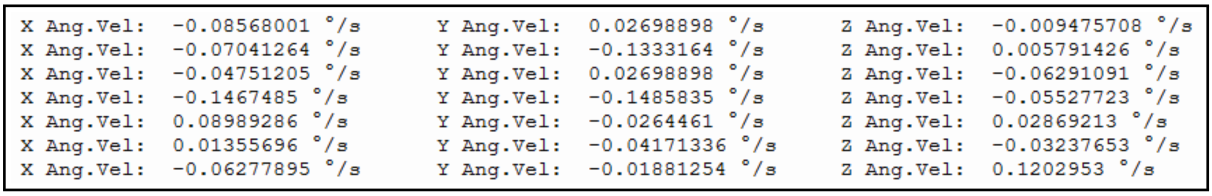
After completing the calibration process, 3 offsets values were provided:



```
Calibration for Gyro is Complete! 29345 points total
[-7.784549, -1.378134, 1.078178]
```

Figure III.19: Final gyroscope calibration offsets.

To align the body frame of the gimbal with the inertial frame, the offsets must be subtracted from the gyroscope outputs. Figure III.20 illustrates the calibrated outputs.



```
X Ang.Vel:  -0.08568001 °/s     Y Ang.Vel:  0.02698898 °/s     Z Ang.Vel:  -0.009475708 °/s
X Ang.Vel:  -0.07041264 °/s     Y Ang.Vel:  -0.1333164 °/s     Z Ang.Vel:  0.005791426 °/s
X Ang.Vel:  -0.04751205 °/s     Y Ang.Vel:  0.02698898 °/s     Z Ang.Vel:  -0.06291091 °/s
X Ang.Vel:  -0.1467485 °/s      Y Ang.Vel:  -0.1485835 °/s     Z Ang.Vel:  -0.05527723 °/s
X Ang.Vel:  0.08989286 °/s      Y Ang.Vel:  -0.0264461 °/s     Z Ang.Vel:  0.02869213 °/s
X Ang.Vel:  0.01355696 °/s      Y Ang.Vel:  -0.04171336 °/s    Z Ang.Vel:  -0.03237653 °/s
X Ang.Vel:  -0.06277895 °/s     Y Ang.Vel:  -0.01881254 °/s    Z Ang.Vel:  0.1202953 °/s
```

Figure III.20: Gyroscope raw data readings from MPU6050 after calibration.

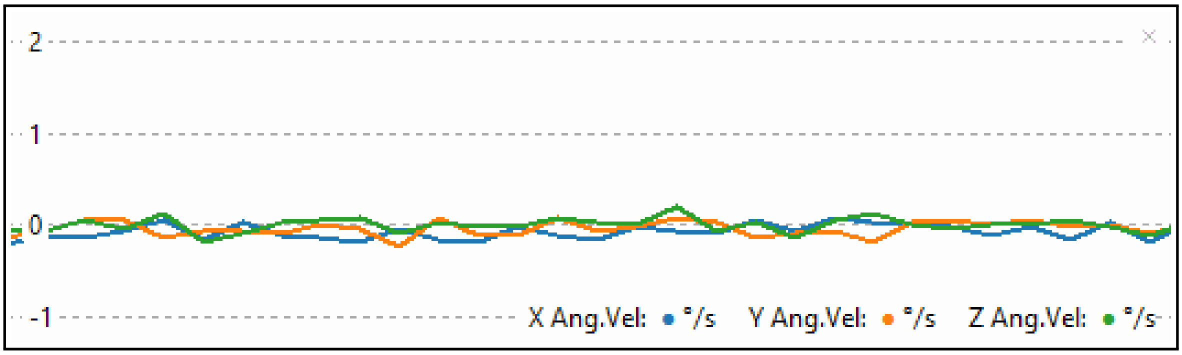


Figure III.21: Visualization of gyroscope signals after calibration.

Accelerometer Calibration

According to the accelerometer data, calibration was not necessary. However, to calibrate the accelerometer outputs, you can use the code available on GitHub, accessible via the link below [8].

[8] https://shorturl.at/p69Z9

Visualizing Angles from IMU Sensors

The formulas discussed in Chapter II (II.2.1)(II.2.1) were implemented to calculate and display the angles derived from both the gyroscope and the accelerometer. In Figures III.22 and III.23 the drift from the gyroscope data and the noise of the accelerometer data over time can be observed in a steady state.

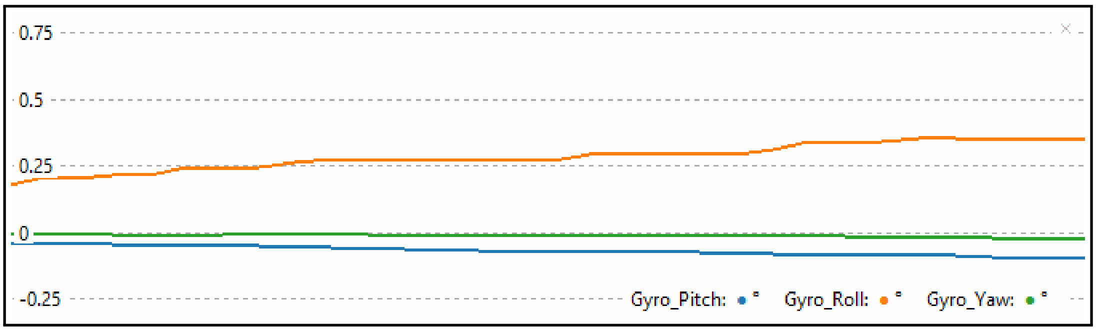


Figure III.22: Gyroscope Drift over time.

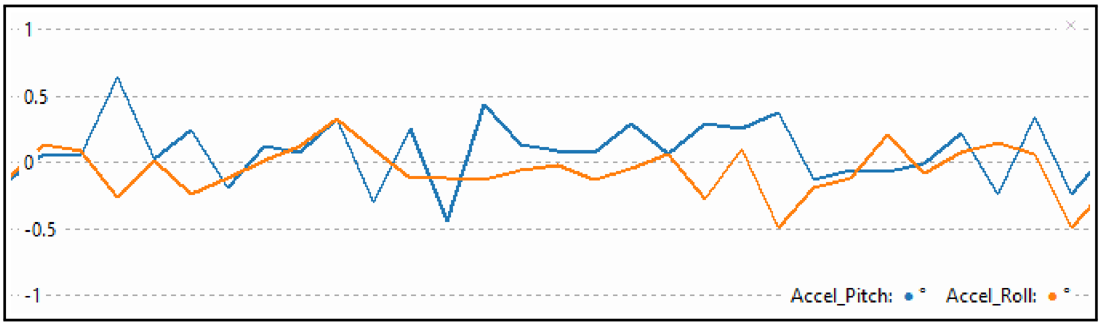


Figure III.23: Accelerometer Noise over time.

This method results in progressively inaccurate measurements over time due to the accumulation of integration errors, ultimately leading to angular drift.

The angle derived from acceleration is frequently subject to significant noise and disturbances, particularly from minor movements.

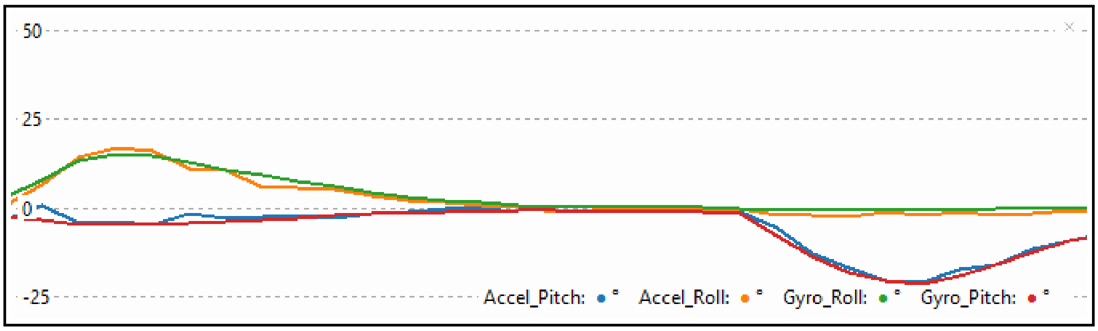


Figure III.24: Comparison of angle estimations: Gyroscope vs Accelerometer.

III.5.2 Implementing the Complementary Filter for IMU Data Fusion

To improve the stability of angle estimation and to ensure an accurate determination of the camera's orientation, data from both the gyroscope and the accelerometer were processed and fused using a complementary filter. In practice, this filter is implemented in a discrete form, where sensor readings are processed at small time intervals. The update equation is applied iteratively, enabling the filter to generate a continuous estimate of the orientation.
The combination of accelerometer and gyroscope data using the complementary filter is expressed as follows:

$$\phi_{\text{est}}(t) = \alpha\left(\phi_{\text{est}}(t-1) + \omega_{x,\text{gyro}} \cdot \Delta t\right) + (1-\alpha)\phi_{\text{acc}}$$
$$\theta_{\text{est}}(t) = \alpha\left(\theta_{\text{est}}(t-1) + \omega_{y,\text{gyro}} \cdot \Delta t\right) + (1-\alpha)\theta_{\text{acc}}$$

By implementing the sensor fusion algorithm, data from the gyroscope and accelerometer are integrated into a unified signal, with different weights assigned to each, as described in Eq.(II.6). The appropriate weighting values for these filters were determined empirically through trial and error.

The accelerometer data was given a weight of 0.01 (1%), while the gyroscope data was assigned a weight of 0.99 (99%), resulting in a smooth and filtered signal of Roll and Pitch angles, as shown in Figures III.25 and III.26.

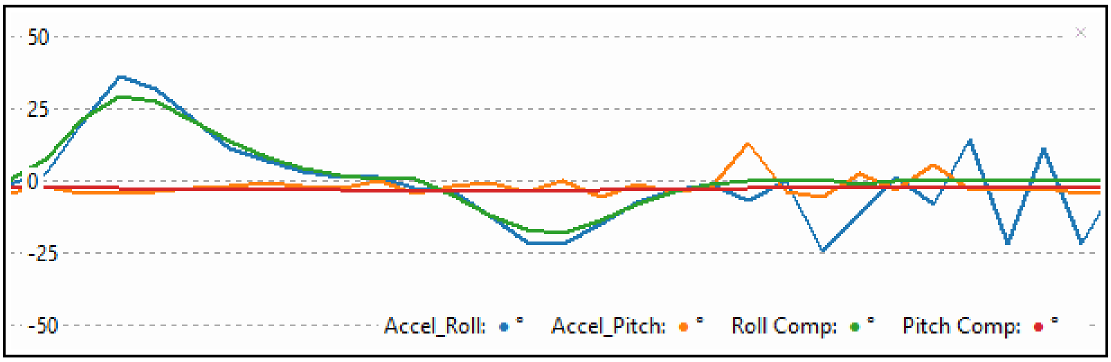


Figure III.25: The filtered Roll angle signal using complementary filter.

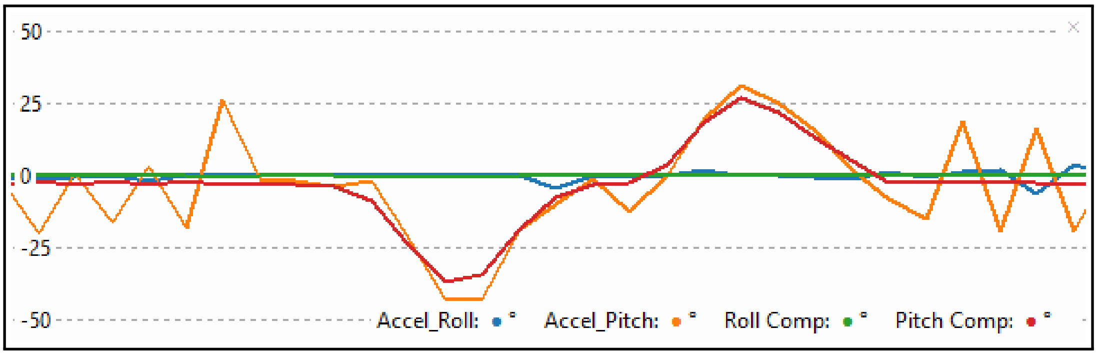
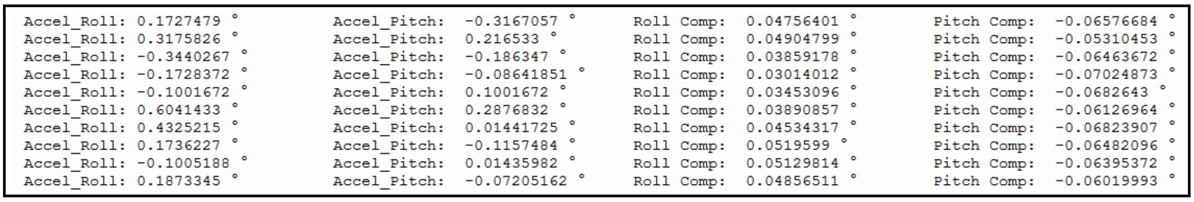


Figure III.26: The filtered Pitch angle signal using complementary filter.

