الجمهورية الجزائرية الدمقراطية الشعبة
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

Mohamed Khider University of Biskra
Faculty of Science and Technology
Electrical Engineering Department
Ref : .....

UNIVERSITY OF BISKRA

جامعة محمد خيضر بسكرة
كاية العلوم و التكنولوجيا
قسم الهندسة الكهربائية
المرجع: .....

Thesis

Submitted for assessment with a view of obtaining

the Degree of LMD doctorate (3rd Cycle)

in Automatic

Presented by:
TELLI KHALED

On: 04 March 2024

Conception Et Contrôle D'un Quadrotor

Membres of the jury :

President :	Pr. BAHRI Mebarek	Prof	University of Biskra
Supervisor :	Pr. BOUMEHRAZ Mohamed	Prof	University of Biskra
Examiner :	Pr. CHELIHI Abdelghani	Prof	University of Constantine
Examiner :	Dr. TOUBA Mostefa Mohamed	MCA	University of Biskra

2024

الجمهورية الجزائرية الدمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research

Mohamed Khider University of Biskra
Faculty of Science and Technology
Electrical Engineering Department
Ref : .....

UNIVERSITY OF BISKRA

جامعة محمد خيضر بسكرة
كاية العلوم و التكنولوجيا
قسم الهندسة الكهربائية
المرجع: .....

Thesis

Submitted for assessment with a view of obtaining

the Degree of LMD doctorate (3rd Cycle)

in Automatic

Presented by:

TELLI KHALED

On: 04 March 2024

Design and Control of a Quadrotor

Membres of the jury :

President :	Pr. BAHRI Mebarek	Prof	University of Biskra
Supervisor :	Pr. BOUMEHRAZ Mohamed	Prof	University of Biskra
Examiner :	Pr. CHELIHI Abdelghani	Prof	University of Constantine
Examiner :	Dr. TOUBA Mostefa Mohamed	MCA	University of Biskra

2024

DEDICATION

I am dedicating this thesis to the people who taught me the value of hard work.

First and foremost, I thank my parents, **Mohamed** and **Moummi Rebiha**, for their affection, advice, sacrifice, and encouragement.

I hope you will find in my work my deep respect and appreciation for you.

To my wife and my children: Abderrahman, Ines, Ahmed, and my little boy Noufel.

To my brothers and sisters: Sara, Mahdi, Walid, Hichem, Dounai, Oussama, and Hatem.

For their invaluable assistance and encouragement at times when I needed support and encouragement.

All my family, friends, and colleagues. All my teachers at the Department of Electrical Engineering. Prof. Boumehraz Mohamed and, in particular, Prof. KRAA Okba, for their inspiration and insights.

Finally, I dedicate it to all those I love and appreciate.

ACKNOWLEDGEMENTS

I am thankful and appreciative to my supervisor, Pr. **Mohamed Boumehraz**, for his assistance and direction during this work.

My gratitude and special thanks are also extended to Pr. **Okba Kraa** for his encouragement and advice.

My appreciation also goes to the members of the jury who have evaluated my work, Pr. BAHRI Mebarek, Pr. TOUBA Mostefa Mohamed, CHELIHI Abdelghani.

I also want to say thank you from the bottom of my heart to all the professors who have helped me and looked at my work with a critical eye.

Abstract

In recent years, Unmanned Aerial Vehicles (UAVs) have attracted considerable interest and found numerous applications in various fields, including civil defence, commercial delivery, recreation, and agriculture. The present study aims to provide a comprehensive exploration of systematic quadrotor development, substantiated by an extensive experimental investigation. The research commences by conducting a detailed analysis of the methodological aspects involved in quadrotor development. It elucidates our engineering approach, which prioritizes robust design principles and lays the foundation for an open-source quadrotor research platform. Subsequently, this thesis reviews quadrotor sizing techniques, highlighting the substantial influence of size on the selection of optimal components. A data-driven clustering technique is exhaustively discussed and implemented in the quadrotor components selection process. In addition, this work addresses aspects of sensor calibration and filtering. It encompasses the calibration procedures for the Inertial Measurement Unit (IMU) and Compass sensor, along with the implementation of sensor signals filtering techniques, with a specific emphasis on the complementary filter technique. Furthermore, the study delves into vibration analysis, utilizing Fast Fourier Transform (FFT) techniques to achieve motor-propeller balancing and noise reduction. Quadrotor dynamics identification is meticulously investigated through white, grey and black-box approaches, leading to a comprehensive understanding of the system dynamics. Finally, the thesis explores control design by leveraging the identified model to design an efficient controller. Overall, this research provides insights into the design and control of quadrotor systems, offering a comprehensive methodology with the aim to be a valuable resource for future research and development endeavours in this rapidly evolving field.

Keywords: UAV, Quadrotor, Design, Methodology, Quadrotor Sizing, Vibration Analysis, Control, Sensor Calibration, Filter, Identification, ARX Model, ANFIS, PID controller, LQR controller.

الملخص

في السنوات الأخيرة، أثارت مركبات الطيران بدون طيار (UAVs) اهتمامًا كبيرًا ووجدت تطبيقات عديدة في مجموعة متنوعة من المجالات، بما في ذلك الدفاع المدني، والتوصيل التجاري، والترفيه، والزراعة. يهدف هذا البحث إلى تقديم استكشاف شامل لتطوير الكوادروتور النظامي، مدعومًا بتحقيق تجريبي واسع النطاق. يبدأ البحث من خلال إجراء تحليل مفصل للجوانب المنهجية المشمولة في تطوير الكوادروتور. يوضح نهجنا الهندسي، الذي يعطي الأولوية لمبادئ التصميم القوية ويضع الأسس لمنصة بحث كوادروتور مفتوحة المصدر. بعد ذلك، يستعرض هذا البحث تقنيات تحديد حجم الكوادروتور ويسلط الضوء على التأثير الكبير للحجم على اختيار المكونات الأمثل. تُناقش بشكل شامل تقنية تجميع البيانات وتُنفذ في عملية اختيار مكونات الكوادروتور. بالإضافة إلى ذلك، يتناول هذا العمل جوانب لعمليات معايرة الأجهزة الاستشعارية وتصفية الإشارات. تشمل إجراءات المعايرة لوحدة القياس الممانعة (IMU) ومستشعر البوصلة، إلى جانب تنفيذ تقنيات تصفية إشارات الأجهزة الاستشعار، مع التركيز الخاص على تقنية المرشح المكمل. وعلاوة على ذلك، يتناول البحث تحليل الاهتزاز باستخدام تقنيات تحويل فوريه سريعة (FFT) لتحقيق توازن المحرك والمروحة وتقليل الضوضاء. يتم التحقيق بدقة في تحديد ديناميات الكوادروتور من خلال نهج الصندوق الأبيض والرمادي والأسود، مما يؤدي إلى فهم شامل لديناميات النظام. وأخيرًا، يستكشف البحث تصميم التحكم من خلال الاستفادة من النموذج المحدد لتصميم وحدة تحكم فعالة. بشكل عام، يهدف هذا البحث إلى تقديم رؤى حول تصميم وتحكم الأنظمة الكوادروتورية، ويقدم منهجًا شاملاً بهدف أن يكون مصدرًا قيمًا لجهود البحث والتطوير المستقبلية في هذا الميدان المتطور بسرعة.

الكلمات المفتاحية: المركبات الجوية غير المأهولة:، طائرة رباعية، التصميم، المنهجية، تحجيم طائرة رباعية، تحليل الاهتزاز ، التحكم معايرة المستشعر، المرشح، التحديد ، نموذج رجعي تلقائي مع مدخلة خارجية, النظام التمثيلي الضبابي العصبي التكيفي ، متحكم النسبية المشتقة التكاملية, متحكم خطي تربيعي.

Publications in journals

- **Telli, K.**, Boumehraz, M. (2023). Black-Box System Identification for Low-Cost Quadrotor Attitude at Hovering. *in Electrotehnica, Electronica, Automatica (EEA)*, 4(70), 88-97. DOI:10.46904/eea.22.70.4.1108009.

- **Telli, K**., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., & Mansoor, W. (2023). A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems*, 11(8), 400. https://doi.org/10.3390/systems11080400

Publications in international conferences

- **Telli, K.**, Boumehraz, M., Titaouine, A. (December 2018). Adaptive Sliding Mode Controller and Observer for Altitude and Attitude Control of a Quadrotor. *in Second International Conference on Electrical Engineering ICEEB'2018*, Biskra, Algeria.

- **Telli, K.**, Kraa, O., Boumehraz., M. (March 2023). Quadcopter Control Based on Experimental Dynamic Identification. *in International Conference on Advances in Electrical and Computer Engineering 2023 (ICAECE'2023), TEBESSA Algeria*.

- **Telli, K**., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., & Mansoor, W. (November 2023). Quadrotor Experimental Dynamic Identification with Comprehensive NARX Neural Networks. *6th IEEE International Conference on Signal Processing and Information Security (ICSPIS), Dubai*.

Publications currently undergoing peer review for international conference publications:

- **Telli, K**., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M. (April 2024). Drone-Enabled Connectivity: Advancements and Challenges in B5G / 6G Networks. *The 8th International Conference on Image and Signal Processing and their Applications, ISPA 2024*.

National patent applications currently under review and pending approval:

- **Telli, K.**, KRAA O., M Boumehraz., Universal Test Bench for Multirotors.

- **Telli, K.,** KRAA., Hybrid-Transformable-Strategy, Multi-Drone, Multi-Brain Aircraft.

TABLE OF CONTENTS

LIST OF TABLES

TABLE **Page**

LIST OF FIGURES

Figure **Page**

LIST OF ACRONYMS

AI	Artificial intelligence
AIC	Akaike's Information Criteria
AMR	Anisotropic Magneto-Resistance
ANFIS	Adaptive Network-based Fuzzy Inference System
ARMAX	AutoregRessive Moving Average with eXogenous inputs
ARX	AutoregRessive with eXogenous inputs
BEC	Battery Elimination Circuit
BJ	Box Jenkins
BLDCM	Brushless Direct Current Motor
CCW	Counter Clockwise
CFRP	Carbon Fibre Reinforced Polymer
CKF	Cubature Kalman Filter
COM	Center Of Mass
CW	Clock Wise
DC	Direct Current
DMP	Digital Motion Processor
DOD	Depth Of Discharge
DOF	Degrees Of Freedom
EKF	Extended Kalman Filter
EMF	Electromotive Force
EMST	Electric Multirotor Sizing Tool
ESC	Electronic Speed Controller
FET	Field-Effect Transistor
GPL	General Public License
GPS	Global Positioning System
HOSMC	High Order Sliding Mode Control
HPF	Hight Pass Filter
I2C	Inter-Integrated Circuit
IAGA	International Association of Geomagnetism and Aeronomy
IC	Integrated Circuit
IDE	Integrated Development Environment
IGRF	International Geomagnetic Reference Field
IMU	Inertial Measurement Unit
IR	Infrared
IV	Instrumental Variable
KF	Kalman Filter
Li-ion	Lithium-Ion
LiPo	Lithium-Polymer
LPF	Low Pass Filter
LQR	Linear Quadratic Regulator

LS	Least Squares
LSM	Least Squares Method
MCU	Micro Controller Unit
MF	Membership Functions
MOI	Moment Of Inertia
NiMH	Nickel–Metal Hydride
PDF	Particle Filter
PE	Prediction Error
PID	Proportional Integral Derivative
PRBS	Pseudo Random Binary Sequence
PSD	Power Spectral Density
PWM	Pulse Width Modulation
RMSE	Root Mean Square Error
ROS	Robot Operating System
RW	Rotor Wing
SCL	Serial Clock
SDA	Serial Data
SEM	Sensor Error Models
SMC	Sliding Mode Control
SOC	State Of Charge
TS	Takagi Sugeno
UAV	Unmanned Aerial Vehicle
UKF	Unscented Kalman Filter
UWB	Ultra-Wide Band
VFD	Variable Frequency Drive
VSFs	Variable Structure Filters
VTOL	Vertical Take-Off and Landing