Figure III.27 illustrates the stability of the roll and pitch angles after applying the complementary filter, in comparison to the accelerometer data.



```
Accel_Roll: 0.1727479 °       Accel_Pitch:  -0.3167057 °      Roll Comp:  0.04756401 °        Pitch Comp:  -0.06576684 °
Accel_Roll: 0.3175826 °       Accel_Pitch:  0.216533 °        Roll Comp:  0.04904799 °        Pitch Comp:  -0.05310453 °
Accel_Roll: -0.3440267 °      Accel_Pitch:  -0.186347 °       Roll Comp:  0.03859178 °        Pitch Comp:  -0.06463672 °
Accel_Roll: -0.1728372 °      Accel_Pitch:  -0.08641851 °     Roll Comp:  0.03014012 °        Pitch Comp:  -0.07024873 °
Accel_Roll: -0.1001672 °      Accel_Pitch:  0.1001672 °       Roll Comp:  0.03453096 °        Pitch Comp:  -0.0682643 °
Accel_Roll: 0.6041433 °       Accel_Pitch:  0.2876832 °       Roll Comp:  0.03890857 °        Pitch Comp:  -0.06126964 °
Accel_Roll: 0.4325215 °       Accel_Pitch:  0.01441725 °      Roll Comp:  0.04534317 °        Pitch Comp:  -0.06823907 °
Accel_Roll: 0.1736227 °       Accel_Pitch:  -0.1157484 °      Roll Comp:  0.0519599 °         Pitch Comp:  -0.06482096 °
Accel_Roll: -0.1005188 °      Accel_Pitch:  0.01435982 °      Roll Comp:  0.05129814 °        Pitch Comp:  -0.06395372 °
Accel_Roll: 0.1873345 °       Accel_Pitch:  -0.07205162 °     Roll Comp:  0.04856511 °        Pitch Comp:  -0.06019993 °
```

Figure III.27: Stability enhancement of Roll and Pitch angles using the complementary filter.

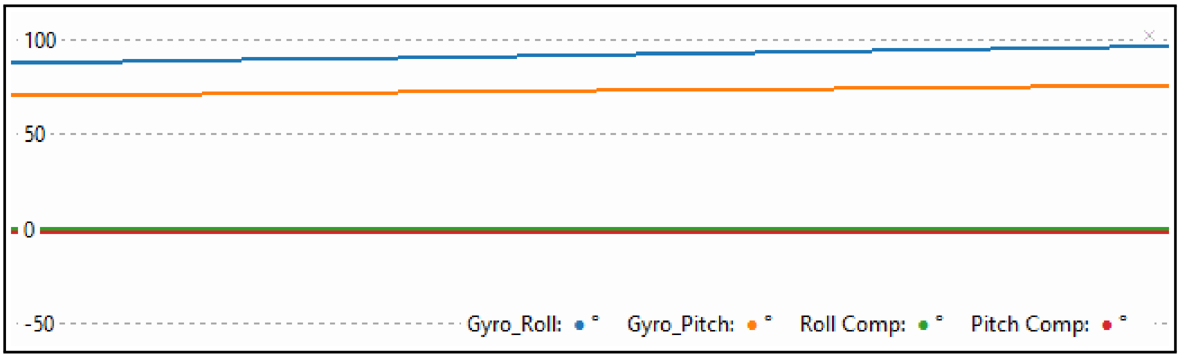


Figure III.28: Drift correction of Roll and Pitch angles using the complementary filter.

This approach allowed the accelerometer to correct the gyroscope's long-term drift, while the gyroscope smoothed out short-term variations. The result was a stable estimate of orientations.

III.5.3 Integration of Filtered Data into Control Algorithm

After calculating the required angles, the camera's positional data is acquired by the sensor. The project utilizes three MG996R servomotors, controlled by a motor control system that includes the RPi Pico W, a PWM driver (PCA9685), the MPU6050 sensor, and the RPi 4. The RPi Pico W calculates the angles from the IMU sensor and transmits this data via the UART protocol to the RPi 4. The RPi 4, connected to the PCA9685, sends commands to the motor driver, which generates the PWM signals needed to regulate the servomotors based on the received angular data.

A PID controller is integrated into the system to autonomously regulate motor direction in response to the camera's position. This controller establishes a feedback loop between motor control and sensor data acquisition, adjusting motor positions across the axes using PWM signals. The PID controller minimizes error by applying corrections based on the constants $K_p = 1$, $K_i = \frac{1}{200}$, and $K_d = \frac{1}{600}$, ensuring system responsiveness and stability. The algorithm flowchart is depicted in Figure III.29.

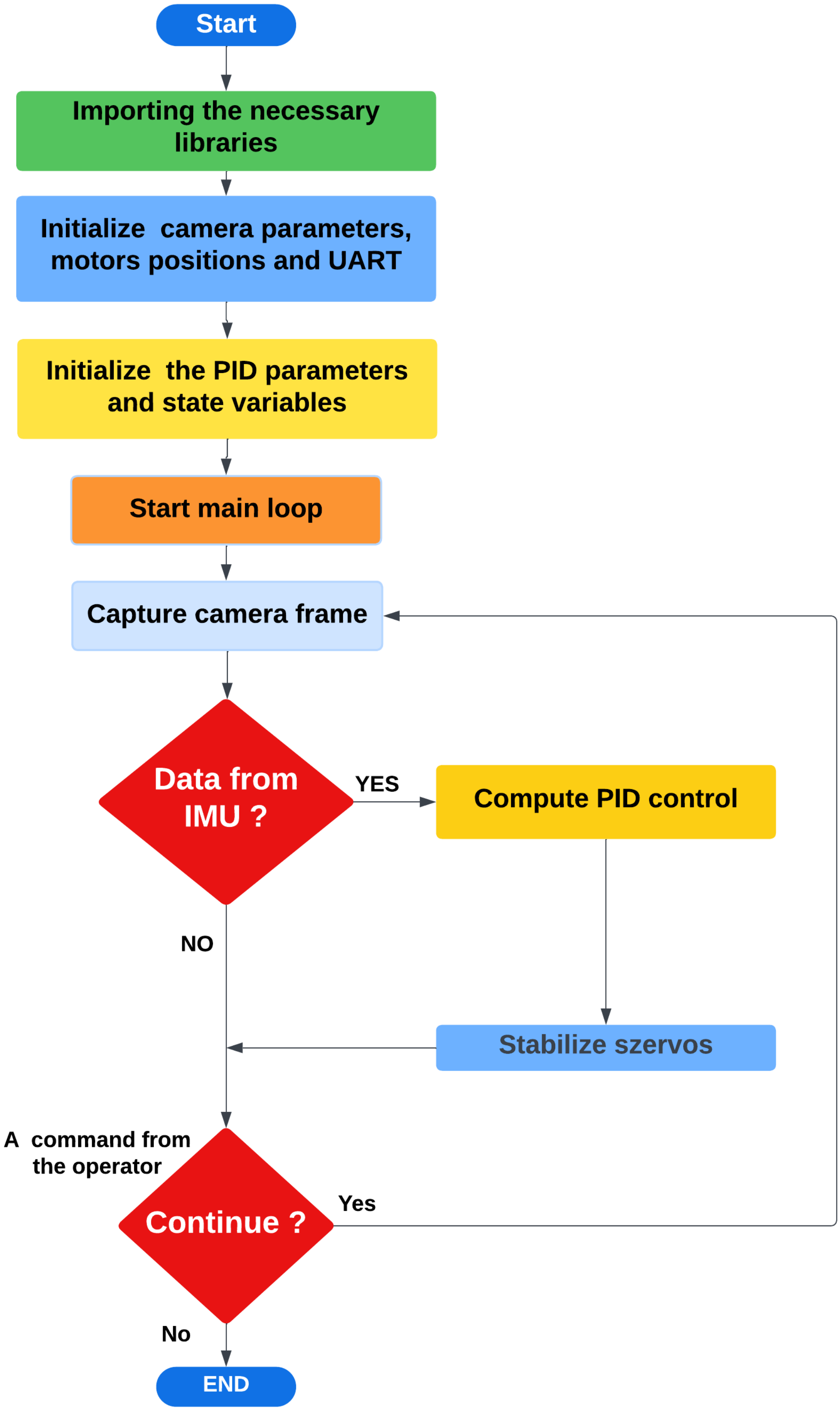

Start
Importing the necessary libraries
Initialize camera parameters, motors positions and UART
Initialize the PID parameters and state variables
Start main loop
Capture camera frame
Data from IMU ?
YES
Compute PID control
NO
Stabilize szervos
A command from the operator
Continue ?
Yes
No
END

Figure III.29: Flowchart of the Stabilization Algorithm.

Stabilization Control Results

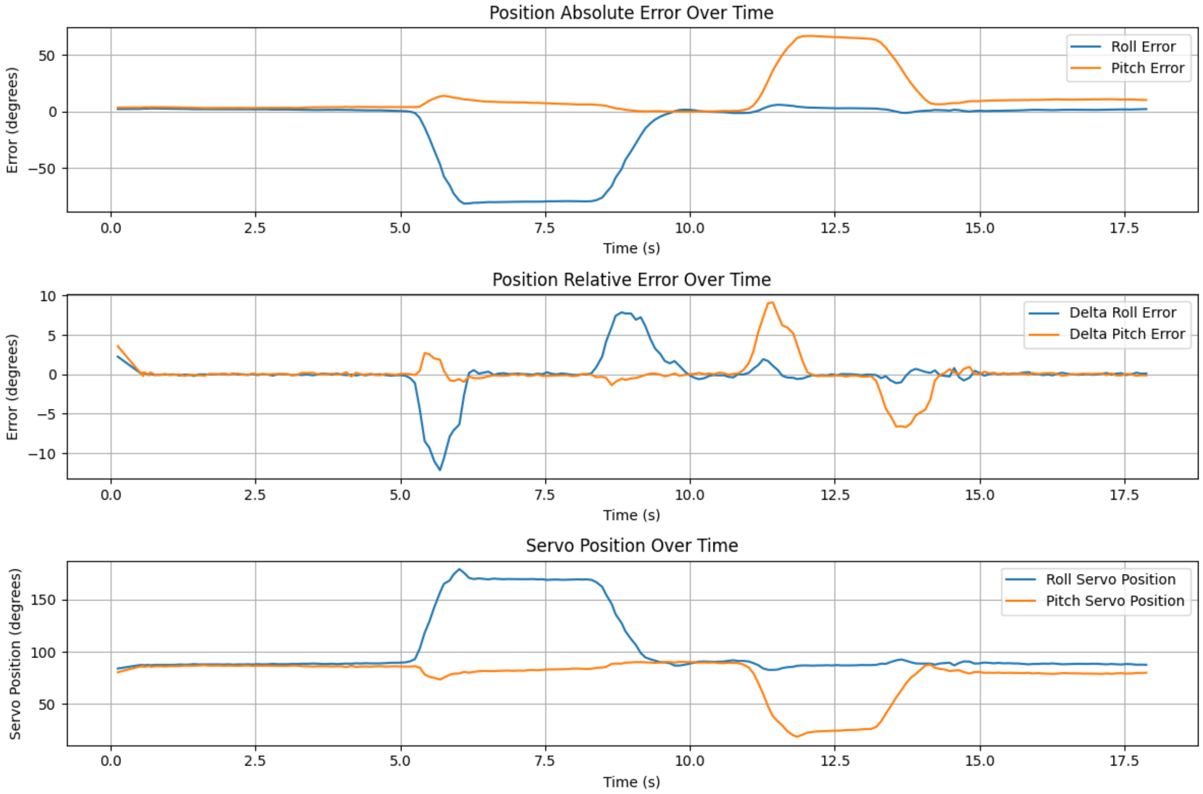


Figure III.30: Results of stabilization control.

Figure III.30 presents the results of the stabilization control. The graphs demonstrate how the control system dynamically compensates for disturbances in both axes by adjusting the servo positions to counteract the deviations, thereby restoring stability to the system.

Initially, the graphs show that the roll and pitch errors are effectively zero, indicating that the system is stable with no significant deviations from the desired orientation. This suggests that the control system is maintaining the roll and pitch angles as intended.

At around 5 seconds, a perturbation is introduced, causing noticeable deviations in the roll and pitch angles. Specifically, the roll axis undergoes a substantial displacement of approximately -90°, indicating a significant deviation from its initial position. Simultaneously, the pitch axis experiences a smaller but still notable deviation of about 10°.

The third graph, which plots the servo positions, clearly reflects the control system's response to these deviations. To counteract the -90° deviation in the roll axis, the control system adjusts the roll angle to approximately 180°, effectively bringing the system back toward the desired orientation. This significant correction highlights the system's capability to handle large disturbances.

For the pitch axis, the control system makes a more modest adjustment, shifting the angle from 90° to around 80°. This smaller correction corresponds to the lesser deviation observed in the pitch axis, demonstrating the control system's proportional response to the magnitude of the disturbance.

In the second graph, the relative error plots indicate that the control system is actively correcting the errors introduced by the perturbation, making continuous adjustments to return the system to equilibrium.

This behavior aligns with the principles discussed in Chapter I.3.2, where the system's dynamic response to disturbances and its subsequent correction mechanisms were outlined.

III.6 Object Detection

Yolov8n-based Detection

The primary objective in this project is to ensure that the airborne camera remains focused on the object of interest, which necessitates an accurate detection and tracking mechanism.

The work started with implementing the YOLOv8n nano model on the Raspberry Pi 4. As the smallest and fastest model in the YOLOv8 family, it is designed for scenarios with constrained computational resources.

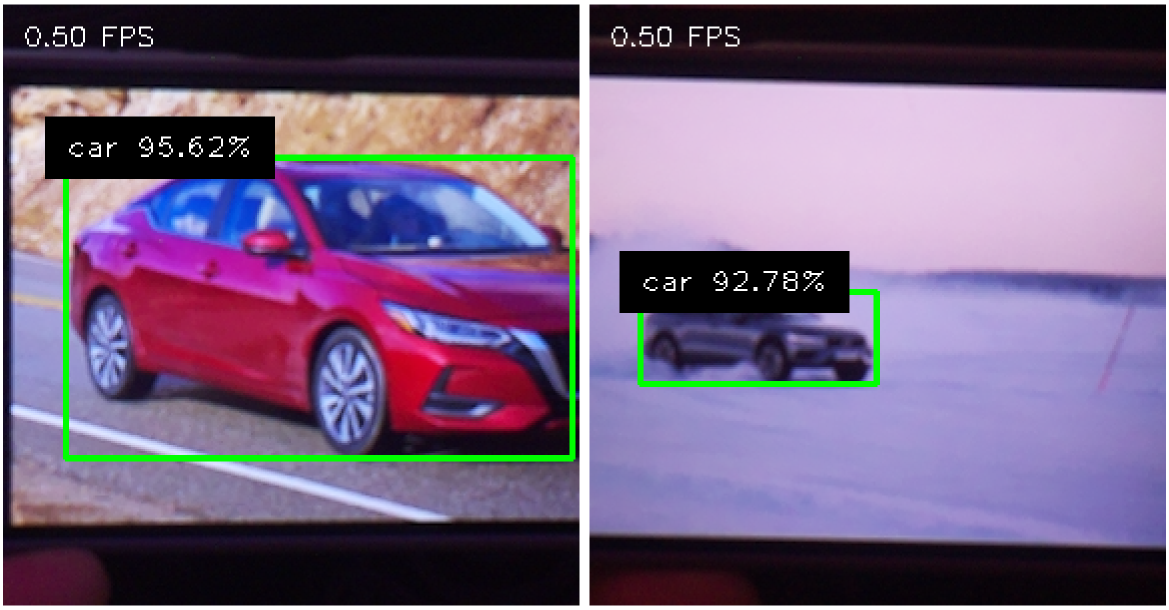


Figure III.31: Detection results using YOLOv8n model at a resolution of 300 × 300 pixels.

MobileNet SSD-based Detection

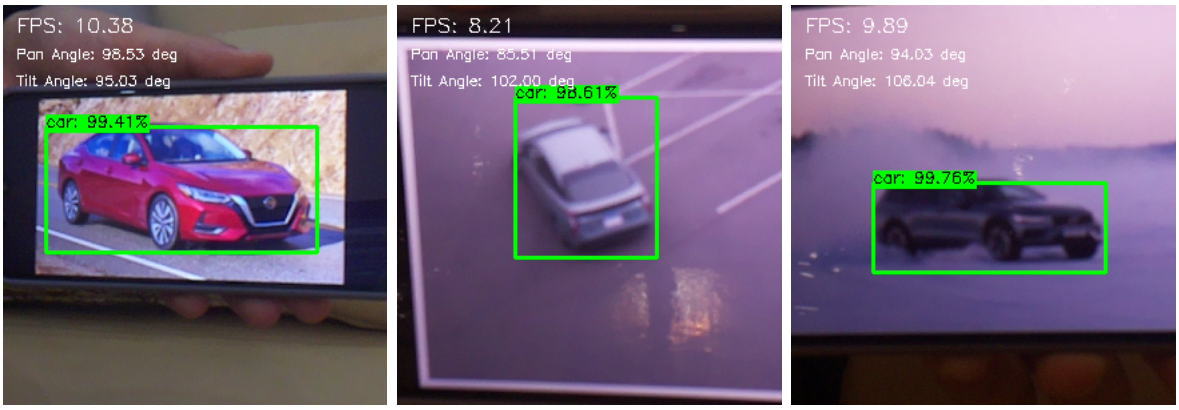


Figure III.32: Detection results using MobileNet SSD at a resolution of 300 × 300 pixels.

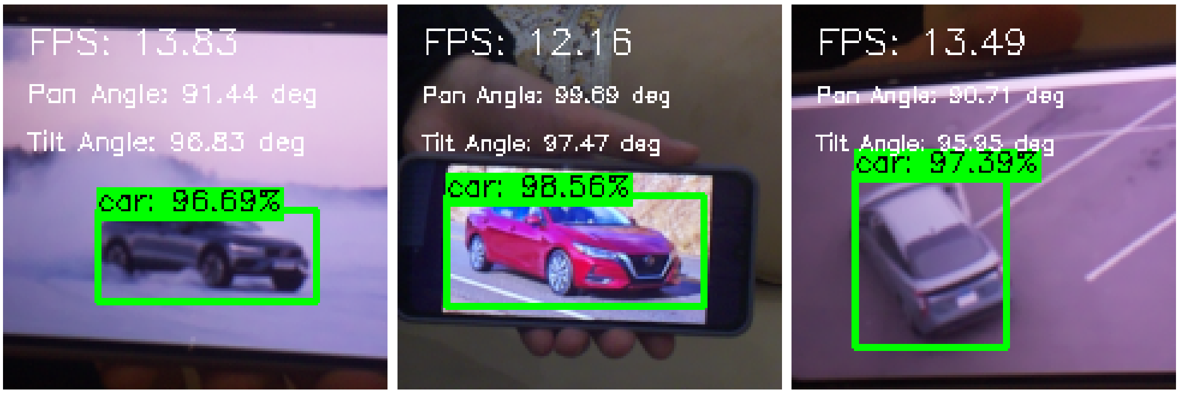


Figure III.33: Detection results using MobileNet SSD at a resolution of 160 × 160 pixels.

A closer look at Figures III.33 and III.32 reveals that the implementation of the YOLOv8n model resulted in 0.5 FPS, while we achieved around 13 FPS with MobileNet SSD at a resolution of 160 × 160. Consequently, the servo control was slow, necessitating the adoption of a different approach.

Given the constraints of the Raspberry Pi 4, which has a quad-core ARM Cortex-A72 CPU and 4GB of RAM III.1, the detection algorithm needed to be both computationally lightweight and effective. Therefore, color-based detection was selected as the primary method for object identification due to its relatively low computational requirements and the effectiveness of color as a distinguishing feature in many scenarios.

Color-based Detection

Target detection is performed using an OpenCV-based application [9].The process converts the input image from RGB to HSV color space, defining a specific range of HSV values to capture the target object's color. These thresholds are determined experimentally under different lighting conditions to isolate the object effectively. A binary mask is then created, where pixels within the range are set to white (1), representing the object, and those outside the range are set to black (0), representing the background.

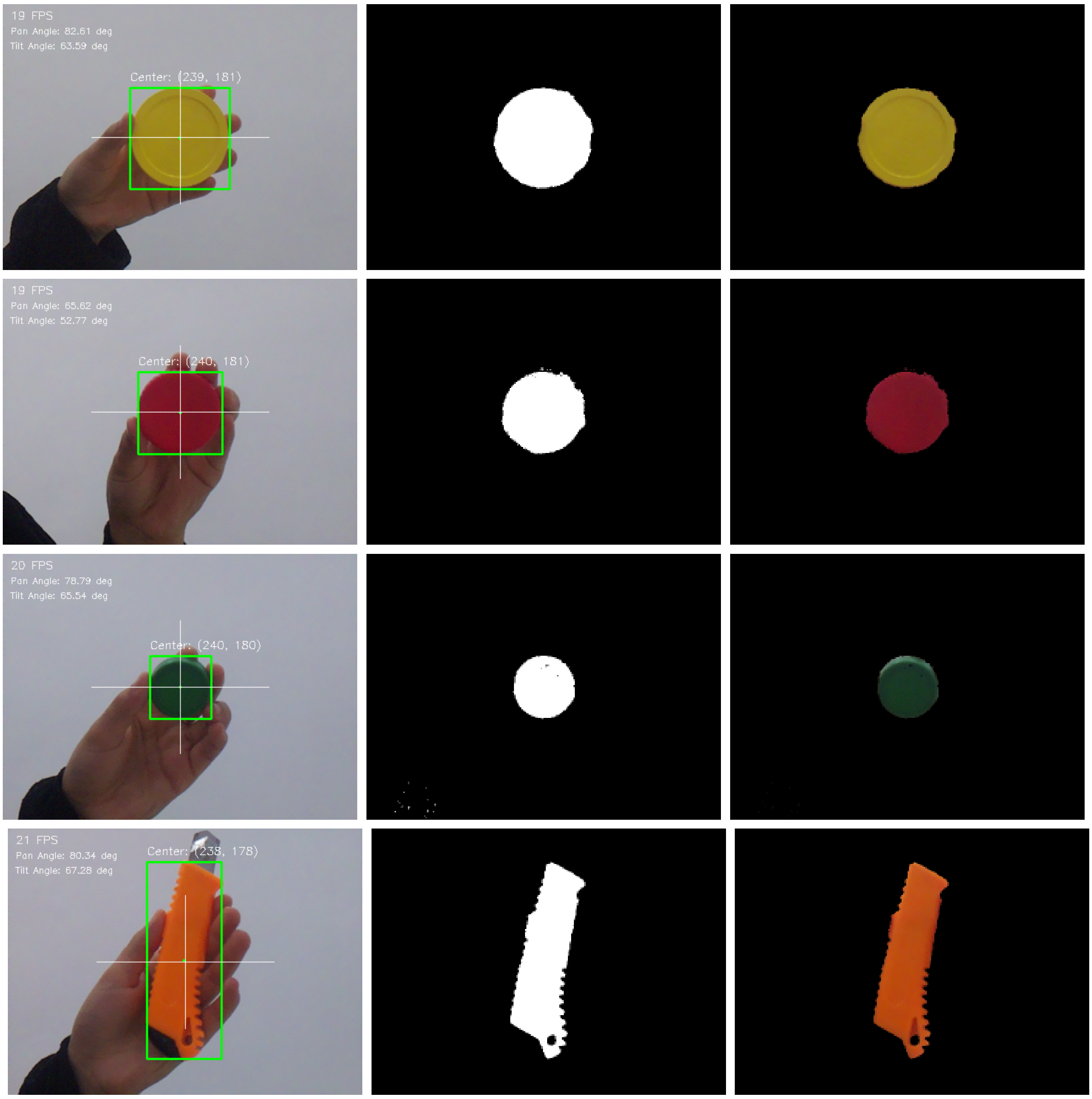


Figure III.34: Detection of Various Colors Based on the Defined HSV Ranges.