Overview

Nowadays, Unmanned Aerial Vehicles (UAVs) have become more common in a wide range of applications, such as military and civilian ones. Used fo aerial surveillance, exploration, and inspection in complicated and risky areas, they provide significant advantages and higher confidence in mission accomplishment. These features motivate the research community and industrial community to invest in this field and continue UAV expansion. Currently, the UAV segment is a continually growing and active section of the broader aircraft industry. UAVs are far more suited for many civil purposes than manned and fixed-wing aircraft. The most common UAV applications are related to defence, but they can also be used for a variety of civil purposes such as border and road traffic monitoring, security, infrastructure inspection, and so on. A majority of these applications require flights and hovering at lower altitudes with Vertical Take-Off and Landing (VTOL) options. The smaller UAVs are better suited to these tasks than fixed-wing UAVs. Moreover, UAVs are most commonly used by academic communities and government research teams as experimental platforms to evaluate their navigation, guidance, and control algorithms. UAVs have been classified in many ways depending on characteristics like payload and size, rotors and wings, altitude, risk-based use, and energy use [1-3]. After selecting the UAV type to be studied, [4] this paper demonstrates how the size of the quadrotor frame affects the manoeuvrability as well as how the propeller size and rigidity should be considered in order to get the desired velocity. In order to reduce vibration, many researchers are compelled to perform static and dynamic balancing [5]. For manned and unmanned aircraft, some literatures pays attention to UAV identification [6–8]. A successful model is determined by how effectively the input and sensor output signals are gathered, and this is only guaranteed if and only if the sensors are calibrated properly, because uncalibrated sensors lead to unstable control and mismatched model representation [9–11]. The control issue has captured the attention of many researchers in the control and robot communities due to the fact that it brings interesting control challenges and a great opportunity to test and develop a new control strategy methodology. Many review and survey articles outline quadrotor control [12–14]. It is imperative to conduct a comprehensive study of the quadrotor system in order to achieve robust results. The focus of many research endeavors in the field has been directed towards individual aspects of the system, leading to the neglect of other critical components. A holistic approach, which considers all aspects of the quadrotor system,

is essential in ensuring the validity and reliability of the research findings. In this study, we focus on the development of a comprehensive engineering pathway for the design and control of quadrotor. To this end, we have designed and built a reliable experimental platform in the form of a quadrotor, utilizing optimization and sizing techniques to ensure its reliability. The proposed platform incorporates a novel strategy for reducing vibrations using Fast Fourier Transform (FFT) in both static and dynamic modes, which results in improved stability and control of the UAV. Additionally, we have calibrated all quadrotor sensors using a reliable calibration method to mitigate the impact of sensor errors and noise on the accuracy of measurements. The dynamic model of the quadrotor was identified using various identification strategies and the results were analysed using techniques such as root mean square (RMS), residuals, and uncertainty analysis. Finally, we have developed a practical control strategy that takes into account real-world effects such as sensor noise, disturbances, uncertainties, and variations in parameters and models, providing a robust solution for real-world quadrotor applications. In conclusion, this study represents a systematic and comprehensive approach to the design and control of UAVs, offering valuable insights for the advancement of UAV technology.

Thesis Purpose

The research landscape in the field of UAVs engineering lacks comprehensive and integrated references that address the technical and robust design of UAVs in a holistic manner. The current body of literature is fragmented, focusing primarily on individual aspects such as control, identification, or calibration. In order to achieve a fully integrated project that encompasses all facets of UAV design and engineering, it is crucial to understand the most essential elements involved. This thesis aims to fill this gap by proposing a dependable engineering method for the design and control of UAVs, with a specific focus on the "quadrotor" type. The methodology begins with the definition of objectives, constraints, and requirements. This is followed by sizing and optimization analysis, as well as morphology assessments. The optimal design is then determined through simulations or experimental testing. Vibration reduction strategies are implemented, and sensors are calibrated to improve readings and minimize their impact on system stability. After coding and testing, the UAV is

prepared for real-world applications, including control, fault detection, vision systems, identification, sensor calibration, tracking, and other potential developments.

Contributions

This doctoral thesis endeavours to explore and advance robust engineering methodologies in the field of UAV design and control, with a specific focus on the "quadrotor" type as a fully academic open-source platform. The main contributions of this work are:

- Development of a comprehensive engineering pathway for the design and control of UAVs. This pathway employs a systematic approach to the design and control of UAVs, ensuring that the relevant aspects of the system are considered and incorporated into the final solution.

- Construction of a reliable experimental platform (quadrotor) using dependable engineering techniques. The quadrotor platform has been designed and built to be highly reliable and capable of providing accurate data for the analysis and validation of the proposed control strategies.

- Implementation of an efficient strategy for reducing vibrations using FFT in both static and dynamic modes. This strategy takes advantage of the unique properties of FFT to identify and mitigate sources of vibration in the system, resulting in improved stability of the UAV.

In conclusion, this thesis represents a contribution to the field of UAV design and control, and provides a foundation for future work in this area. The proposed engineering pathway, experimental platform, and control strategies represent a significant advancement in the state of the art and have the potential to impact a wide range of applications in the field of UAVs.

This thesis organised as following:

Chapter 1 is quadrotor overview and design methodology, explores various aspects related to methodology of quadrotor design and UAV classification. It covers topics such as quadrotor definition, airframe shape and configuration, stress and deformation analysis, and the morphology of quadrotors. The chapter also discusses the vibration characteristics of quadrotor airframes. Additionally, the chapter emphasizes the importance of a solid engineering process in the design methodology of quadrotors. It highlights the need for defining goals, constraints, and other parameters before proceeding with sizing and functional optimization. The design methodology is divided into many areas, including methods that enhance specific characteristics through input processing and database utilization, and the others based on existing designs and design trends.

Chapter 2 is quadrotor hardware and software overview, explores in-depth quadrotor the used components and software architecture. It provides an overview of the quadrotor system architecture benchmark and discusses key components such as the flight controller and computer. The chapter also covers fundamental sensors used in UAV quadrotor systems and the propulsion chain, including actuators like propellers, motors, ESCs. The communication protocols and the role of batteries are examined, along with an exploration of open-source projects for hardware and software autopilot systems. Finally, the chapter introduces the quadrotor testbench, which serves as a platform for evaluating quadrotor system performance.

Chapter 3 is sensor calibration and filtering, delves into the topic of sensor calibration and filtering. It begins with an overview of sensor calibration, addressing sensor errors and noise. The focus is on deterministic errors, particularly in the context of Inertial Measurement Unit (IMU) sensors. The chapter defines the IMU sensor error model and explores the calibration process for IMU sensors. Additionally, the chapter discusses magnetometer errors, their model, and the calibration process to mitigate these errors. It introduces the barometer sensor and Global Positioning System (GPS) as essential sensors in quadrotor systems. The chapter then shifts its focus to sensor filtering, providing an overview of techniques such as the Kalman Filter (KF) and Extended Kalman Filter (EKF). However, the primary emphasis is on complementary filtering. Complementary filtering is explored as a method to combine multiple sensor inputs and reduce noise, enhancing the overall accuracy of the sensor measurements. Furthermore, the chapter introduces vibration analyses in the time and frequency domains. This analysis is crucial for understanding the impact of vibrations on sensor measurements and developing methods to mitigate their effects.

Chapter 4 is dynamic modelling identification and control, focuses on dynamic modeling, identification, and control of quadrotors. It begins with an overview of quadrotor dynamic modeling, covering approaches such as Euler-Lagrange, Newton-Euler, and Quaternion methods. The chapter highlights the extraction of the quadrotor dynamic model from these approaches. Next, the chapter explores dynamic identification techniques for quadrotors. It discusses the design considerations for system identification and introduces the use of adaptive neuro-fuzzy Inference system for identification purposes. The chapter then delves into quadrotor control, starting with an overview of quadrotor control architecture. It discusses linear control techniques, including PID controllers and state feedback control and Linear Quadratic Regulator, the tuning of weighting matrices Q and R for LQR control is also addressed. Additionally, the chapter explores sliding mode control and its application in quadrotor altitude control. The design of a sliding mode observer is also discussed.

Chapter 5 is experimental results and discussion; this section presents the experimental results and discussions regarding the sizing and frame design of the quadrotor. The stress and deformation analysis, various predictions, and geometric

parameters of the quadrotor are discussed in detail. The hardware and software selections and the general wiring of the quadrotor are also addressed. The outcomes of the vibration-reduction strategies are presented and discussed. The sensor calibration and filtering results are also included. Subsequently, the test bench and the results of the quadrotor dynamics identification, utilizing both white-box and black-box approaches as well as intelligent methods, are presented. The comparison between the model generated from the identification process and the real-world response of the quadrotor under different inputs is performed. The final part of this thesis focuses on the results of the control of the quadrotor using various control strategies such as PID, state feedback control, LQR, and tuning based on the identified model using Simulink. The results and performance of these strategies are presented and analysed.

The conclusion highlights key findings in quadrotor design, calibration, and control, emphasizing the vital role of robust engineering processes, precise sensor calibration, and effective control strategies in elevating quadrotor performance.

CHAPITRE 1: QUADROTOR OVERVIEW AND DESIGN METHODOLOGY

CHAPTER 1

QUADROTOR OVERVIEW AND DESIGN METHODOLOGY

Contents

CHAPITRE 1: QUADROTOR OVERVIEW AND DESIGN METHODOLOGY

1.1 Introduction

The field of UAV has gained considerable momentum in recent years, with a focus on the analysis and optimization of quadrotor systems. The design methodology of these aerial systems involves a rigorous examination of their physical characteristics, including airframe shape and configuration, quadrotor sizing, and optimization methodology. This critical analysis plays a crucial role in determining the overall morphology of the quadrotor, which ultimately affects its performance in various operational scenarios. A comprehensive understanding of these design elements is essential in ensuring that quadrotors meet the demanding requirements of various applications, such as aerial photography, package delivery, and surveillance. With the continued growth of the UAV industry, the importance of quadrotor analysis and design methodology will only continue to increase, providing opportunities for further research and development in this exciting field.

1.2 UAV Classification

An unmanned aerial vehicle, popularly known as a "drone" is an aircraft that is remotely piloted or fully autonomous, equipped with central processing and perception technologies and, in some cases, special devices [15-18]. UAVs are divided into a variety of categories according to the flying principle, mission, weight, propulsion, control, altitude range, configuration, purpose, launch method, payload, autonomy level, size, endurance, and range etc.

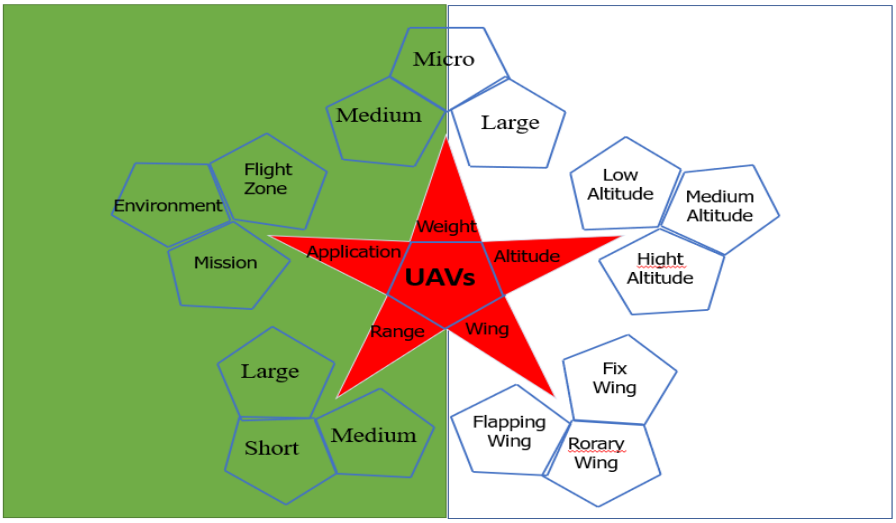


Figure 1.1: UAV Popular Classification

CHAPITRE 1: QUADROTOR OVERVIEW AND DESIGN METHODOLOGY

Numerous attempts have been made to classify UAVs based on a variety of features [1-3] [19-21]. Figure 1.1 illustrate the UAV popular classification.

While the most commonly used classification of UAVs is described below, it is important to note that this classification is subject to change and may vary depending on specific contexts and advancements in technology.

1.2.1 Weight-Base

The UAVs are divided into a variety of classes according to their weight as follows:

Nano UAV < 250 gm
250gm > **Micro UAV** < 2 kg DJI Spark, DJI Mavic, Parrot Bebop2
2 kg > **Small UAV** < 25 kg DJI Matrice600, DJI Inspire2, Airborne Vanguard
25 kg > **Medium UAV** < 150 kg AAI Shadow 200, Scorpion 3 Hoverbike
Large UAV > 150 kg Griff 300, Ehang 216, Boeing X-45A UCAV

1.2.2 Altitude and Range-Based

The UAVs split into a variety of class according to their Altitude (vertical) and Range (horizontal) as follow:

Handheld Altitude < 600m and Range < 2 km
Close Altitude < 1500m and Range <10 km
NATO Altitude < 3000m and Range <50 km
Tactical Altitude < 5500m and Range <160 km
MALE Altitude < 9100m and Range <200 km (Medium Altitude)
Hypersonic Altitude < 15200m and Range >200 km (high Altitude)

1.2.3 Application-Based

The UAVs divided into a variety of class according to their applications as follow:

- Personal-applications such as hobbies, photography, racing, agriculture etc.
- Education-applications such as control developments or validation, path planning, SLAM, networks, vision etc.
- Commercial applications such as surveillance, distribution, and infrastructure etc.