[9] Open Source Computer Vision Library (OpenCV) is an open-source software library that provides a vast collection of tools and functions for computer vision, image processing, and machine learning. It is widely used in a variety of real-time applications, including object detection, image recognition, and video processing.

While color-based detection is computationally efficient, it presents several challenges:

- **Lighting Sensitivity:** Changes in lighting can affect the perceived color of the object, potentially leading to detection inaccuracies. This challenge is mitigated by using the HSV color space, but further optimization may be necessary for environments with extreme lighting conditions.
- **Background Interference:** If the background contains colors similar to the object, false positives may occur. The system's accuracy can be improved by refining the color thresholds or incorporating additional features such as shape or size.
- **Object Color Variability:** Objects that do not have a uniform color or that change color (e.g., due to rotation or deformation) can be challenging to detect consistently.

III.6.1 Integration of Visual Data into Control Algorithm

The final and most critical phase involves integrating visual data into the control algorithm governing the gimbal system. This process requires converting visual inputs into actionable control commands, which dynamically adjust the camera's orientation to maintain focus on the object of interest according to the algorithm in Figure III.35.

Image sequences were streamed at a resolution of 480 × 360 pixels and a frame rate of 60 fps. The control objective was to position the target at the center of the image plane with zero steady-state error and a smooth transient response, ensuring effective visual tracking.

The control system operates in a closed-loop configuration, starting with object detection, followed by error computation, control signal generation, and the subsequent adjustment of the camera's orientation. A proportional-derivative (PD) control approach is implemented to progressively minimize the error, ensuring the object remains as close to the center of the frame as possible, with $K_p = \frac{1}{200}$, and $K_d = \frac{1}{800}$.

Once the object is detected and its centroid is identified in the image plane coordinates, the tracking angles are calculated by the core processing unit (RPi 4). Continuous feedback from visual data informs servo adjustments, minimizing the deviation between the object's centroid and the center of the image frame, despite any movement of either the camera or the object. The performance of the tracking system is evaluated based on its ability to quickly reduce errors and maintain a stable lock on the object.

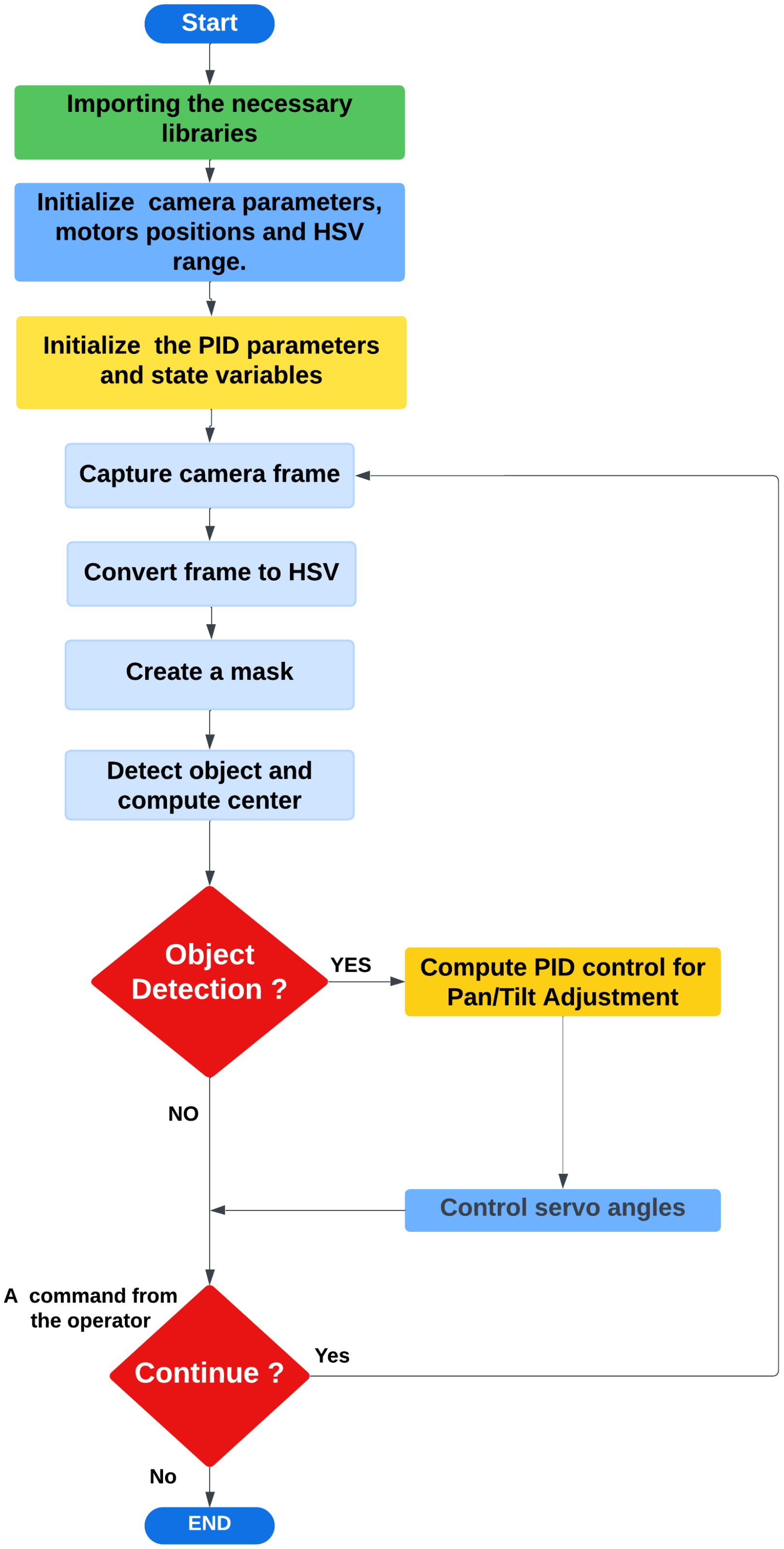

Start
Importing the necessary libraries
Initialize camera parameters, motors positions and HSV range.
Initialize the PID parameters and state variables
Capture camera frame
Convert frame to HSV
Create a mask
Detect object and compute center
Object Detection ?
YES
Compute PID control for Pan/Tilt Adjustment
NO
Control servo angles
A command from the operator
Continue ?
Yes
No
END

Figure III.35: Flowchart of the Tracking Algorithm.

Tracking Results

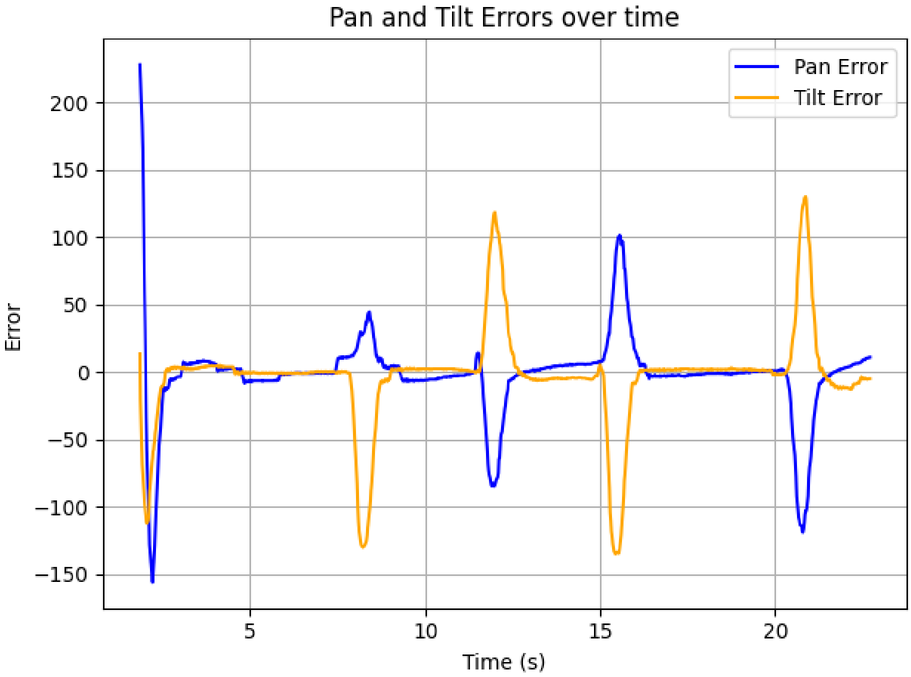


Figure III.36: Results of tracking control.

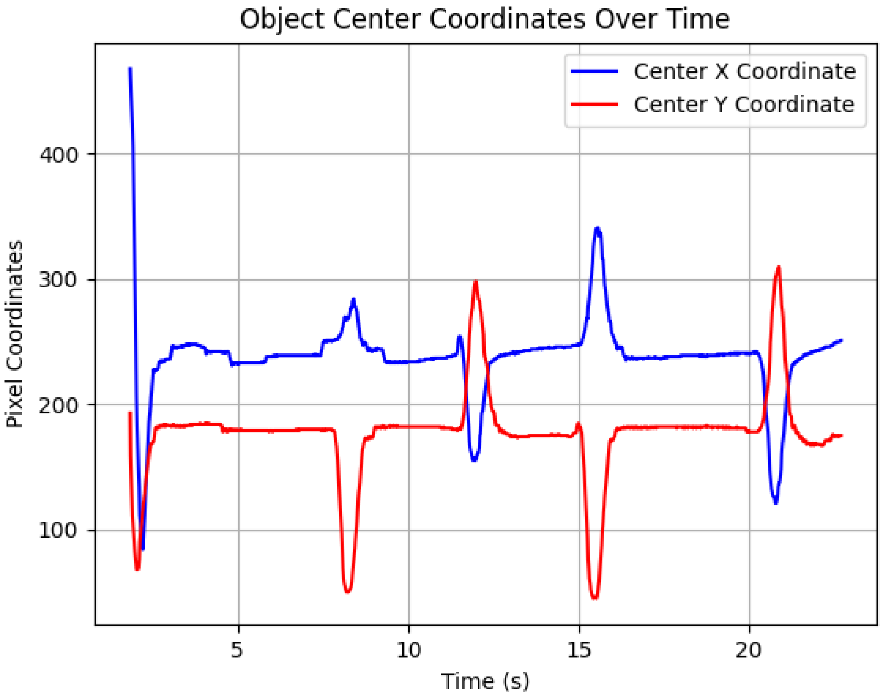


Figure III.37: Center of the object over time.

As depicted in Figures III.36 and III.37, the control system actively minimizes both pan and tilt errors after 4 movements. The optimal tracking position is (240, 180), which represents the center of the frame. The RMSE is calculated to verify the precision of the tracking.

The Root Mean Square Error (RMSE)

The RMSE is a commonly used metric to measure the differences between values predicted by a model and the actual observed values.