- Government-applications such as rescue operation, road-surveillance, agriculture.
- Military-applications such as security-surveillance, rescue operation, spying and battle.

The Figure 1.2 provides a detailed overview of UAV application classes.

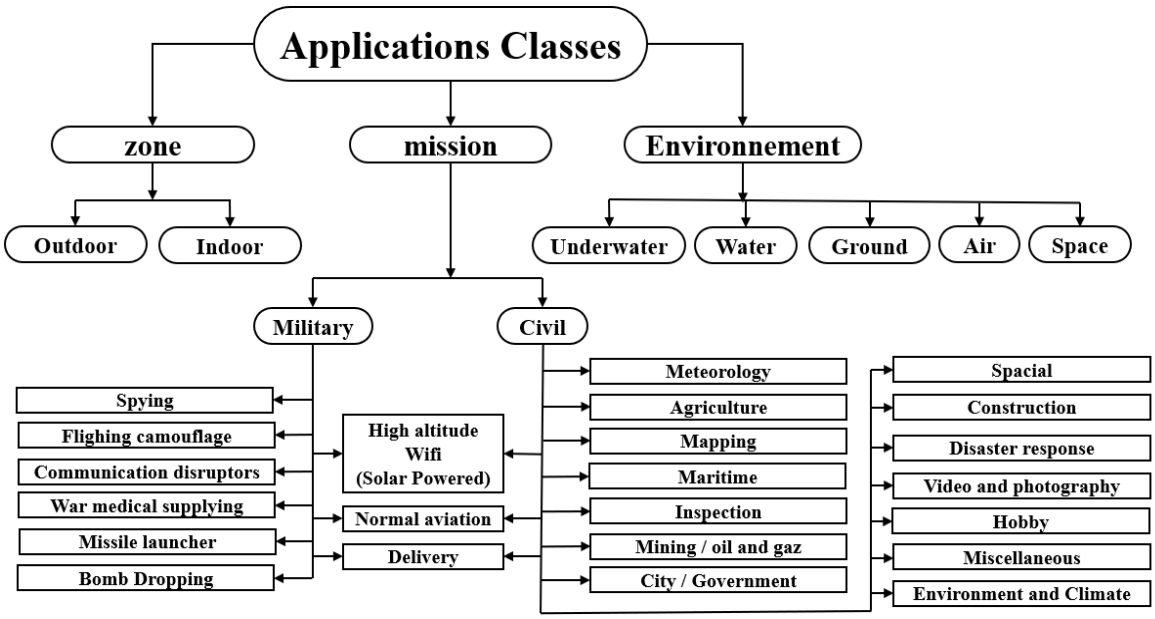


Figure 1.2: Classification of UAVs' Application [21]

The widely recognized UAV classification is shown in Tables 1.1 and 1.2. The UAV rotor-wing class, also known as vertical take-off and landing rotorcraft, is used in missions where hovering is required. Helicopters and quadrotors are the most known rotary-wing aircraft, where their rotors provide thrust and motion.

Table 1.1: UAVs Heaviness and Motorized Classifications

UAV					
Lighter Than the air		**Heavies than the air**			
Non-motorized	Motorized	Non-motorized	**Motorized**		
Balloon, Captive	Airship	Glider, Kite	Aeroplan	**Rotorcraft -as Quadrotor**	Ornithopter

Table 1.2 UAVs Wing Classifications

UAV							
Fixed-Wing **Aeroplane With fixed-wing**	Rotor-Wing (RW)						Flapping-Wing (Ornithopter)
	Octocopter (Eight-RW)	Hexacopter (Six-RW)	**Quadrotor (Four-RW)**	Tricopter (Three-RW)	Bicopter (Two-RW)	Helicopter (One-RW)	

Figure 1.3 depicts both the Rotor-wing UAV class and the hybrid one.

Figure 1.3: Rotor-wing/ hydride UAV class

Based on the classification in Tables 1.1–1.2, the quadrotor was chosen as one of the rotor-wing UAV classes categorized under the motorized heavier-than-air category. It is a micro-UAV, handheld, and designed for educational purposes.

1.3 Quadrotor Overview and Design Methodology

1.3.1 Quadrotor Definition

1.3.1.1 Quadrotor Overview

The quadrotor or quadrotor in figure 1.4 is a type of multirotor or rotor wing, which is a lifting-rotor or wing that rotates in order to create aerodynamic lift. It features four rotor-propellers, two rotating clockwise (CW) and two rotating counter clockwise (CCW) at its extremities. The propellers generate torque and lift near its rotation centres. Variations in propeller velocity produce changes in altitude and attitude. The pitch and roll are controlled by adjusting the total thrust center, whereas the yaw is controlled by adjusting the net torque. The quadrotor is controlled with the aid of sensors and a microcontroller, which analyses the signals emitted by the sensors, which provide complete sensitivity to the quadrotor's movements and surroundings. A microcontroller-programmed algorithm collects and analyses data in order to maintain the quadrotor's stability and perform the required tasks. Certain tasks necessitate the installation of sensors or specialized devices, such as a camera arm, a carrying hook, etc., in the quadrotor. Other applications require an onboard computer to accomplish complex tasks. In some particular environments, there are needs for engineering topology where the quadrotor can change its morphology (shape or size) to adapt to the environment and the task at hand, In some situations, it requires an adaptive algorithm, particularly in the presence of anomalies or when there are changes in the natural environment.

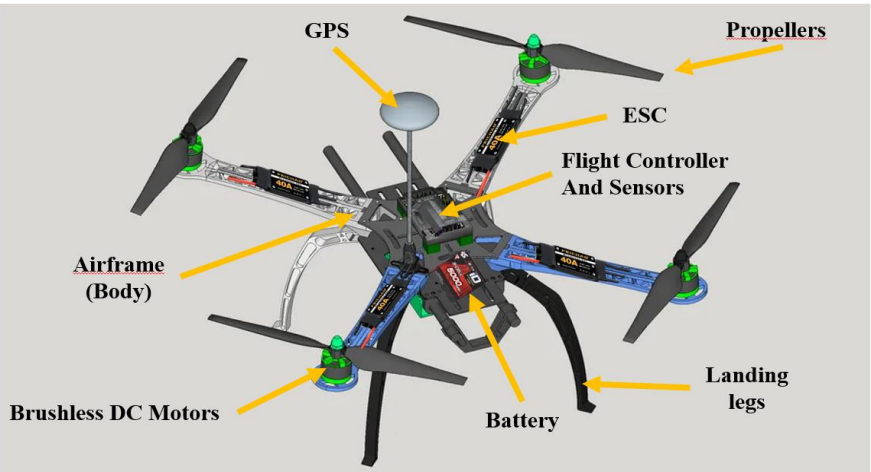


Figure 1.4: Quadrotor with its Main Components

1.3.1.2 Quadrotor Airframe Shape and Configuration

The quadrotor's structure and configuration are the first features that catch the eye. Most quadrotor frames are made of carbon fiber or plastic. It has numerous advantages, such as strong tensile strength, higher hardness, and low weight due to its high carbon content. Quadrotors are available in a variety of sizes and shapes depending on the application, further increasing their adaptability. Some quadrotors have a fixed shape, while others have an adaptive shape that changes while flying to accommodate a particular task or environment, which makes the platform more complex and decreases its controllability.

The airframe of the quadrotor is meticulously engineered to eliminate any related structural natural frequencies within the rotor-propeller velocity range. The frame material and the geometric characteristics are the main factors that affect the hardness of the quadrotor arms. Some studies [22-24] investigate a few various types of beam structures under the restrictions of weight and size. For micro quadrotors, five distinct types of beam structures are created and examined, rectangular and circular hollow T shapes, and N shapes. In the same research, it was discovered that a closed-shape beam produces higher natural frequencies than the competition, outperforming them. This work [5] examines the modeling and vibration analysis of the quadrotor body frame to check the range of failure frequencies by modifying the boundary condition. The study discovered that the failure frequencies for fixed-body-base range from 1197.8Hz to 1299.8Hz and 1084.5Hz to 4411.9Hz for fixed propeller-arm using fixed propeller arm and body base boundary condition. The quadrotor frame configuration can be divided into two categories:

A) Ring configuration: The arms are not crossed in the ring configuration shown in Figure 1.5.a, but they form a square shape, and each corner is occupied by a propeller-motor. It is more rigid and heavier than the cross one, which may limit manoeuvrability in some ways, but it is useful for reducing vibration.

B) Cross configuration: This kind of quadrotor has a two-arm configuration in a cross configuration where four ensembles of motor-propeller are fixed at the ends of each arm and symmetrically distributed. The other parts, e.g., the autopilot and sensors, are placed in the middle area of the quadrotor, where the IMU sensor is

preferred to be placed at the gravity center. Further, the cross-configuration is divided into two famous kinds, "+" and "x" configurations, whereas the "X" configuration shown in Figure 1.5.c is much more popular than the "+" Figure 1.5.b, because the roll and pitch control authority is 30% larger in the "+" configuration [25].

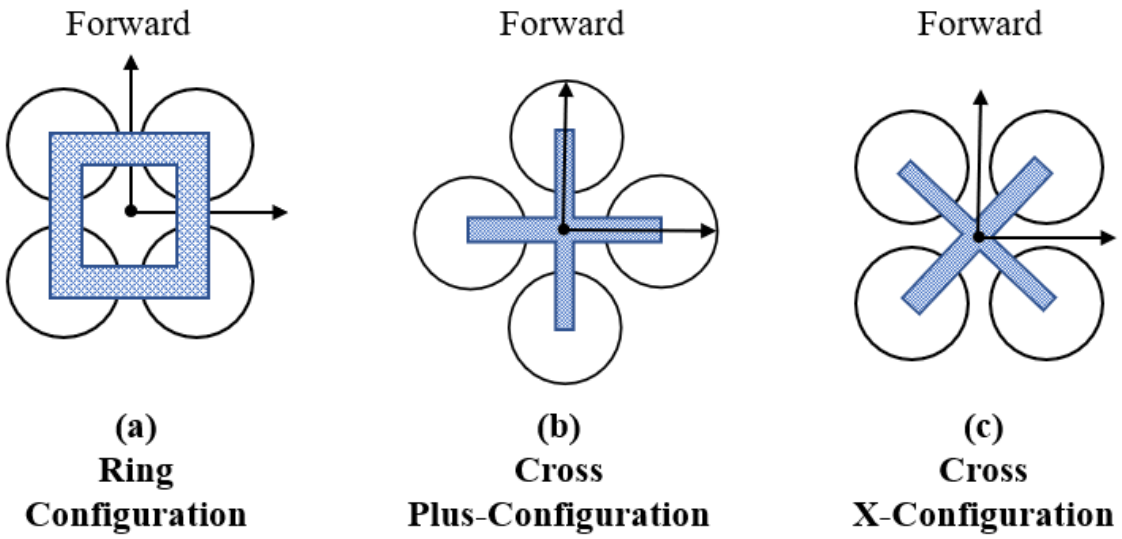


Figure 1.5: Quadrotor configurations ring and cross (X and +)

1.3.1.3 Quadrotor Airframe Stress and Deformation Analysis

The frame of a quadrotor is subjected to various types of stresses such as bending, torsion, and tension. To ensure the structural integrity of the frame, it is essential to perform a stress analysis. The objective of the stress analysis is to determine the distribution of stresses throughout the frame and to ensure that they do not exceed the maximum allowable stress.

A) Stress analysis: Determine the load cases that the frame will be subjected to includes the weight of the quadrotor, the lift generated by the propellers, and any external loads such as wind and turbulence. Once the load cases have been defined, the next step is to determine the deformation of the frame under each load case. This is typically done using finite element analysis (FEA), ANSYS software, which allows us to model the frame as a collection of small elements and to calculate the

deformation of each element. Once the deformation of the frame has been calculated, the next step is to determine the stresses in each element. This is typically done by calculating the strain in each element, and then using Hooke's law to determine the stress in each element. Finally, it is essential to validate the results of the stress analysis by comparing the calculated stresses with the maximum allowable stress for the material used in the frame. If the calculated stresses exceed the maximum allowable stress, the design of the frame may need to be modified to reduce the stress levels.

B) Deformation Analysis: In addition to stress analysis, it is also important to perform a deformation analysis to determine the shape of the frame under load. The objective of the deformation analysis is to ensure that the frame retains its intended shape and does not deform excessively under load. The deformation analysis is typically performed using the same FEA, ANSYS software as used in the stress analysis. The first step in performing the deformation analysis is to determine the load cases that the frame will be subjected to, as described in the stress analysis section. Once the load cases have been defined, the next step is to calculate the deformation of the frame under each load case. Once the deformation of the frame has been calculated, it is essential to validate the results by comparing the calculated deformation with the allowable deformation for the frame. If the calculated deformation exceeds the allowable deformation, the design of the frame may need to be modified to reduce the deformation levels. Performing stress and deformation analysis is an essential step in the design of a quadrotor frame, as it ensures that the frame will be able to withstand the loads that it will be subjected to during flight and that it will retain its intended shape.

1.3.1.4 Mounted of Motor Propeller

In a typical scenario, the motor-propeller is placed at the end of each arm, facing upward. However, in order to increase the quadrotor's carrying capacity without increasing its size, the addition of more motor propellers facing downward becomes necessary. This is known as the co-axis, noticed from the experiment that the space between the co-axial propellers affects the system efficiency, and it is recommended

that the ratio of the distance between the propellers divided by the propeller radius be greater than 1.357 times [26].

The motor-propeller can be mounted either facing downwards or upwards, and each orientation has advantages and disadvantages in Table 1.3. The downwards setting generates push forces while the upward setting produces pull forces.

Table 1.3: Advantages and disadvantages motor-propeller mounted

	Motor-prop upward facing	**Motor-prop downward facing**
Advantages	• Preventing propellers from crashing during the landing. • more area of the camera or hand or suspension object	• Preserving motors from rainfall. • Providing a more accurate barometer measurement and Low interference magnet for GPS and magnetometer if placed upper the quadrotor frame
Disadvantages	• Propeller motors exposed to rain. • The inference magnet may affect the GPS and magnetometer accuracy if it is not well placed. When the barometer is not well placed, the produced air from motor-prop affects the reading.	• No free space for a camera, hand, or suspension object.

The conventional configuration illustrated in Figure 1.6.a, is a motor-propeller with a fixed horizon orientation (zero-degree angle with the planed surface), but when the motor-propeller becomes tiltable as illustrated in Figure 1.6.b, many things will change. This type of design is advantageous since it creates thrust along the three body axes (x, y, and z) rather than the conventional one that effects only the z axis, and there is no need to change the motors' velocity in order to produce pitch and roll orientations. This design offers various advantages, including the elimination of the requirement for additional camera stabilization load since it plays its role.

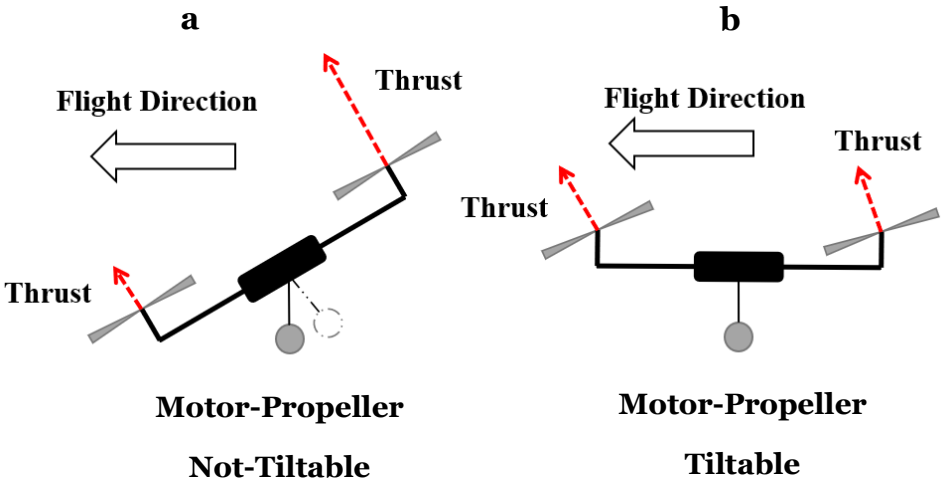


Figure 1.6: Quadrotor with motor-prop tiltable and not tiltable for forward flight

1.3.1.5 Morphology of Quadrotor

Conventional quadrotors are successfully implemented in a variety of applications, including personal, educational, commercial, government, and military ones, under normal conditions; however, these conditions are not always available for certain special tasks and environments. This motivates researchers to develop unconventional quadrotors' ability to freely alter the direction and length of their arms independently, due to their adaptive morphologies, or transformable morphologies, altering its mechanical structure in accordance with the needs, which enhance performance and stability in a variety of unusual scenarios, compared to conventional quadrotors, like flying in congested surroundings, avoiding collision with objects, transporting and grabbing goods, traversing tight and confined places, reducing various airborne damages, optimizing energy usage, and enhancing flying flexibility with more manoeuvrability. The reconfigurable drones divide into eight primary classes presented in Figure 1.7. Whereas Figure 1.8 depicts some reconfigurable drones, some hybrid hovering and terrestrial others with passive and active legs and wheels, some with rolling cage, others foldable and self-arrangeable, and some bio-inspired.

NB: *Our design for a novel convertible multirotor is currently undergoing the patenting process.*

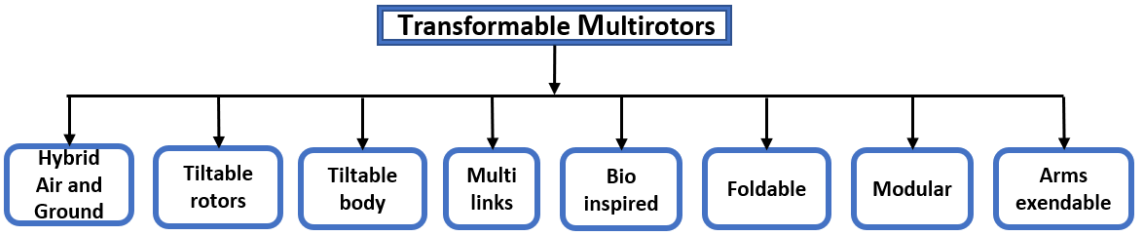


Figure 1.7: Classification of Transformable Multirotors UAVs [46]

Figure 1.8: Example of some multimodal transformable drones

1.3.1.6 Quadrotor Airframe Vibration Analyses

A quadrotor consists of electronic and mechanical parts including four motor-props running at high speeds and provides thrust forces at the quadrotor's arms extremities, resulting in body frame deformation. The body frame and other electronic components will be subjected to vibration as a result of this. Numerous researchers have examined the deformation and vibration of body-frame in an effort to minimize and overcome this phenomenon [42-43]. The von mises stress is utilized to anticipate the materials yielding under diverse loading conditions. It is a scalar value indicating stress at a specific point. To reduce power consumption and maintain safe deformations, Other has modified the quadrotor design of the middle plate [44]. Sandwich architecture with multi-layered for Quadrotor has been investigated to reduce rotator inertia and the total weight [45]. Analysed the

properties of carbon fibre-reinforced polymer (CFRP) used in aerospace and military application which shows high strength-to-weight ratio, high fatigue strength and good corrosion resistance, 1600 Kg/m^3 of density, 0.3 Poisson's ratio and 70000 Mpa of young modulus of elasticity [5]. Later in the same paper, the most susceptible area of failure and its corresponding frequency range were investigated by fixing the propeller arm in the first case and the body base in the second case, where it was shown that the natural frequency of a fixed propeller arm ranges from 1084.5 Hz to 4411.9 Hz and from 1197.8 Hz to 1299.8 Hz for a fixed body base [5].

1.3.2 Design Methodology

This research aims to advance academic understanding of quadrotor design and control, with the goal of serving as a quick and easy reference for other researchers in this field. Due to the costs and risks involved with UAV development, a successful UAV designer requires a solid engineering process. Begins with defining the goals, constraints, etc., and then proceeds through sizing, functional optimization and allotment, morphology analyses, and finally expert intervention. In some scenarios, human expertise may be required to select features that meet the requirement with the fewest unwanted side effects. The simulation facilitates the resolution of the conceptual UAV design. According to the previous engineering step, a quadrotor must be manufactured with the desired hardware components that correspond to the optimal result feature. When an unbalanced motor-propeller rotates at high speeds, it produces vibrations that directly affect the system's stability. Using the IMU accelerometer and some balancing techniques, the intensity of vibration will be diminished; however, all of these steps will only be successful if the sensors are accurately calibrated. Next, the quadrotor will be programmed to achieve minimal stability, with the intention of investing in the final quadrotor model's development and evaluation, control re-development, sensor recalibration, optimization, fault detection, path planning, dentification, etc. Figure 1.9 shows our design flowchart, and practical steps of design are presented to illustrate the application of this technique Figure 1.10.

CHAPITRE 1: QUADROTOR OVERVIEW AND DESIGN METHODOLOGY

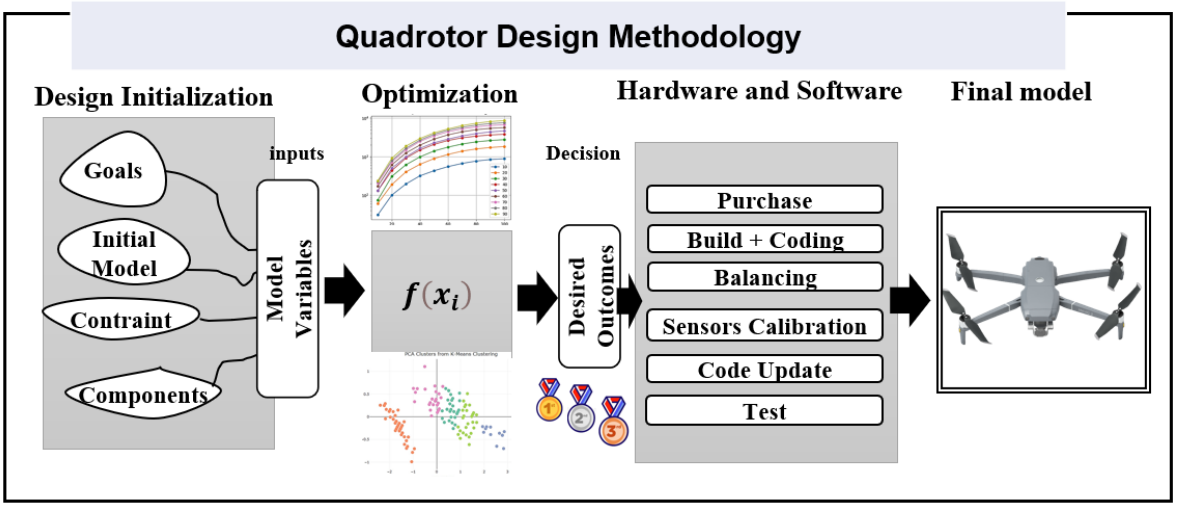


Figure 1.9: Quadrotor Design Methodology Steps

The first phase in a quadrotor designing, we determine the required features and design goals as well as the potential constraints that could impede the construction of the quadrotor, includes specifying the payload capacity, flight time, maximum flight speed, and wind resistance, among other factors of required features. While in targeted goals we specified the goals from this design such as the platform for educational purposes, and support tasks and missions such trajectory following, support the communication, data transfer, and saving, has ability to detect its location, heading, altitude, attitude, support for future additional devices, such as a flight computer, a camera, a hand, etc., these are only a few examples of design goals. The potential constraints must be identified know in order to be taking into consideration before building the quadrotor, including technical, financial, logistical, and regulatory constraints. The outputs of this step of quadrotor design are the initial parameters x_1, x_2... x_{end} for the drift model of the quadrotor. These are just some of the many features that we might consider when sizing a quadrotor, based on these parameters, we can start sizing the quadrotor components, beginning with an initial estimate for quadrotor components, such as the motors, propellers, battery, frame, ESCs, flight controller, and radio control system.

The secode phase in a quadrotor designing, come only after determining the sizing requirements, we can then proceed to purchase the components. It's important to choose components that meet the design specifications and are compatible with

each other. If we purchase components without considering the design requirements, we may end up with an ineffective or inefficient system that doesn't meet our needs. In summary, the recommended process is to first make the quadrotor sizing based on the required features and then buy the components accordingly.