RMSE can be used to evaluate how well the control system maintains the position of the object's centroid relative to the desired target position, typically the center of the image frame (240, 180), and provides a quantitative measure of the control system's tracking performance. A lower RMSE indicates more precise tracking. The RMSE is given by the following formula:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{III.1}$$

Where:

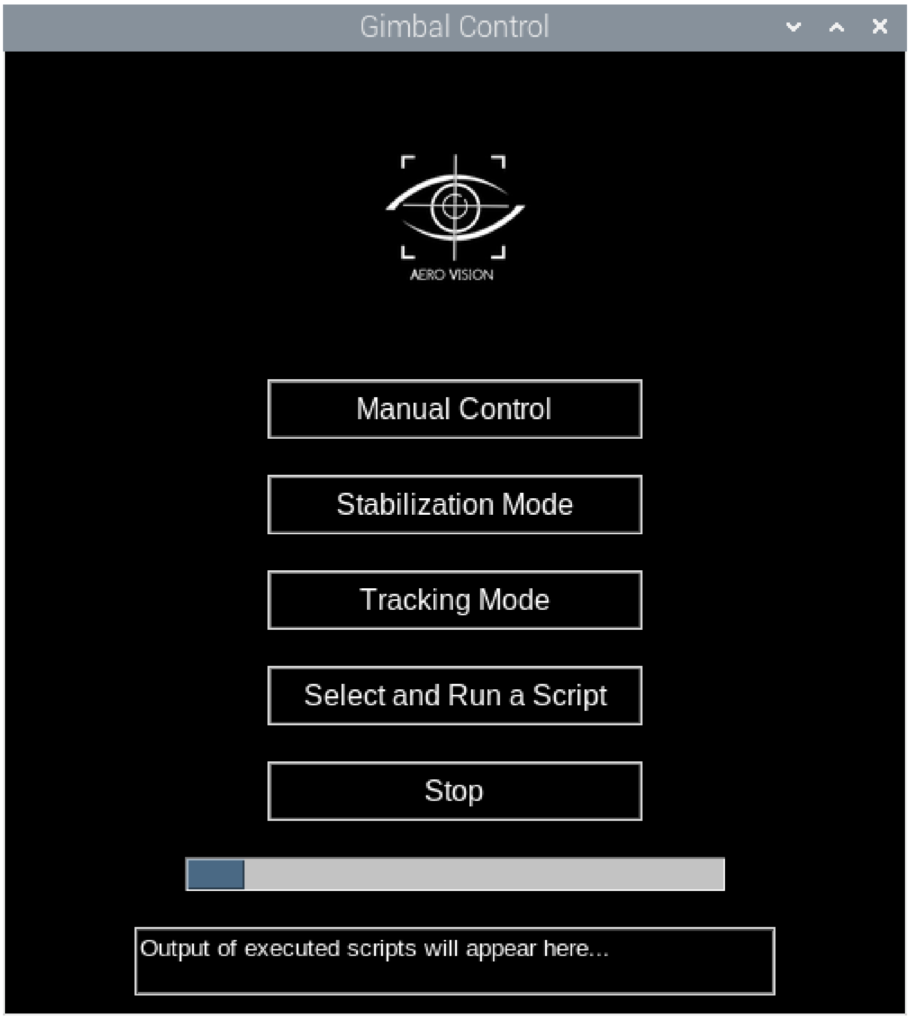
- y_i is the observed position of the object's centroid,
- $\hat{y}_i$ is desired target position,
- n is the number of observations.

The obtained RMSE values were 23 for the pan error and 20 for the tilt error.

III.7 Graphical User Interface (GUI) for Gimbal Control

In order to facilitate efficient manipulation and control of the gimbal system, a Graphical User Interface (GUI) was developed III.38. The GUI was created to offer an intuitive and user-friendly platform, allowing users to easily switch between various gimbal operation modes such as manual control, stabilization mode, and tracking mode. Additionally, the interface enables the execution of predefined scripts and provides real-time feedback on system performance through an output display.

Moreover, the GUI also includes an integrated progress bar to indicate the execution status of operations, ensuring the user is constantly informed of the system's state.



Output of executed scripts will appear here...

Figure III.38: Graphical User Interface (GUI) for Gimbal Control.

III.8 Discussion

Despite some limitations, the gimbal's performance is generally satisfactory but leaves room for further improvement

Stabilization

In terms of stabilization, the gimbal effectively stabilizes two of the three axes using input from the IMU sensor to detect the camera's angular position and adjust each motor to achieve the desired angle. While the stabilization process works well on the roll and pitch axes, it lacks precision on the yaw axis.

The filter applied in this project effectively reduced unpredictable spikes and inconsistencies from the accelerometer, as well as drift from the gyroscope. Test runs and subsequent analysis revealed that the complementary filter significantly improved overall performance.

While obtaining roll and pitch values from the IMU was relatively straightforward, acquiring accurate yaw values posed a greater challenge. A simple yaw reading could be obtained by integrating the gyroscope values, but this method resulted in significant drift over time, rendering it unreliable for accurately determining the yaw angle.

Visual Servoing

The first issue encountered was the system's poor performance when YOLOv8n model was implemented. Therefore, we switched to MobileNet SSD, where the frames per second (FPS) increased to an average of 13 FPS, but it did not meet the control requirements. These models were not suitable due to the computational constraints of the RPi 4.

For real-time tasks, external hardware accelerators, such as the Coral USB Accelerator, which is a device that connects via USB and includes a Google Edge TPU [10] coprocessor, specifically designed to accelerate machine learning models, particularly for low-power and edge devices, or an NVIDIA Jetson device, would have been necessary, but they were not available.

Consequently, the detection algorithm needed to be both lightweight and efficient. Thus, color-based detection was chosen as the primary method for object detection due to its low computational requirements, achieving a performance of 60 fps.

For the visual servoing task, the gimbal successfully extracted visual data from the camera sensor and integrated this information into the control loop. Horizontal and vertical errors were quickly minimized, ensuring a stable lock on the object of interest, with a root mean square error (RMSE) of 23 for pan error and 20 for tilt error.

One major limitation of the current system is the quality of the camera used, which restricts its effectiveness under real-world sunlight conditions. Future work should focus on integrating a higher-performance camera, enabling better functionality in various lighting conditions and improving the system's overall tracking accuracy.

PID Parameters Tuning

The PID parameters were determined through trial and error, ultimately set to values that yielded acceptable results with minimal instability for both camera stabilization (using PID) and visual tracking (using PD). However, further refinement of these parameters may enhance the system's responsiveness, improving the overall stability of the camera.

One major limitation of the current system is the quality of the camera used, which restricts its effectiveness under real-world sunlight conditions. Future work should focus on integrating a higher-performance camera, enabling better functionality in various lighting

[10] **TPU:** Tensor Processing Unit is a type of chip designed by Google specifically to accelerate machine learning tasks, especially those involving neural networks, making them faster and more efficient compared to regular processors.

conditions and improving the system's overall tracking accuracy.

III.9 Conclusion

This chapter has presented a comprehensive overview of the implementation and testing of the proposed system, with a focus on the practical application of the theoretical concepts discussed in earlier chapters.

The prototype development involved meticulous assembly and wiring, which provided a stable foundation for the subsequent gyrostabilization and detection functionalities.

The successful processing and calibration of IMU data, coupled with the implementation of a complementary filter, enabled accurate data fusion, which is critical for the system's stability and control.

The integration of visual data through color-based detection further enhanced the system's capabilities, enabling effective detection and tracking. This approach offered a practical solution to the challenges posed by limited computational resources.

Overall, the implementation phase has demonstrated the feasibility of the proposed design, effectively addressing both the technical challenges and the functional requirements of the system.

The results of this chapter provide a solid foundation for future work, including the refinement and optimization of the system. Potential improvements may involve the integration of more advanced techniques, such as machine learning-based detection, advanced control algorithms, as processing capabilities evolve.

General Conclusion

The study presented in this thesis successfully developed and tested a prototype, illustrated in Figure III.12, for a stabilized camera control system using a 3-axis gimbal with three servomotors for UAV-based active object tracking.

Throughout the project, several challenges were encountered, leading to the discontinuation of certain concepts due to time constraints. The design and development of the gimbal's control system proved to be more time-consuming than initially anticipated. Additionally, interfacing the system components and achieving seamless integration of their functionalities presented further difficulties. The project addressed critical challenges related to stabilizing the camera on three axes—roll, pitch, and yaw—while integrating visual data to enable real-time tracking of moving targets.

In terms of stabilization, the system performance is generally satisfactory on the roll and pitch axes, as shown in the experimental results, owing to the accurate processing of IMU data. The complementary filter applied to sensor data significantly reduced noise and improved the precision of stabilization. However, yaw axis stabilization proved to be more challenging due to gyroscope drift, highlighting the need to fuse the gyroscope data with another sensor data.

For object tracking, several detection methods were evaluated, including advanced neural networks such as YOLOv8n and MobileNet SSD. However, due to the computational limitations of the Raspberry Pi 4, these models did not meet real-time control requirements. A lightweight color-based detection algorithm was therefore implemented, achieving an impressive frame rate of 60 FPS. The successful integration of visual data into the control system enabled reliable tracking of the target, with a root mean square error (RMSE) of 23 for pan error and 20 for tilt error.

The implementation of PID controllers played a crucial role in stabilizing the system for both camera stabilization and object tracking. While the selected PID and PD parameters were sufficient for basic operation, further tuning could enhance the system's responsiveness and reduce instability, particularly in dynamic environments.

The first prototype, Aero Vision 0.1, served as a proof of concept for the system, demonstrating the viability of the design. Additionally, the study emphasizes the potential for developing domestic capabilities in Algeria to produce such systems, which holds significant promise for both economic growth and strategic security.

Perspectives

Despite these achievements, there remains significant room for further enhancement and future research.

First, the gimbal's mechanical design will be revised to meet specific performance requirements, with brushless motors replacing servomotors to enable smoother and more precise motion. Refining yaw axis stabilization is also a key priority, with the integration of a magnetometer and advanced filtering techniques to provide better long-term stability and eliminate drift. Achieving full stabilization across all axes is essential.

To improve detection and tracking accuracy, higher-performance cameras will be incorporated, enhancing functionality in various lighting conditions. Additionally, more complex object detection algorithms will be explored to improve performance in challenging environments with complex backgrounds. These algorithms will be supported by more powerful computing platforms, such as NVIDIA devices, and will include depth extraction for 3D target awareness, further enhancing tracking precision.

Integrating the inertial stabilization system with the tracking algorithm will optimize system performance, particularly in dynamic environments. Future research could also focus on adaptive PID tuning methods that respond to varying environmental conditions or investigate alternative control strategies, such as sliding mode control (SMC), fuzzy logic, or neural network-based control systems. These approaches could offer improved stability and adaptability in unpredictable environments.

For the system to become fully operational, extensive field testing will be necessary. Future work should involve testing the tracking algorithm on real UAVs under various conditions to identify practical limitations and provide valuable insights for refining both hardware and software components. In parallel, developing a user interface will facilitate easier human-machine interaction, enhancing usability and control.

Bibliography

[1] Vestibulo-ocular function during co-ordinated head and eye movements to acquire visual targets. - Barnes - 1979 - The Journal of Physiology - Wiley Online Library.

[2] The human eye: a biological camera. PM324:105–139, May 2021. Publisher: SPIE.

[3] M Ashok Kumar and S Kanthalakshmi. H control law for line of sight stabilization in two-axis gimbal system. *Journal of Vibration and Control*, 28(1-2):182–191, 2022.

[4] Karl Johan Åström and Richard Murray. *Feedback systems: an introduction for scientists and engineers.* Princeton university press, 2021.

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14*, pages 850–865. Springer, 2016.

[6] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[7] François Chaumette and Seth Hutchinson. Visual servo control. ii. advanced approaches [tutorial]. *IEEE Robotics & Automation Magazine*, 14(1):109–118, 2007.

[8] Rita Cunha, Miguel Malaca, Vasco Sampaio, Bruno Guerreiro, Paraskevi Nousi, Ioannis Mademlis, Anastasios Tefas, and Ioannis Pitas. *Gimbal Control for Vision-based Target Tracking.* September 2019.

[9] Rita Cunha, Miguel Malaca, Vasco Sampaio, Bruno Guerreiro, Paraskevi Nousi, Ioannis Mademlis, Anastasios Tefas, and Ioannis Pitas. Gimbal control for vision-based target tracking. In *Proceedings of the European Conference on Signal Processing*, 2020.

[10] Goutam Das, Nurul Anwar, Shovan Chowdhury, and Kazi Afzalur Rahman. Design and fabrication of an image processing based autonomous weapon. *International Journal of Engineering Research*, 5(12):931–935, 2016.

[11] Dmytro Dashkov and Oleksii Liashenko. Motion capture with mems sensors. *Advanced Information Systems*, 7(2):57–62, 2023.

[12] Raspberry Pi Foundation. Raspberry pi pico w datasheet, 2023.

[13] Raspberry Pi Foundation. Raspberry pi 4 model b datasheet, 2024.

[14] Raspberry Pi Foundation. Raspberry pi camera module v2 datasheet, 2024.

[15] IMU Data Fusing. Complementary, kalman, and mahony filter. *URL: http://www. olliw. eu/2013/imu-data-fusing/.(accessed: May 23, 2018)*, 2013.

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[17] Jakob Hansen and Rui Figueiredo. Active Object Detection and Tracking Using Gimbal Mechanisms for Autonomous Drone Applications. *Drones*, 8:55, February 2024.