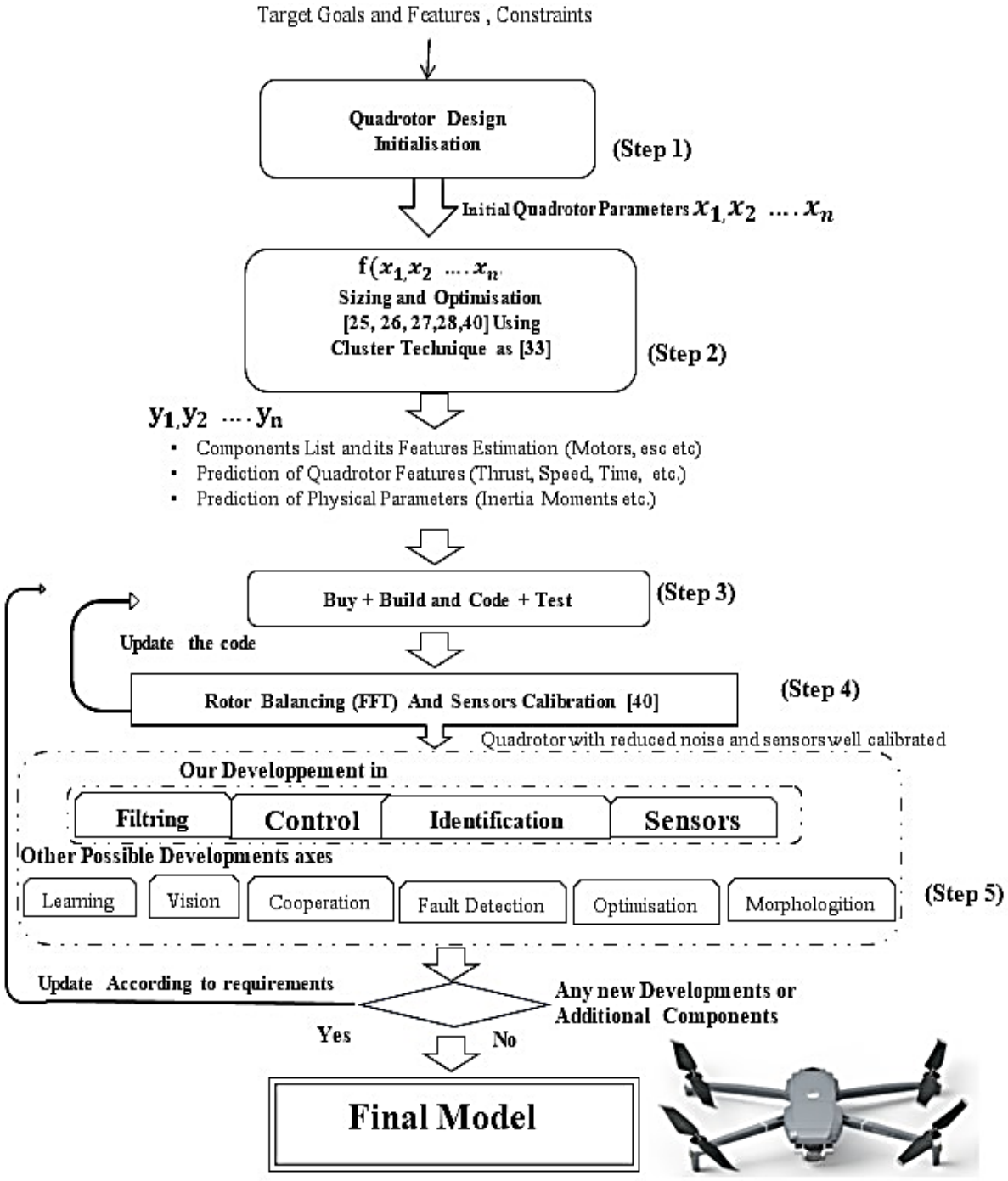


Figure 1.10: Flowchart of our Quadrotor Methodology Design

The initial parameters of the quadrotor model are employed to anticipate a range of suitable components and their associated features, denoted as $y_1, y_2, \ldots$ etc These features can encompass various aspects of quadrotor components, such as motor and propeller dimensions, battery characteristics, as well as properties like moment of inertia, mass, and thrust-to-drag coefficients. They can also relate to quadrotor behaviors, including total thrust, quadrotor speed, flight time, and more.

1.3.2.1 Quadrotor sizing and optimisation methodology

As previously stated, sizing plays a crucial role in achieving the optimal design of drones. In our research within this domain, we can narrow downsizing methods to two main areas.

The **first area** encompasses methods that rely on enhancing specific characteristics by processing inputs and utilizing a database. Through an iterative process, these methods have the potential to improve various aspects such as flight time, payload size, and so on.

The **second area** involves methods that are based on existing designs "*Design Trends*", assuming that these designs have already undergone improvements. By analysing these designs, one can uncover the correlations between the properties of the components and the factors used to create another model, which is constructed based on these revealed relationships. Below, we present a review of the most significant research papers in the field of drone design and sizing.

Online airplane calculators such as eCalc [27] or DriveCalc [28], which utilize databases containing the characteristics of all known airplane components. For many manufacturers, e.g., blade characteristics, motor characteristics, battery characteristics, electronic speed controller (ESC) characteristics, etc., these components are selected individually up to user design, but this is unfortunately unwieldy when sizing multiple models and when the payload scale factor deviates from the market norm. These constraints on optimal design, as well as the obstacles and drawbacks, prompted a number of researchers to develop a wide range of

efficient sizing algorithms and methodologies [29-31]. This permits the configuration to be optimized for various missions and requirements. With regards to optimization, the goal here is to determine the optimal system structure to meet a set of requirements such as hovering time, take-off, maximum rate, etc., through manipulating the input values of user design parameters based on component catalogue data and analytical expressions, it is a multirotor optimization problem that uses an iterative process and requires a well-defined objective function and design variables with well-known constraints. The first trend researchers counts on estimating the relations between different multirotor components by using statistical methods or clustering approaches [32-33] as illustrated in Figure 1.11 to Figure 1.16 , some of them calculate the take-off weight using analytical methods, while others obtain the produced power or propulsion from a simplified propulsion model. These works [34-36] are useful for calculating multi-rotor target weight, but they do not provide useful data for multi-rotor component sizes. Therefore, it cannot be relied upon for the optimal selection of multi-rotor components. In [29], the researcher relied on the parametrical approach and used an optimization tool called the Electric Multirotor Sizing Tool (EMST) that needed a massive component model to calculate the performance characteristics of a multirotor, and it shows its accuracy with many examples in the same literature. This paper [37] discussed an absolutely vital issue; it reverses the solution by finding the optimal propulsion system based on the design specification inputs. The proposed optimization methodology simplifies the multi-rotor sizing problem and decouples it into twelve sub-problems as showed Figure 1.11. Some of these sub-problems optimize the parameters of each component, including the weight, and try to match and locate the corresponding component in the database that contains the real product parameters. Unfortunately, since the available components in trade are limited, this may affect the level of fragmentation and poorly matched optimization results, which will restrict the improvement of the design. The optimization problem proposed in this work [38] has been implemented as a toolbox with sub-functions available online as user services. Sizing methodology based on flight time estimation it well treated in [39] where used an iterative method for determining the time necessary to drain the batteries at stable power demand Figure 1.12.

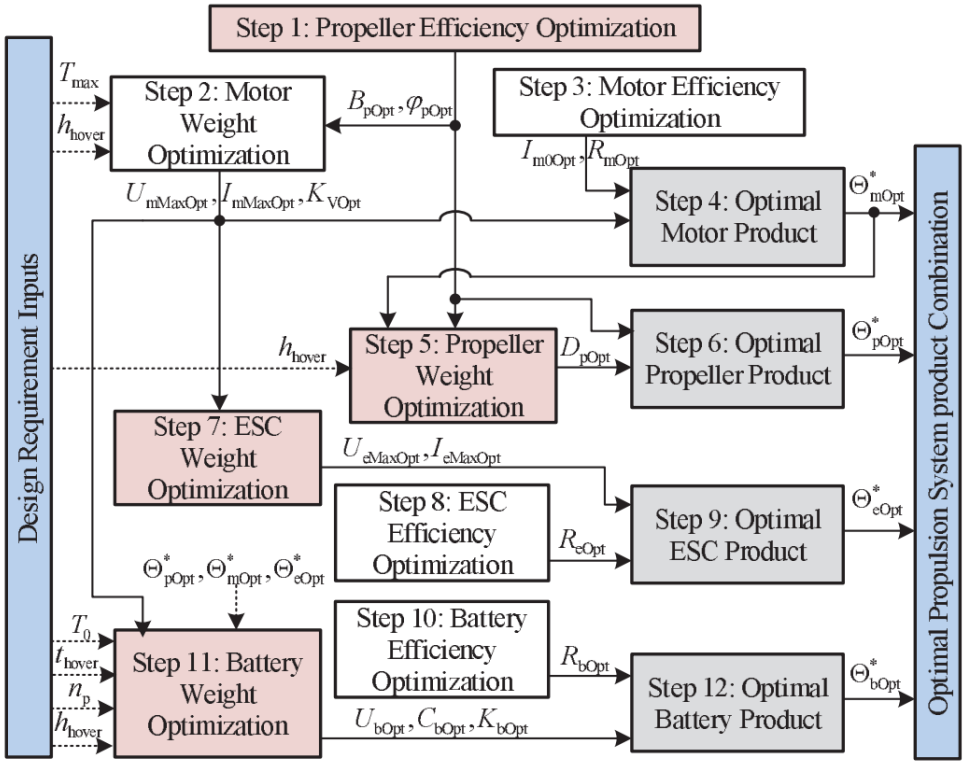


Figure 1.11: Procedures of the propulsion system optimization problem [34]

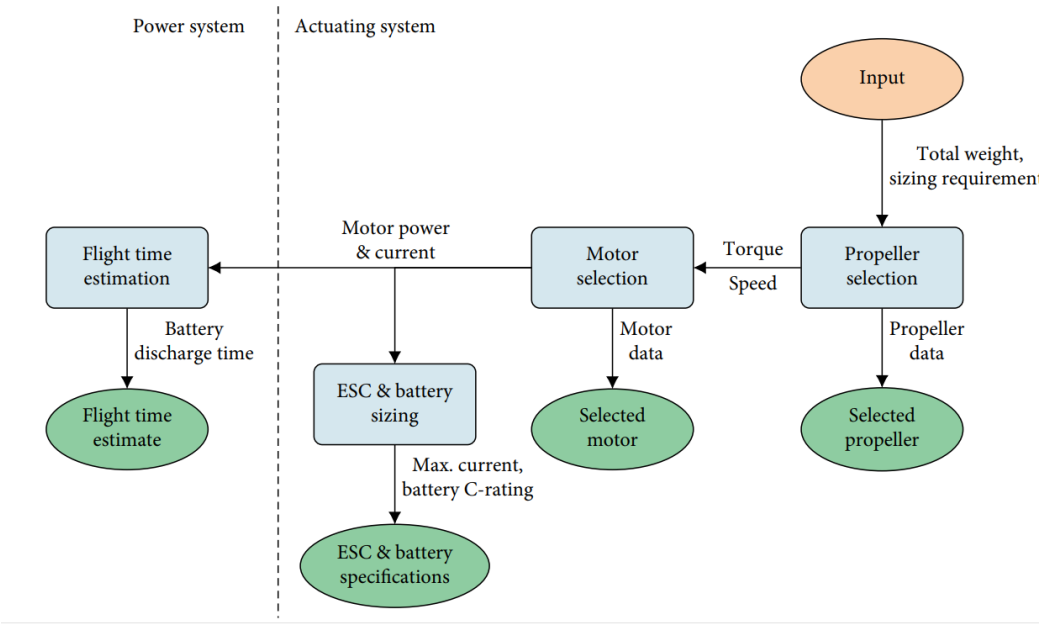


Figure 1.12: Simplified Flowchart of Sizing Methodology for Flight Time Estimation [39]

The preliminary design for the quadrotor may be based on existing design trends, which are analysed to extract the correlation in the configurations of the

quadrotor. All these relationships between the quadrotor components are illustrated as clusters and presented in Figure 1.13-1.16, where later, these relationships were translated to linear regression [32-33].

Figure 1.13 depicted the relationship between carbon-fiber propellers with various masses and the radius and area of their blades, which influence propeller mass and help in UAV sizing.

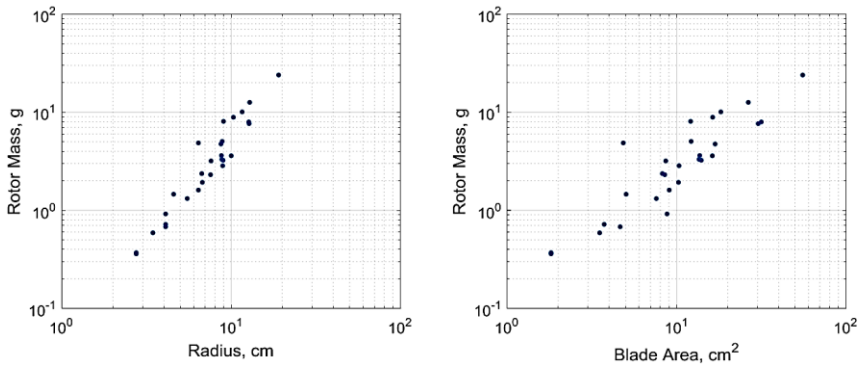


Figure 1.13: Correlation of motor mass with radius and blades area with [33]

Figure 1.14 illustrated the strong relation between the BLDCM weights and their speeds and sizes power.

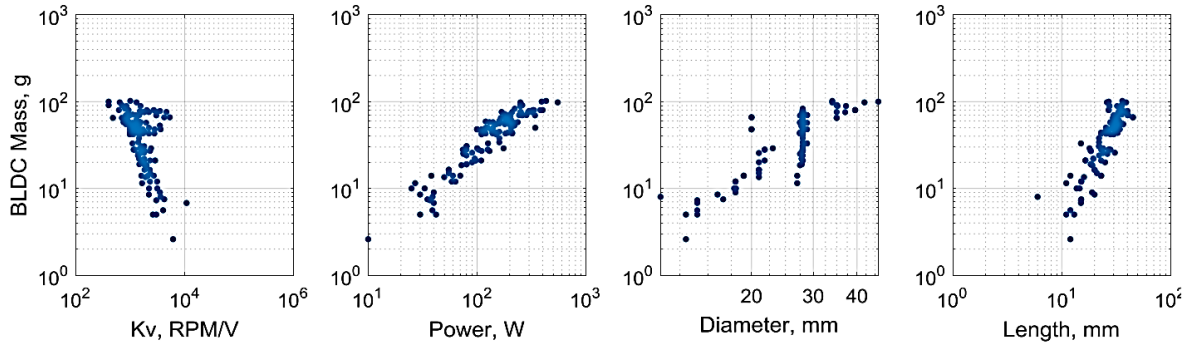


Figure 1.14: BLDC Motor mass correlation with motor KV, Power, Diameter and Length [33]

The influence of battery weight and the propeller diameter on the airframe weight for some quadrotors is shown in the Figure 1.15.

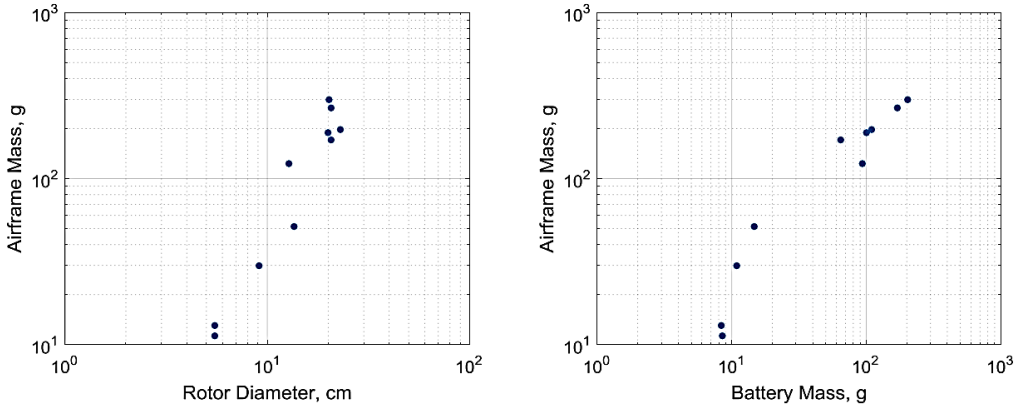


Figure 1.15: Correlation between airframe mass with battery mass and motor diameter [33]

When the battery's capacity increases, the battery's weight will also increase, as demonstrated in the Figure 1.16.

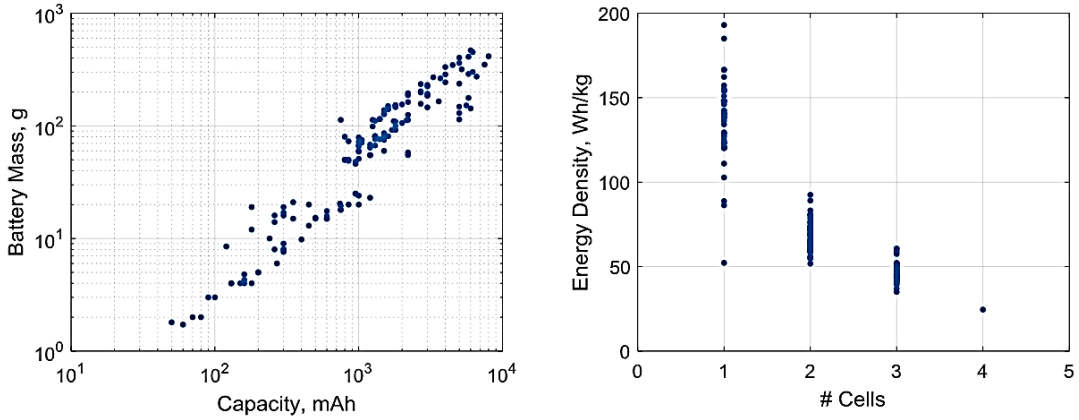


Figure 1.16: Correlation battery mass with battery capacity and cell number [33]

This paper [32] has conducted extensive research on (Design Trends) and preliminary design' of multiprop UAVs, they have gathered data from numerous open sources and analyzed various flying configurations to identify common statistical trends in sub-systems. This analysis aims to draw conclusions and identify patterns that are prevalent among multiple configurations. The paper categorizes each correlation into sub-systems or Design Trends. Only correlations that have reached a significant level are presented below:

The first design trend is determined based on the correlations found in the variables "Propeller Diameter," "Weight," and "Pitch." Figure 1.17 illustrates the relationship between propeller diameter and maximum thrust, representing the maximum thrust achievable by each prop-motor combination.

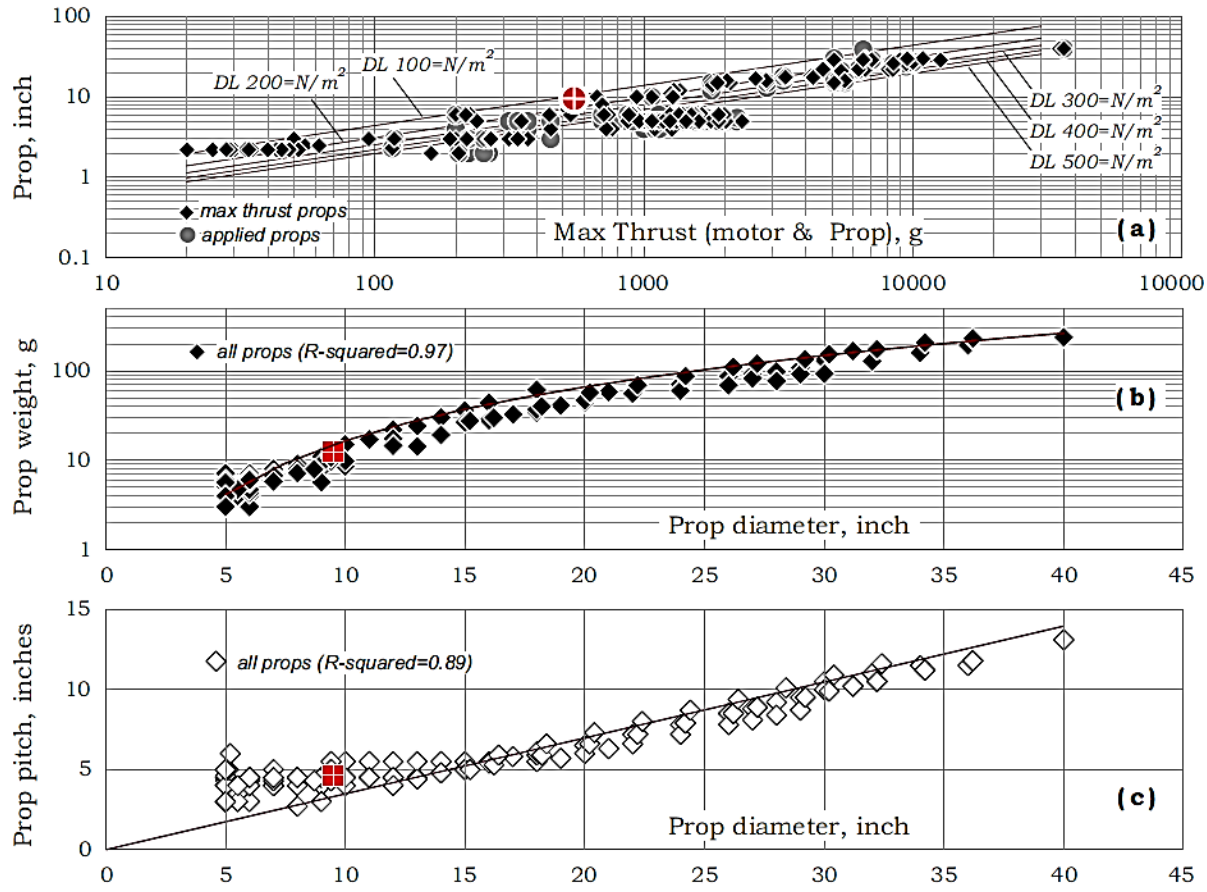


Figure 1.17: Correlation of design trends for propeller diameter with propeller pitch and weight (b, c), and with the maximum thrust of motor-prop (a) [32]

When represented using logarithmic scales, the data suggests a consistent linear trend, as anticipated for constant disc loading. It is worth noting that propeller diameter can be expressed as a function of thrust / disc loading, as defined by the equations in Table 1.4 as first designe trend:

Table 1.4: First design trend "Propeller Diameter" and "Weight and Pitch" [32]

$D_{[inch]} = \sqrt{T_{Max\ [g]}} \frac{4.4}{\sqrt{DL_{[N/m^2]}}}$.	$W_{Prop\ [g]} \cong 0.156 \cdot D_{\ [inch]}{}^2$.

$W_{Prop} \cong \rho_m \cdot (2c) \cdot t \cdot D,$	$W_{Prop} \cong \rho_m \cdot (2c) \cdot t \cdot D,$
$D_{[inch]}$:Propeller diameter. T_{Max}:Thrust. $DL_{[inch]}$ Disc loading. W_{Prop}: Propeller weight. p_m: blade material density. c: average chord t:constant	

The Second design trend is determined based on the correlations found in the variables of Motor Table 1.5.