[18] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[19] José Luis Hernández-Hernández, Mario Hernández-Hernández, Severino Feliciano-Morales, Valentín Álvarez-Hilario, and Israel Herrera-Miranda. Search for optimum color space for the recognition of oranges in agricultural fields. In *International Conference on Technologies and Innovation*, pages 296–307. Springer, 2017.

[20] JM Hilkert. Inertially stabilized platform technology concepts and principles. *IEEE control systems magazine*, 28(1):26–46, 2008.

[21] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). *arXiv preprint arXiv:1704.04861*, 126, 2017.

[22] Z. Hurak and M. Rezac. Image-Based Pointing and Tracking for Inertially Stabilized Airborne Camera Platform. *IEEE Transactions on Control Systems Technology*, 20(5):1146–1159, September 2012.

[23] Zdeněk Hurák and Martin Řezáč. Combined line-of-sight inertial stabilization and visual tracking: application to an airborne camera platform. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 8458–8463. IEEE, 2009.

[24] Thinh Huynh, Minh-Thien Tran, Dong-Hun Lee, Soumayya Chakir, and Young-Bok Kim. A Study on Vision-Based Backstepping Control for a Target Tracking System. *Actuators*, 10(5):105, May 2021.

[25] InvenSense Inc. Mpu-6050 6-axis gyroscope and accelerometer datasheet, 2023.

[26] Asharul Islam Khan and Salim Al-Habsi. Machine Learning in Computer Vision. *Procedia Computer Science*, 167:1444–1451, January 2020.

[27] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018.

[28] Haitao Li and Jiangkun Yu. Anti-disturbance control based on cascade eso and sliding mode control for gimbal system of double gimbal cmg. *IEEE Access*, 8:5644–5654, 2019.

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[30] Xuancen Liu, Yueneng Yang, Chenxiang Ma, Jie Li, and Shifeng Zhang. Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System. *Applied Sciences*, 10(15):5064, July 2020.

[31] Amir Naderolasli and Mohammad Tabatabaei. Two-axis gimbal system stabilization using adaptive feedback linearization. *Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering)*, 13(3):355–368, 2020.

[32] VPS Naidu and B Indhu. Gimballed camera control for on-point target tracking [j]. *American Research Journal of Electronics and Communication*, 1(1):1–9, 2015.

[33] Paraskevi Nousi, Emmanouil Patsiouras, Anastasios Tefas, and Ioannis Pitas. Convolutional neural networks for visual information analysis with limited computing resources. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 321–325. IEEE, 2018.

[34] Tower Pro. Mg996r high torque servo datasheet, 2023.

[35] Li Qingqing, Jussi Taipalmaa, Jorge Pena Queralta, Tuan Nguyen Gia, Moncef Gabbouj, Hannu Tenhunen, Jenni Raitoharju, and Tomi Westerlund. Towards active vision with uavs in marine search and rescue: Analyzing human detection at variable altitudes. In *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 65–70. IEEE, 2020.

[36] R J Rajesh and P Kavitha. Camera gimbal stabilization using conventional PID controller and evolutionary algorithms. In *2015 International Conference on Computer, Communication and Control (IC4)*, pages 1–6, Indore, September 2015. IEEE.

[37] Shah Mihir Rajesh, S Bhargava, and K Sivanathan. Mission planning and waypoint navigtion of a micro quadcopter by selectable gps co-ordinates. *Int. J. of Advanced Research in Computer Science and Software Eng*, 143, 2014.

[38] Anitha Ramachandran and Arun Kumar Sangaiah. A review on object detection in unmanned aerial vehicle surveillance. *International Journal of Cognitive Computing in Engineering*, 2:215–228, 2021.

[39] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[40] D. J. Regner, J. D. Salazar, P. V. Buschinelli, M. Machado, D. Oliveira, J. M. Santos, C. A. Marinho, and T. C. Pinto. OBJECT TRACKING CONTROL USING A GIMBAL MECHANISM. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XLIII-B1-2021, pages 189–196, June 2021. ISSN: 2194-9034 Journal Abbreviation: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[42] Murat Sahin. Gimbal Axes Control with PID Controllers. *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji*, 11, February 2023.

[43] T. Sai, B. Aditya, A. Reddy, and Dr Srinivasulu. Real Time Object Detection Using Raspberry Pi. *International Journal for Research in Applied Science and Engineering Technology*, 11:834–838, January 2023.

[44] Iqbal H Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160, 2021.

[45] Anton Saveliev and Igor Lebedev. Method of autonomous survey of power lines using a multi-rotor uav. In *Frontiers in Robotics and Electromechanics*, pages 359–376. Springer, 2023.

[46] NXP Semiconductors. Pca9685 16-channel, 12-bit pwm fm+ i2c-bus led controller, 2015.

[47] S Senthil Kumar and G Anitha. Fuzzy logic-based self-tuning pid controllers using parameters adaptive method for stabilization of a two-axis seeker gimbal. *IETE Journal of Research*, 69(9):6188–6197, 2023.

[48] AKM Asfaque Shakil Shaon, Subrata Bhowmik, Bikash Kumar Bhawmick, Piyas Das, and Nipu Kumar Das. Design and implementation of a self-balancing robot. In *Proc. Int. Conf. on Mechanical Eng. And Renewable Energy*, 2017.

[49] Bharath Singh and Jaison Be. Gyro-Stabilized Camera Control in drones for Military Applications. *IOP Conference Series: Materials Science and Engineering*, 1012:012017, January 2021.

[50] Z Soleimanitaleb and MA Keyvanrad. Single object tracking: A survey of methods, datasets, and evaluation metrics. arxiv 2022. *arXiv preprint arXiv:2201.13066*.

[51] Jingxuan Sun, Boyang Li, Yifan Jiang, and Chih-yung Wen. A camera-based target detection and positioning uav system for search and rescue (sar) purposes. *Sensors*, 16(11):1778, 2016.

[52] Roshan Umate. Construction and design of a camera stabilizer based on a gyroscope and accelerometer sensor. *Journal of Advanced Research in Dynamical and Control Systems*, 11:3013–3018, July 2020.

[53] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1328–1338, 2019.

[54] Yiming Wang, Qian Huang, Chuanxu Jiang, Jiwen Liu, Mingzhou Shang, and Zhuang Miao. Video stabilization: A comprehensive survey. *Neurocomputing*, 516:205–230, January 2023.

[55] Hewa Majeed Zangana, Ayaz Khalid Mohammed, Zina Bibo Sallow, and Firas Mahmood Mustafa. Exploring image representation and color spaces in computer vision: A comprehensive review. *The Indonesian Journal of Computer Science*, 13(3), 2024.

[56] Fu Zhang. Simultaneous self-calibration of nonorthogonality and nonlinearity of cost-effective multiaxis inertially stabilized gimbal systems. *IEEE Robotics and Automation Letters*, 3(1):132–139, 2017.

[57] Yi Zhang, Zhuohui Huang, and Jie Jiang. Emerging photoelectric devices for neuromorphic vision applications: principles, developments, and outlooks. *Science and Technology of Advanced Materials*, 24, March 2023.

[58] Ying Zhao and Arjuna Flenner. Deep models, machine learning, and artificial intelligence applications in national and international security. *AI Magazine*, 40(1):35–36, 2019.

[59] Andrew Zulu and Samuel John. A Review of Control Algorithms for Autonomous Quadrotors. *Open Journal of Applied Sciences*, 04(14):547–556, 2014.

Appendix

Design of an Inertially Stabilized Airborne Camera for UAV-based Active Object Tracking

Startup Project under Ministerial Decree N$^{\text{o}}$ 1275

People’s Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Saad Dahleb Blida1
Institute of Aeronautics and Space Studies
Aeronautical Construction Department

Manuscript

Towards achieving the startup diploma under Ministerial Decree N° 1275.

Option: **Avionics**

Design of an Inertially Stabilized Airborne Camera for UAV-based Active Object Tracking

Startup project presented as part of the ”1275” ministerial decree, provided by Institute of Aeronautics and Space Studies.

Project code: 05_15_202

Submitted by:

- **Ms. KADDOURI Hind Yasmine**

Supervised by:

- **Dr. KRIM Mohamed**
- **Mr. MEHAYA Mohammed Mehdi**

Economic Partner:

- **Mr. BELARBI Adel**

Trainer:

- **Dr. LEBSIR Abdelkader**

Jury Members:

Dr. DILMI Smain	President
Dr. AZMEDROUB Boussad	Examiner
Dr. HADJI Ahmed	Incubator Representative

Defended on September 18th, 2024

Introduction

The development and innovation in the field of unmanned aerial vehicles (UAVs) have expanded the possibilities for civilian applications, notably in areas such as surveillance, reconnaissance, and search and rescue operations. The startup project aims to bridge the gap between the growing demand for advanced UAV technologies and the limited availability of locally produced solutions in Algeria, particularly focusing on imaging systems for UAV-based object tracking and stabilization.

The primary focus of this venture is the design of an inertially stabilized airborne camera system integrated with UAVs for active object tracking. The system utilizes a 3-axis gimbal to stabilize camera movement in real-time, ensuring that images remain clear and the object of interest stays centered despite UAV motion or environmental disturbances. This technology is critical for both military and civilian use, especially in search and rescue, infrastructure inspection, and other applications requiring precise, real-time imaging.

Given Algeria's strategic interest in developing local UAV technologies, this startup not only fills a technological gap but also contributes to economic development by providing an affordable, locally manufactured solution for both government and private entities. The combination of advanced engineering and local production ensures that this initiative will foster innovation, reduce dependency on imports, and open new opportunities for technological development within the country.

Jury Members

Jury Members	Name and Surname	Grade	Institution	Signature
President	Dr. DILMI Smain	MCA	Univ. Blida 1	
Supervisor	Dr. KRIM Mohamed	MCB	Univ. Blida 1	
Co-Supervisor	Mr. MEHAYA Mohammed Mehdi	Computer Science Engineer		
Examiner	Dr. AZMEDROUB Boussad	MCB	Univ. Blida 1	
Incubator/Entrepreneurship Center Representative	Dr. Hadji Ahmed	MCB	Univ. Blida 1	
Economic Partner Representative	Mr. BELARBI Adel	Director	TAER R&D Sidi Bel Abbes	
Head of the Technology and Innovation Support Center				

Infromation Card

About the supervision team of the work group

Supervision Team	Speciality	Faculty	Institution
Supervisor: **Dr. Krim Mohamed** **Mr. Mehaya Mohammed Mehdi**	**Telecommunication** **Computer Science**	**IASS**	**Univ. Blida 1**
Economic Partner: **Mr. Adel Belarbi**	**Structure**	**TAER R&D**	**TAER, Sidi Bel Abbes**

About the working group project team

The Work Team	Speciality	Faculty	Institution
Ms. Hind Yasmine KADDOURI	**Avionics**	**IASS**	**Univ. Blida 1**

Business Model Canvas - BMC

	Project Holder	Supervisors: Dr. Krim Mohammed Mr. Mehaya Mohammed Mehdi	Project code
Business Model Canvas - BMC	**Hind Yasmine KADDOURI**	**Economic Partner:** **Mr. Adel Belarbi**	**05_15_202**

Startup Pro

Design of an Inertially Stabilized Airborne Camera for UAV-based Active Object Tracking

Key Partners	Key Activities	Value Proposition	Customer Relationships	Customer Segments
– Local Drone Manufacturers. – Universities and Research Centers. – Government Agencies. – Suppliers of Electronic Components. – Software Development Firms.	– Design, manufacture, and continuous improvement of the cameras. – Marketing and Sales. – Technical Support and Training. – Customization and Client Projects. – Continuous Research and Innovation.	– Stabilized drone camera for accurate surveillance and detection. – Locally produced. – Versatile solution for various missions. – Educational tool that is useful for research centers and universities. – Ability to customize the camera system based on the client's requirements.	– Provide excellent customer service before and after the sale. – Technical support for integration with UAV systems. – After-sales maintenance service. – Long-term partnership by offering customization and continuous updates to meet their evolving needs.	– Local Drone Manufacturers. – Authorities responsible for surveillance missions (wildfires, search and rescue...). – Research Laboratories and Universities. – Environmental and Safety Organizations. – Private Security and Industrial Sectors.

Key Resources

- Technical Personnel.
- Supply Chain for Electronic Components.
- R&D team.