Table 1.5: Second design trend about variables of Motor [32]

$W_{M\,[g]} \cong 0.032 \cdot T_{Max[g]}$	$W_{M\,[g]} \cong 0.512 \cdot D_{[inch]}^{2}$	$D_{M[mm]} \cong 1.52 \cdot l_{\,[mm]}^{0.46}$
$H_{M[mm]} \cong 0.11 \cdot l_{\,[mm]}^{0.76}$	$D_{M[mm]} \cong 1.79 \cdot T_{Max\,[g]}^{0.4}$	$D_{M[mm]} \cong 7.09 \cdot W_{M\,[g]}^{0.4}$
$KV \cong 241 \cdot 10^3 \cdot T_{Max\,[g]}^{-0.8}$	$KV \cong 15.35 \cdot 10^3 \cdot W_{M\,[g]}^{-0.8}$	
W_M : Motor weight. T_{Max} :Max-thrust (motor/prop) D_M :Motor diameter. H_M: Motor height. KV: Motor KV value $l_{[mm]}$: Quadrotor Frame size		

This design trend is established by analyzing the correlations among various variables, including motor weight, max-thrust (motor/prop), motor diameter, motor height, motor KV value, and quadrotor frame size as showing in Table 1.4-1.5 and illustrated in Figure 1.17 to Figure 1.19

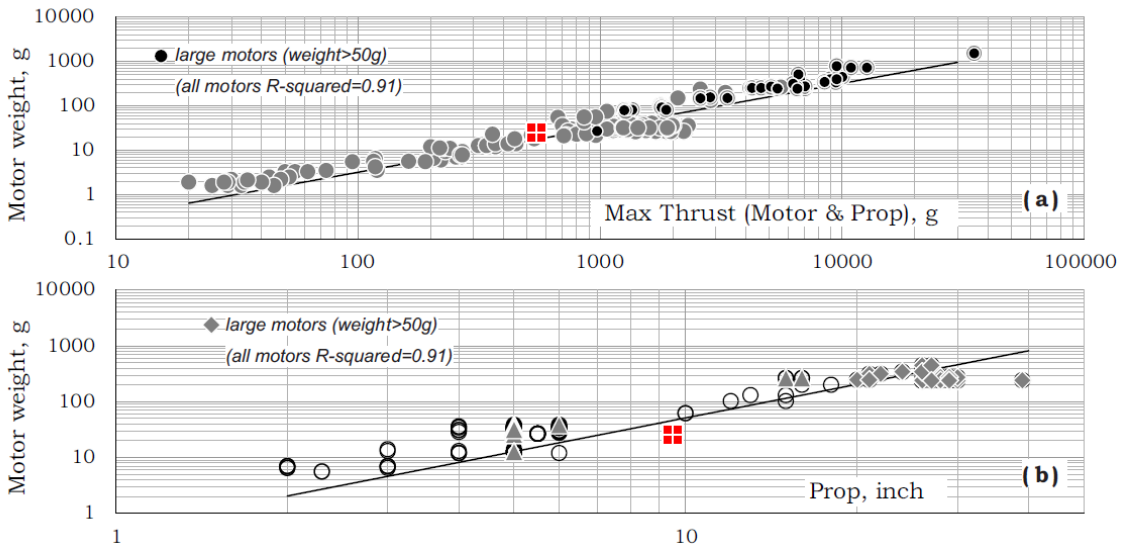


Figure 1.18: Correlation of design trends for motor weight with motor max-thrust (a) and propeller diameter (b) [32]

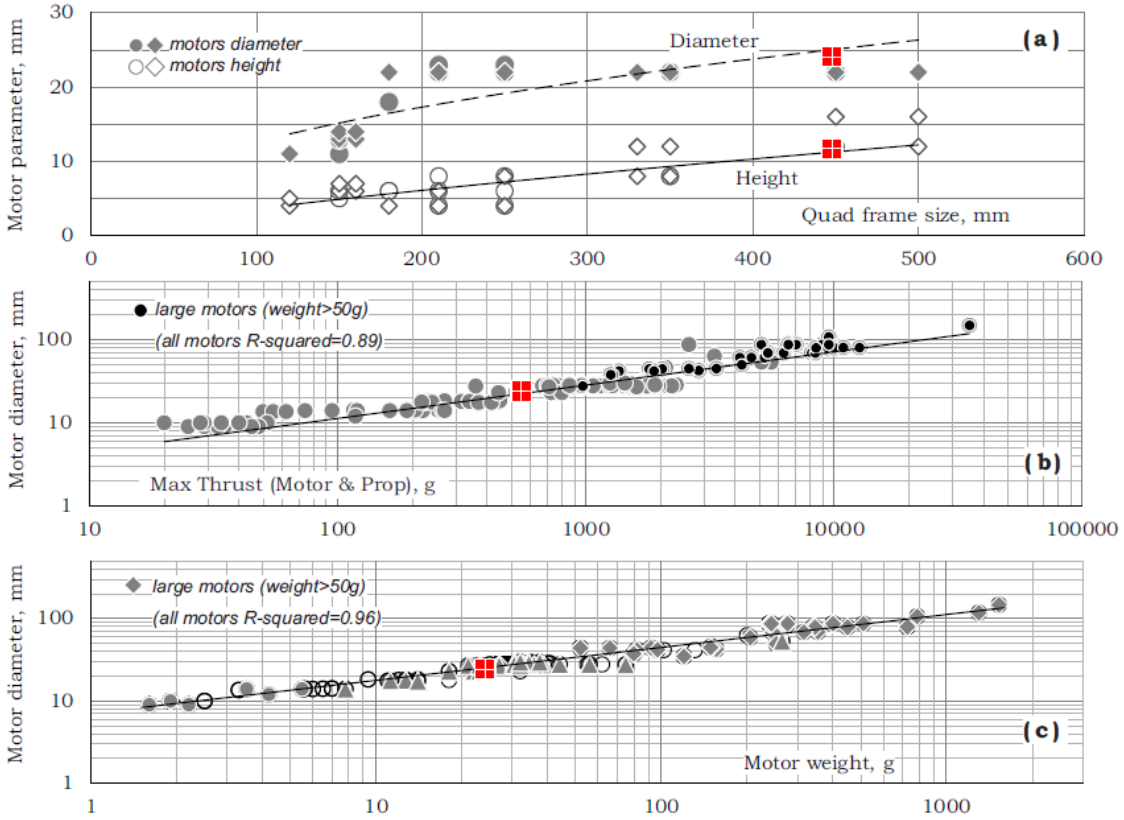


Figure 1.19: Correlation of design trends for motor diameter and height with quadrotor frame size (a), motor thrust (b) motor weight (c) [32]

Table 1.6: Third design trend "operational parameters of motor & propeller [32]

$I_{max[A]} \cong 0.516 \cdot T_{Max[g]}^{0.528}$	$I_{cont[A]} \cong 6.53 \cdot W_{M[g]}^{0.338}$	$U_{[Volt]} \cong 2.21 \cdot T_{Max\,[g]}^{0.284}$
$U_{[Volt]} \cong 5.65 \cdot W_{M\,[g]}^{0.282}$	$P_{[W]} \cong 1.122 \cdot T_{Max\,[g]}^{0.813}$	
$Q_{[Nm]} \cong 2.2 \cdot 10^{-5} \cdot T_{Max\,[g]}^{1.3}$	$P_{[W]} = \Omega_{[rad/\,sec]} Q_{[Nm]} \cong 0.922 \cdot T_{Max\,[g]}^{0.8}$	

Table 1.7: Fourth design trend "propeller rotational velocity" [32]

$\Omega_{[RPM]} \cong \frac{4 \times 10^5}{\sqrt{T_{Max\,[g]}}}$	$\Omega_{[RPM]} \cong \frac{1 \times 10^5}{D_{[inch]}}$

Table 1.8: Fifth design trend "Batteries Characteristics" [32]

$C_{Batt\ [mAh]} \cong 0.129 \cdot l_{\ [mm]}^{1.69}$	$M_{Batt\ [g]} \cong 4.68 \cdot 10^{-2} \cdot C_{Batt\ [mAh]}^{1.10}$
$V_{Batt\ [liter]} \cong 3.1 \cdot 10^{-5} \cdot C_{Batt\ [mAh]}^{1.063}$	$N_c \cong \left\lfloor 2.25 \cdot \log_{10}\left(M_{Batt\ [g]}\right) - 1.25 \right.$
$V_{Batt\ [liter]} \cong 5.98 \times 10^{-4} \cdot M_{Batt\ [g]}^{0.966}$	$E_{Batt\ [Wh]} \cong 9.25 \times 10^{-2} \cdot M_{Batt\ [g]}^{1.072}$
$E_{S\ [Wh/kg]} \cong 74.2 \cdot C_{Batt\ [mAh]}^{0.079}$	$E_{\rho\ [Wh/liter]} \cong 112.017 \cdot C_{Batt\ [mAh]}^{0.116}$
$E_{Batt\ [Wh]} = 3.7 \cdot 10^{-3} \cdot C_{Batt\ [mAh]} \cdot N_c$	

Table 1.9: Sixth design trend "Payload" [32]

$\left.\frac{W_{PL}}{GW}\right\|_{\text{FSHs \& MPCs}} \cong 0.36$	$\left.\frac{W_{PL}}{GW}\right\|_{\text{RWUAVs}} \cong 0.2.$

Table 1.4 to Table 1.9 provide detailed information on the design trends related to "Motor & Propeller Operational Parameters," "Propeller Rotational Velocity," "Batteries Characteristics," and "Payload." These tables present the correlations and findings regarding these specific variables. While the paper [32] provides comprehensive details about the equations and their associated variables.

Additionally, other works, such as [40-41], address questions related to determining optimal values, maximum magnitudes, and the interplay between quadrotor physics parameters like weight, speed, power, and more. These works offer insights into understanding the appropriate values and the interactions among different characteristics in the quadrotor system.

For a given weight, similarity evaluation Total thrust F_{TT} or weight W_{eight} (F_{TT}=$W_{eight} = mg$) and the drag force are proportional to Sw^2 by same degree [40].

$$F_{TT} \propto Sw^2 \propto L^2w^2 \tag{1.1}$$

w is the air speed at wing and L is quadrotor radius, while S is the wing area. Horizontal flight power P is proportional to Sw^3 led to $P \propto L^2 w^3$, for a constant weight, based on thrust formula and from (1.1):

$$S \propto 1/w^2 \Rightarrow L \propto 1/w, \text{ or } w \propto 1/L \quad (1.2)$$

Result: *If the aircraft is twice as large for the same weight, it will travel at half the speed.*

$$P \propto L^2 w^3 \text{ and from (2), } P \propto 1/L \quad (1.3)$$

Result: *The twice-as-large aircraft will require only half the power for same weight.*

The propeller theory provides the following similitude laws, which are simple to derive from a simplified model [40].

$$\text{Thrust } T: \quad T \propto Sw^2 \propto L^2(NL)^2 \propto N^2 L^4 \quad (1.4)$$

$$\text{Torque } M: \quad M \propto Sw^2 L \propto N^2 L^5 \quad (1.5)$$

$$\text{Power } P: \quad P \propto Sw^2 L\text{N} \propto N^3 L^5 \quad (1.6)$$

N is the rotational speed of quadrotor. Since we are interested in determining the blade optimal dimensions at given thrust, then if the thrust is constant $T \propto N^2 L^4$ led to $N \propto 1/L^2$ therefore:

$$\text{Power become P} \propto N^3 L^5 \propto 1/L \quad (1.7)$$

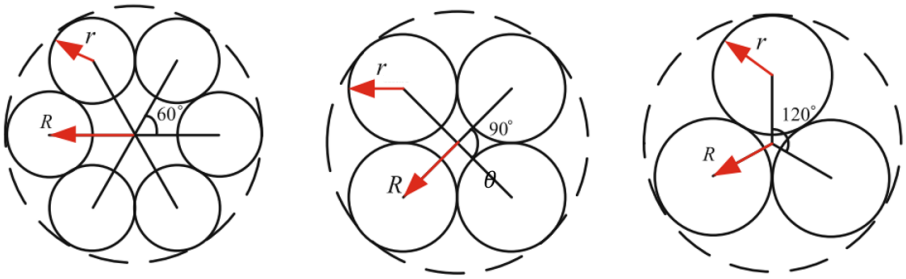
$$\text{Reynolds number become constant} \quad R_{nb} \propto \text{N}L^2 \quad (1.8)$$

Result: *Consequently, a twice-as-large propeller will spin at one-fourth the speed and require half the power.*

Another quadrotor investigation in [41] yields the following findings. The mass m and inertia moment I approximate the following relation:

$$\text{m} \sim R^3, \; I \sim R^5 \text{ where } R \text{ is } R = r/\sin\left(\frac{\theta}{2}\right) \quad (1.9)$$

Where r is the propeller maximum possible radius which prevents the adjacent blades from overlap. θ is the angle between the quadrotor frame adjacent arms equal to 90° in quadrotor case and dominated by law, $\theta = 360°/n$ for the other multirotor configuration as showing in Figure 1.20, where "n" is the arms number of multirotor, n=4 for quadrotor configuration.



(a) **n=6 (θ = 60°)** **Hexarotor**

(b) **n=4 (θ = 90°)** **Quadrotor**

(c) **n=3 (θ = 120°)** **Trirotor**

Figure 1.20: Multirotor configuration, arms number vs propeller radius

Because the purpose of this paper is to provide a comprehensive understanding of drone design and control, rather than focusing solely on optimization aspects such as improving flight time or payload capacity. In order to achieve this goal, the researchers decided to enhance the selection of drone components by utilizing existing toolboxes such as eCalc and FlyEval. These quick tools facilitate the evaluation and analysis of various drone parameters, contributing to a more informed and efficient design process with the least amount of effort and cost. By incorporating these tools into study, we aim to enhance the overall understanding and effectiveness of quadrotor design and control. However, this does not preclude us from developing the design in the future by employing optimization strategies, which provides us with a comprehensive view and a somewhat useful extrapolation regarding the quality of the quadrotor behaviors we will obtain after entering the available components as shown in experimental section.

As specified, the purpose of this study is the systematic development of how to design and control a quadrotor, with the goal of serving as a quick and easy reference for other researchers in this field.