Channels

- Direct sales to drone manufacturers.
- Online Marketing and Website.
- Collaborations with local authorities and research centers.
- Presence at professional trade shows and exhibitions.

Cost Structure

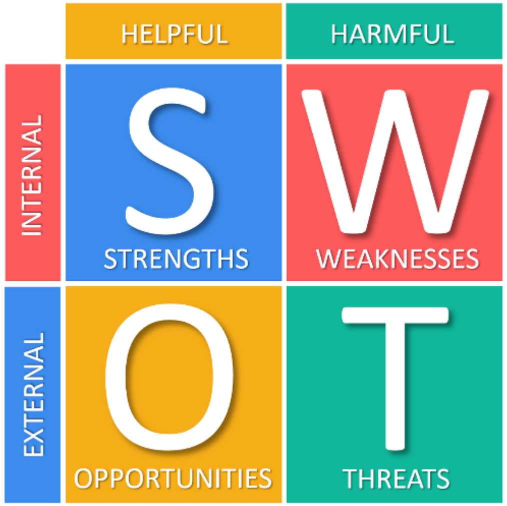
- Component Costs.
- Costs associated with the assembly, production, and quality testing of the camera systems, including labor costs.
- Investment in ongoing R&D activities.
- Marketing and Sales to promote the product.
- Ongoing costs for providing customer support, technical troubleshooting, and system maintenance.
- Costs related to customizing the camera system to meet specific client needs, including software updates or hardware modifications.

Revenue Streams

- Sales of stabilized cameras.
- Maintenance and technical support contracts.
- Leasing or Rental for short-term usage.
- Training Services.
- Customization Services.
- R&D Collaborations.
-

SWOT Analysis

Strengths, Weaknesses, Opportunities, and Threats



- **Suitable for a variety of missions, including aerial mapping, wildfire detection, and infrastructure monitoring.**
- **Integration with drones allows access to remote or dangerous areas without risking human operators.**
- **Opening a wide market.**

- **Further improvements are necessary.**
- **The system currently relies on a Raspberry Pi, limiting its ability to use more advanced object detection models in real-time.**
- **High development costs.**

- **The demand for commercial and industrial drones with specialized imaging needs.**
- **Developing a local industry in Algeria for UAVs and surveillance systems represents a major market opportunity.**
- **Using your product as a pedagogical tool for research centers could stimulate R&D and academic collaborations.**

- **Competing with more advanced systems from large international companies could limit market share.**
- **Laws and regulations regarding drone use.**
- **Local production could be affected by the availability of components or fluctuations in imported parts’ prices.**

La fiche technique de projet startup 1275

البطاقة التقنية للمشروع

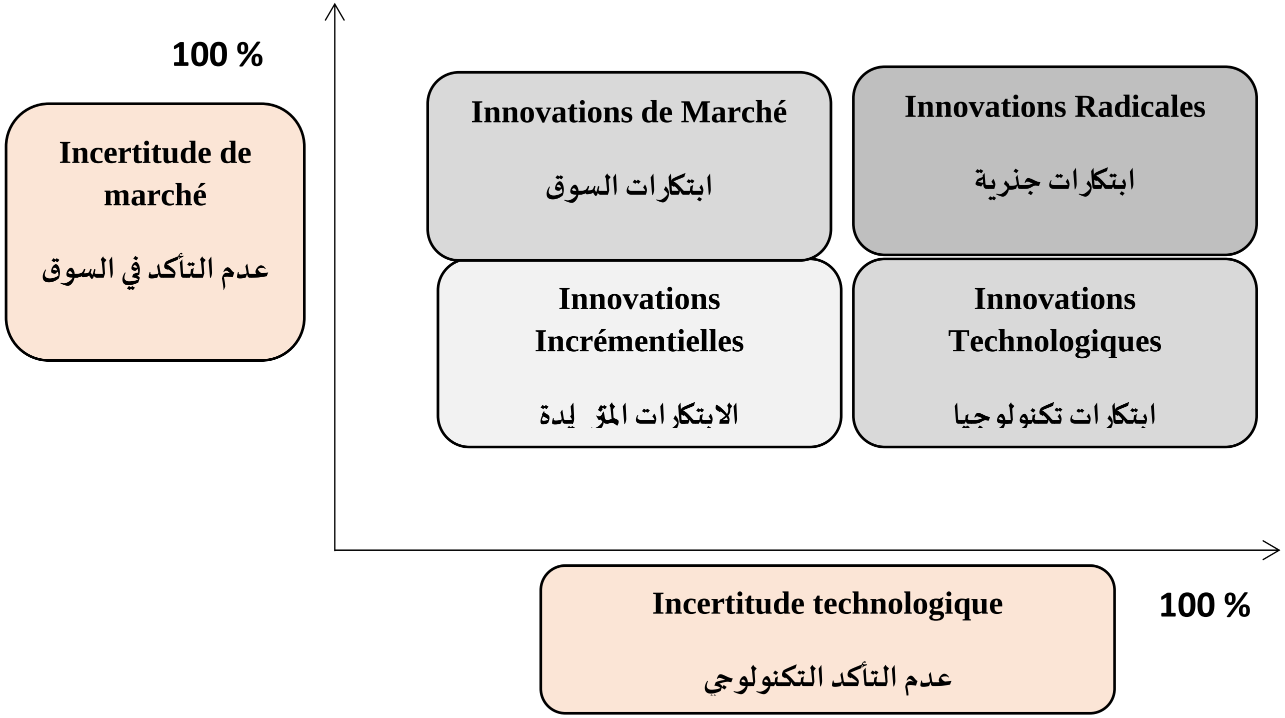
La fiche technique de projet

Carte d’information	
Hind Yasmine KADDOURI	الاسم و اللقب Your first and last Name Votre prénom et nom
Aero Vision	الاسم التجاري للمشروع Intitulé de votre projet Title of your Project
Legal in case of authorization	الصفة القانونية للمشروع Votre statut juridique Your legal status
A professional number will be available.	رقم الهاتق Votre numéro de téléphone Your phone number
kaddourihindyasmine@gmail.com	البريد الالكتروني Votre adresse e-mail Your email address
Mascara	مقر مزاولة النشاط (الولاية- البلدية) Votre ville ou commune d'activité Your city or municipality of activity

طبيعة المشروع (طبيعة الابتكار)

La nature de projet

Sale of services	المنتوج ذو طابع إنتاجي أو خدماتي Vente de marchandises ou de services Sale of goods or services



100 %

Incertitude de marché

عدم التأكد في السوق

Innovations de Marché

ابتكارات السوق

Innovations Radicales

ابتكارات جذرية

Innovations Incrémentielles

الابتكار ات المتزايدة

Innovations Technologiques

ابتكار ات تكنو لو جيا

Incertitude technologique

عدم التأكد التكنولوجي

100 %

القيمة المقترحة أو العرض المقدّم

تحديد المشكل الذي يواجهه الزبون ***Déterminer le problème rencontré par le client***	
The goal is to emerge market in Algeria by providing locally-produced gimbals with AI-based applications that easily integrate with drones and meet the specific needs of local manufacturers, surveillance authorities, research centers, and other stakeholders.	**ما هي المشكلة التي تريد حلها؟**
Lack of UAV project manufacturers	**ما هي البيانات المتوفرة لديك التي تدل على وجود المشكلة المحددة؟**
/	**ما هي المشاريع الأخرى التي استهدفت نفس المشكلة والتي جرى تنفيذها؟**
The goals of my project are to develop locally-produced gimbal systems with AI-based applications that can easily integrate with drones to meet the needs of local manufacturers, surveillance authorities, and research centers. The expected outcomes include: • Improving efficiency in aerial surveillance operations, such as wildfire detection and search-and-rescue missions in difficult-to-access areas. • Boosting local production by providing advanced AI-driven technological solutions to support the local economy.	**ماهي أهداف مشروعك و/أو نتائجه المتوقعة؟**

• Reducing dependence on imported systems, offering cost-effective local alternatives. • Encouraging innovation and scientific research by providing advanced tools for academic institutions and research centers. • Generating job apportunities.	

La proposition de valeur ou l'offre faite

القيمة المقترحة وفق المعايير التالية

La valeur proposée selon les critères suivants

القيمة المقترحة وفق المعايير التالية	
/	القيمة بالتخصيص
/	القيمة بالسعر
/	القيمة بالتصميم
/	القيمة بالأداء العالي
/	القيمة بالخدمة الشاملة
/	قيم أخرى

Annexe

محتوى مخطط العملي للمشروع

Business Model Canvas – *BMC*

1.الشركاء الأساسيون Key Partners

طبيعة الشراكة	معلومات حول الشركاء	الشركاء
-	TAER R&D	**الشريك الأول**
-	/	**الشريك الثاني**
-	/	**الشريك الثالث**
-	/	**الشريك الرابع**
-	/	**الشريك الخامس**

قم بكتابة قائمة **الشركاء الرئيسيون** لمشروعك بالتفصيل مع ذكر الإسم، الهاتف، العنوان...إلخ

2. شرائح العملاء أو الزبائن Customer Segments

Géographique الجغرافية	Démographique (B2C)	Démographique (B2B)	Psychographique العوامل النفسية و الشخصية	Comportemental السلوكيات
Continent عالمية	Age العمر: كبار وصغار	Secteur إقتصادي	Classe sociale لطبقة الاجتماعية	Usage استخدام
Pays دولة:الجزائر	Sexe الجنس: إناث وذكور.	Nombre d'employés عدد العمال في القطاع	Niveau de vie المستوى المعيشي	Loyauté الوفاء
Région الجهة الوسط	Revenus annuels متوسط الدخل: 40 000 دينار جزائري	Maturité de l'entreprise نضج المؤسسة	Valeurs القيم	Intérêt اهتمام
Département الولاية البليدة	Etat matrimonial الحالة الاجتماعية: عازب- متزوج	Situation financière الحالة المالية للمؤسسة	Personnalité الشخصية	Passion الهواية و شغف
Ville الدائرة او البلدية البليدة	Niveau d'étude المستوى الدراسي: لايهم	Détention/ actionnariat الملكية/المساهمة	Convictions المعتقدات	Sensibilité حساسيات
Quartier الحي Oulad-Yaich	Profession المهنة: لاتهم	Valorisation/ capitalisation boursière التقييم / القيمة السوقية	Présence digitale et sur les réseaux sociaux استعمال التكنولوجيا في التواصل	Habitude de consommation عادة الاستهلاك
Climat المناخ كل المناخات	Culture الثقافة:	Business model نموذج الأعمال	Centres d'intérêts مراكز الاهتمام	Mode de paiement طرق الدفع
يمكن أن تكون دولياً.	Religion الدين: كل الديانات	Secteur servi القطاع الذي يخدمه		Connaissance المعرفة
	Langue اللغة: كل اللغات	Technologie utilisée التكنولوجيا المستعملة		Nature de la demande طبيعة الطلب
		Format du produit ou packaging شكل المنتج أو التعبئة والتغليف		Fréquence d'achat عدد مرات الطلب على السلعة

3. قنوات التوزيع Channels

Channels قنوات التوزيع	
المبيعات المباشرة	/
تجار الجملة	/
الموزعون	/
توزيع التجزئة	/

4. العلاقة مع العملاء Customer Relationship

كيف تدير علاقاتك مع العملاء؟	To manage our relationships with customers, we adopt an approach based on building long-term partnerships founded on trust and ongoing collaboration. This approach includes: • Providing continuous technical support: we assist customers with integrating our products into their systems, offering guidance and training to ensure effective use of the product. • Maintenance and after-sales services: we ensure regular maintenance contracts are available to resolve any technical issues that may arise after the sale, enhancing customer satisfaction and maintaining the relationship. • Continuous communication: We maintain regular contact with customers to stay updated on their new needs and developments, helping to adapt future products to better meet their requirements.

- Flexibility in customization: we offer the ability to customize products to meet each customer’s specific needs, providing tailored solutions that add value to the product. - Listening to feedback: We actively gather customer feedback and evaluations to improve products and services, ensuring a great experience that meets their expectations. .	
- Social Media - Web Site	**ماهية أهم البرامج التي ستعتمد عليها في ادارة العلاقة مع الزبون** Microsoft Dynamics Monday CRM Zoho CRM الخ................................

1. هيكل التكاليف structure Costs

هيكل التكاليف structure Costs	En milliers DZD
تكاليف التعريف بالمنتج أو المؤسسةFrais d’établissement	50000 دج
تكاليف الحصول على العدادات (الماء- الكهرباء ....)(eaux-gaz-….) Frais d’ouverture de compteurs	50000 دينار جزائري
تكاليف (التكوين- برامج الاعلام الالي المختصة)Logiciels, formations	100000 دج
تكاليف براءة الاختراع و الحماية الصناعية و التجارية Dépôt marque, brevet, modèle	20 000 دينار جزائري
تكاليف الحصول على تكنولوجيا او ترخيص استعمالهاDroits d’entrée	0 دج
شراء الأصول التجارية أو الأسهم Achat fonds de commerce ou parts	0 دج
الحق في الإيجار Droit au bail	30 000×12= 360 000 دج
وديعة أو وديعة تأمينCaution ou dépôt de garantie	00000 دج
رسوم إيداع الملفاتFrais de dossier	50 000 دينار جزائري
تكاليف الموثق-المحامي-. Frais de notaire ou d’avocat	0 دج
تكاليف التعريف بالعلامة و تكاليف قنوات الاتصالEnseigne et éléments de communication	20 000 دينار جزائري
شراء العقاراتAchat immobilier	0 دج
الأعمال والتحسينات الاماكنTravaux et aménagements	250 000 دج
الآلات- المركبات- الاجهزةMatériel	سيارة 2500 000 دج
تجهيزات المكتبMatériel de bureau	2500000 (Ordinateurs , chaises, bureaux …..) دينار جزائري
تكاليف التخزينStock de matières et produits	0 دج
التدفق النقدي(الصندوق) الذي تحتاجه في بداية المشروع.Trésorerie de départ	0 دج
GLOBAL المجموع =	5 650 000 دج

2. نفقاتك أو التكاليف الثابتة الخاصة بمشروعك

En milliers DZD						نفقاتك أو التكاليف الثابتة الخاصة بمشروعك
PREVISION			REALISATION			
N	N +1	N+2	N	N -1	N -2	
Location 100 000 دينار جزائري						Assurancesالتأمينات
24 000دينار جزائري = 2000×12						Téléphone, internetالهاتف و الانترنت
0						Autres abonnementsاشتراكات أخرى
1000×356= 356 000دج						Carburant, transportsالوقود و تكاليف النقل
200 000 دينار جزائري						Frais de déplacement et hébergement تكاليف التنقل و المبيت
10 000×4= 40 000 دج						Eau, électricité, gazفواتير الماء – الكهرباء- الغاز
0 دينار جزائري						Mutuelleالتعاضدية الاجتماعية
10 000 دينار جزائري						Fournitures diversesلوازم متنوعة
0 دينار جزاري						Entretien matériel et vêtementsصيانة المعدات والملابس
50 000 دينار جزائري						Nettoyage des locauxتنظيف المباني
50 000 دينار جزائري						Budget publicité et communication ميزانية الإعلان والاتصالات
830 000 دج						GLOBAL المجموع =

3. مصادر الإيرادات *Revenue Stream*

***Revenue Stream* مصادر الإيرادات**	
900 000 دج	**Apport personnel ou familialالمساهمة الشخصية أو العائلية**
0 دج	**Apports en nature (en valeur)التبرعات العينية**
0 دج	**Prêt n°1 (nom de la banque)قرض رقم 1 اسم البنك**
0 دج	**قرض رقم 2 اسم البنك Prêt n°2 (nom de la banque)**
0 دج	**قرض رقم3 اسم البنك Prêt n°3 (nom de la banque)**
0 دج	**منحة 1 Subvention n°1 (libellé)**
0 دج	**منحة 2 Subvention n°2 (libellé)**
0 دج	**تمويل آخر Autre financement (libellé)**
900 000 دج	= المجموع GLOBAL

4. رقم الأعمال

بيع المنتج في السنة الأولى
Votre Chiffre d'affaires de la Première Année

بيع المنتج في السنة الأولى	أيام العمل في الشهر	عدد الفئة المستهدفة	سعر الخدمة	رقم الأعمال
(N) 1Mois الشهر	30	20	40 000	800.000
(N+1) 2Mois الشهر	30	20	40 000	800.000
(N+2) 3Mois الشهر	30	20	40 000	800.000
(N+3) 4Mois الشهر	30	20	40 000	800.000
(N+4) 5Mois الشهر	30	20	40 000	800.000
(N+5) 6Mois الشهر	30	20	40 000	800.000
(N+6) 7Mois الشهر	30	20	40 000	800.000
(N+7) 8Mois الشهر	30	20	40 000	800.000
(N+8) 9Mois الشهر	30	20	40 000	800.000
(N+9) 10Mois الشهر	30	20	40 000	800.000
(N+10) 11Mois الشهر	30	20	40 000	800.000
(N+11) 12Mois الشهر	30	20	40 000	800.000
= المجموع GLOBAL	360	240	480000	9.600.000 دينار جزائري
النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟				% 9.600.0

المجموع = 9.600.000 دينار جزائري.

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟1 % 9.600.000 × 1 ÷ 100=9.600.0 دينار جزائري.

بيع المنتج في السنة الثانية

Votre Chiffre d'affaires de la Deuxième Année

بيع المنتج في السنة الثانية	أيام العمل في الشهر	عدد الفئة المستهدفة	سعر الخدمة	رقم الأعمال
الشهر 1Mois (N)	30	20	60 000	1800000
الشهر 2Mois (N+1)	30	20	60 000	1800000
الشهر 3Mois (N+2)	30	20	60 000	1800000
الشهر 4Mois (N+3)	30	20	60 000	1800000
الشهر 5Mois (N+4)	30	20	60 000	1800000
الشهر 6Mois (N+5)	30	20	60 000	1800000
الشهر 7Mois (N+6)	30	20	60 000	1800000
الشهر 8Mois (N+7)	30	20	60 000	1800000
الشهر 9Mois (N+8)	30	20	60 000	1800000
الشهر 10Mois (N+9)	30	20	60 000	1800000
الشهر 11Mois (N+10)	30	20	60 000	1800000
الشهر 12Mois (N+11)	30	20	60 000	1800000
GLOBAL المجموع =	360	240	480000	21600000
النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟				% 11.760.000

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثانية؟10 % = 11.760.000 %

2.160.000 =100 ÷ 10 ×2 1.600.000

2.160.000+ 9.600.000=11.760.000

بيع المنتج في السنة الثالثة

Votre Chiffre d'affaires de la Troisième Année

بيع المنتج في السنة الثانية	أيام العمل في الشهر	عدد الفئة المستهدفة	سعر الخدمة	رقم الأعمال
الشهر 1Mois (N)	30	40	800000	3.200.000
الشهر 2Mois (N+1)	30	40	800000	3.200.000
الشهر 3Mois (N+2)	30	40	800000	3.200.000
الشهر 4Mois (N+3)	30	40	800000	3.200.000
الشهر 5Mois (N+4)	30	40	800000	3.200.000
الشهر 6Mois (N+5)	30	40	800000	3.200.000
الشهر 7Mois (N+6)	30	40	800000	3.200.000
الشهر 8Mois (N+7)	30	40	800000	3.200.000
الشهر 9Mois (N+8)	30	40	800000	3.200.000
الشهر 10Mois (N+9)	30	40	800000	3.200.000
الشهر 11Mois (N+10)	30	40	800000	3.200.000
الشهر 12Mois (N+11)	30	40	800000	3.200.000
GLOBAL المجموع =	360	480	960000	38400000
النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الأولى؟				% 29280000

المجموع 38.400.000دينار جزائري.

النسبة المئوية للزيادة في حجم الأعمال بين كل شهر لسنة الثالثة ؟20 %   المجموع=   29280000   %

5. تطور حجم رقم الأعمال في السنة

الزيادة في حجم الأعمال	النسبة المئوية %
بين السنة 1 والسنة 2	*10%*
بين السنة 2 والسنة 3	*10%*
= المجموع GLOBAL	*20%*

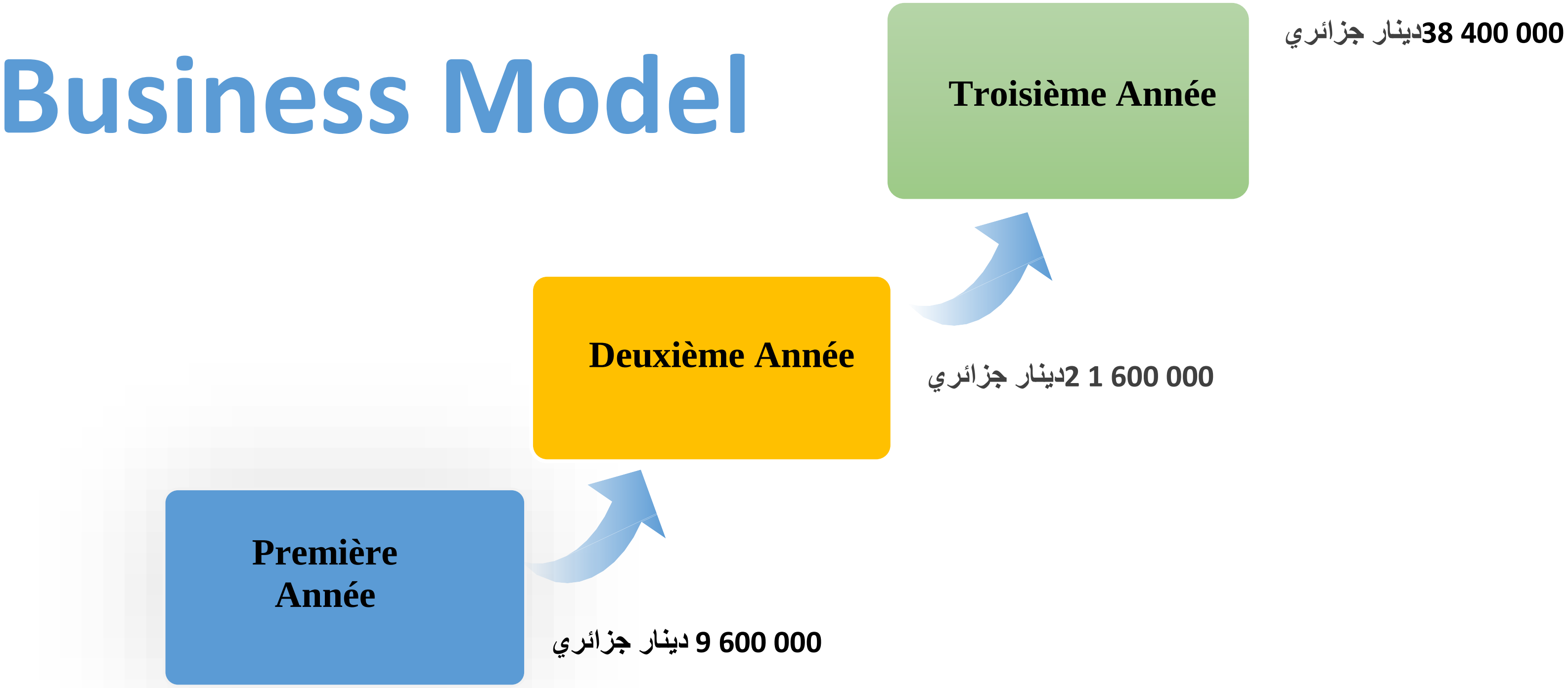
6. حاجتك لرأس المال العامل

Durée moyenneمتوسط مدة	les joues الأيام
الاعتمادات الممنوحة للعملاء بالأيام Des crédits accordés aux clients en jours	0دج
ديون الموردين بالأيام Des dettes fournisseurs en jours	0دج
= المجموع GLOBAL	0دج

7. رواتب الموظفين و مسؤولين الشركة

صافي أجور المسؤولين Rémunération nette dirigeant	رواتب الموظفين Salaires employés	رواتب الموظفين و مسؤولين الشركة
العام الأول 0 دينار، العام الثاني 45000 دينار، العام الثالث 60000 دينار.	0 دج	CHIFFRE D'AFFAIRES
105000دج	0 دج	GLOBAL المجموع =

Business Model



Troisième Année

38 400 000دينار جزائري

Deuxième Année

2 1 600 000دينار جزائري

Première Année

9 600 000 دينار جزائري

تطور حجم الأعمال في السنوات الثلاث