1.4 Conclusion

The focus placed on UAVs prompted us to develop our scientific methodology based on in-depth studies of UAV design and control. The quadrotor was the subject of the most research and was the most prevalent used among these subjects. In this study, we walked through the essential steps that every researcher must follow to accomplish a coherent and well-considered project. The classification of unmanned aerial vehicles was examined first, followed by the most important physical properties of the quadrotor frame and its types, the effect of size on speed and power, the relationships between weight and propeller properties, etc., and the most appropriate choice for motors and blades and all quadrotor parts, as well as the most commonly used techniques for UAV sizing and their benefits and drawbacks. As the UAV industry continues to grow, the importance of UAV will only continue to increase, providing opportunities for further research and innovation in this dynamic field.

CHAPTER 2

QUADROTOR HARDWARE AND SOFTWARE OVERVIEW

Contents **Page**

2.1 Introduction

This chapter provides a comprehensive overview of the most commonly employed hardware/software architecture of quadrotors and explores popular fully-prepared hardware design options for open-source quadrotors, which are low-cost and suitable for high-level UAV control and educational purposes [50-51]. The aim is to serve as a starting point for individuals interested in designing or building their own quadrotor. The chapter begins with a hardware and software overview of the quadrotor system, including an examination of key components such as the flight controller and computer [74-79]. The flight controller processes sensor data, executes control algorithms, and sends commands to the propulsion chain and actuators. The importance of sensors in providing information about the quadrotor's position, orientation, and environmental conditions is also discussed. The propulsion chain, comprising propellers, motors, electronic speed controllers, and batteries, is examined in detail [56-58]. These components collaborate to produce thrust and power for the quadrotor's flight. A deep understanding of their functions is vital for enhancing performance and efficiency. Communication protocols are essential for seamless interaction between different components of the quadrotor system. The chapter explores various communication protocols used in quadrotors, including those between the flight controller and sensors, as well as communication with external devices or ground control stations. Battery technology is discussed due to its direct impact on flight time and overall endurance [63-64]. Considerations for selecting and managing the battery system to ensure safe and efficient operation are covered. The chapter highlights notable open-source projects in both hardware and software that have contributed to quadrotor technology advancement [65-68]. These projects offer valuable resources and platforms for researchers, hobbyists, and educational institutions interested in quadrotor development. Lastly, the chapter touches upon the quadrotor testbench, a crucial tool for evaluating and refining quadrotor controllers. Various types of testbeds have been developed to assess factors such as aerodynamics, altitude and attitude control, and motion control in different degrees of freedom [231-233]. The significance of testbeds in the iterative design and improvement of quadrotor controllers is discussed. By combining knowledge of quadrotor hardware and software components, exploring open-source projects, and utilizing a comprehensive testbench, individuals can embark on designing and building their own quadrotor with a solid foundation and access to abundant resources. The aim of this chapter is to provide a comprehensive

overview of the state-of-the-art in quadrotor hardware design and to provide a starting point for anyone looking to design or build their own quadrotor.

2.2 Quadrotor Hardware and Software Overview

The widespread use of drones in a wide range of fields has resulted in the emergence of numerous topologies and architectures for designing the contents of UAVs including the quadrotors [47], These architectures change by changing the application to which the quadrotor is subject. In general, the quadrotor is controlled by an onboard autopilot system that reads, gathers, and analyses signals from a vast array of onboard sensors and controls the actuators in order to stabilize the quadrotor or some devices, such as a camera, payload, or special device, as well as track objects, avoid obstacles, and perform numerous other tasks. Through communication equipment, a link is established between the quadrotor and the ground station, where the operators conduct data processing, display, and remote command [48-49]. In Figure 2.1 shows the system architecture of the benchmark quadrotor PX4, which is the most popular platform among researchers. The first option involves using only an open-source flight controller, such as Pixhawk, to handle the core flight control tasks while a separate ground station computer manages additional tasks like mission planning, telemetry, and data analysis.

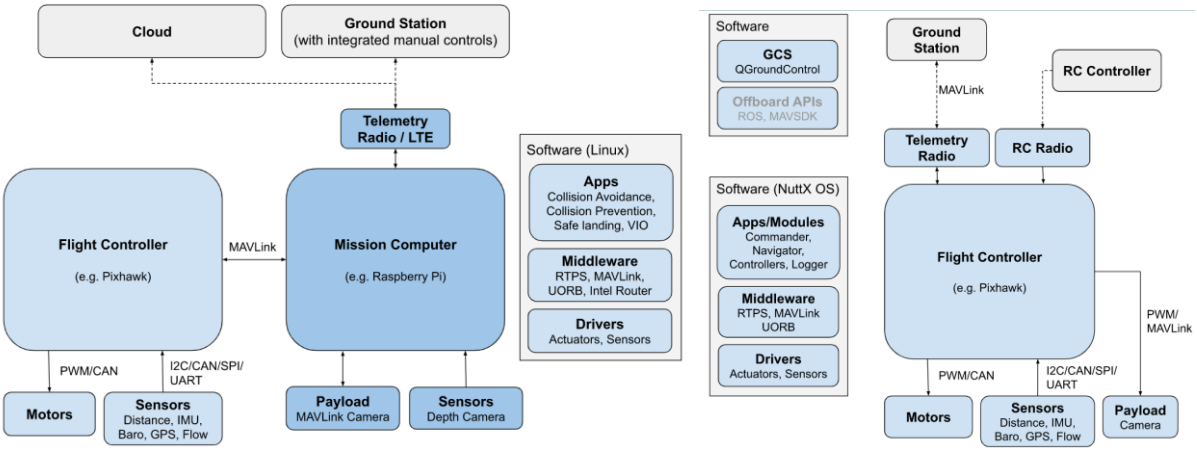


Figure 2.1: Hardware/Software architecture PX4 quadrotor benchmark [50]

In the second option an onboard flight computer or (mission computer) e.g., (Raspberry Pi) has been used in addition to the flight controller, it commonly utilizes

Linux (ROS-based) as its operating system to execute advanced functionality such as collision prevention and obstacle avoidance, and it uses the MAVLink protocol for its communications between the ground stations and its flight controller. Numerous other ready-to-use platforms have been introduced in recent years, each with its own advantages and disadvantages, covering a broad spectrum of price points. [49].

For the software architecture MAVCONN, an aerial middleware, is suggested by the Pixhawk project [51]. The MAVCONN offers hardware-level synchronization of visual and inertial data and serves as a bridge between both the ground station and low-level components of the system. The gateway system that would include common features and communication channels has suggested by [52]. Where the container manages message subscription automatically, delivery messages, and message failure situations, its concept focuses, network-centric embedded systems. An architectural framework implementing fundamental components for the improvement of real time applications and fault-tolerance is proposed in [53]. [54] suggested a UAV collaboration architecture, where this architecture consists of two distinct layers, the executive-layer is the first layer responsible for producing planning and high-level decisions, where the second layer is for execution, the requirements tasks.

Our perspective on the overall architecture of quadrotors is depicted in Figure 2.2, which comprises the following components.

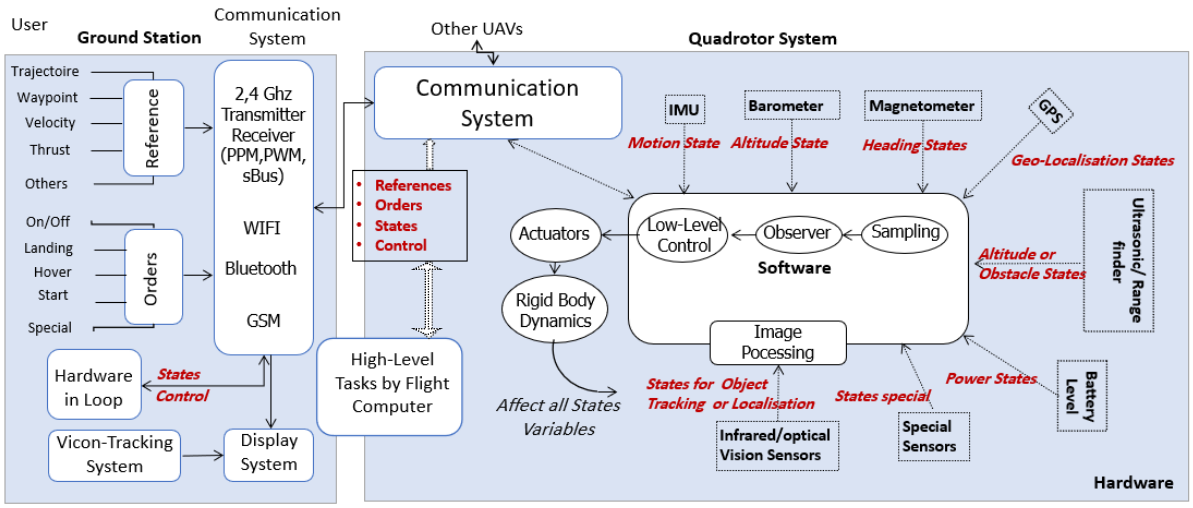


Figure 2.2: Our perspective for overall architecture of quadrotors

2.2.1 Fight Controller and Computer

Fight Controller is a printed circuit board that behaves as the quadrotor's brain. The embedded single-board microcontroller aids in running all activities tasks such as signals acquisition and processing for state estimation, via analog or digital inputs wired to on-board sensors as Inertial Measurement Unit (IMU), pressure sensors, magnetometers, GPS, camera, radio commands, etc, and tasks as well as code execution for closed-loop control, and therefore via its outputs can send control signal for actuators by means of Electronic Speed Controllers (ESC), and can send communication signals via multi protocols such as I2C protocol serial communication that connecting lower speed peripheral integrated circuit (IC) to this microcontroller unit (MCU) chips for quick intra-board communication. The quadrotor requires two levels of control; the inner stabilization loop, which stabilizes the quadrotor at desired angles and rates, and the outer stabilization loop, which is responsible for stablization the quadrotor trajectory to get the quadrotor in desired 3D coordinates.

The Flight Controller is responsible for keeping the quadrotor stable and airborne in real-time (inner loop stabilisation). Receives attitude and altitude references from either the remote pilot or the onboard flight computer.

The Flight Computer is an embedded computer with high-level tasks such as high-level decisions, mission planned generate the trajectory for the outer controller loop, and request states information from flight controller as task require, navigation using GPS data, tracking using camera data etc. An interrupt handler is required by the flight computer to process sensor data as it becomes available. ROS, a widely adopted system for flight calculations, facilitates seamless interaction with various sensor computations and outputs in autonomous quadrotors, which can incorporate a mix of onboard cameras, lidar, or GPS to determine the drone's spatial position. The CPU of a flight computer may become overloaded if computer vision algorithms require excessive computations, which may cause the system to stutter or even crash. In general, it is preferable to have a dedicated microcontroller for stabilization that guarantees the vehicle will be stabilized regardless of the flight computer's health. If the flight computer encounters an error, you can always switch back to the pilot transmitter to land the quadrotor safely. The flight controller is therefore a dedicated

board whose sole purpose is to stabilize the vehicle, whereas the flight computer handles all on-board decision-making and sensing in the same manner as a pilot.

2.2.2 Sensors

A sensor is a device that distinguishes variations in physical quantity and, as a result, generates an output as confirmation of the amount of change. Figure 2.3 depicts the fundamental sensors used in UAV quadrotor system.

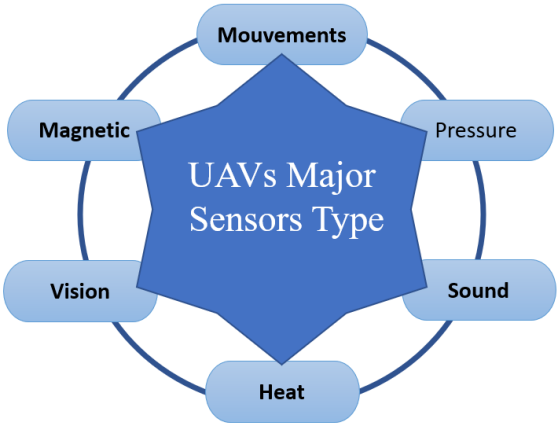


Figure 2.3: UAVs the most used sensor types

Gyroscope, is one of the principal sensors used in quadrotors for measuring and maintaining orientation or the rotation in three degrees of freedom (3 DOF). The gyroscope is essential for the proper control of a quadrotor's attitude.

Accelerometer, is the second main sensor and effectively detects linear motion on every axis of the quadrotor, covering translational three degrees of freedom (3 DOF). Together with the gyroscope in a so-called Inertial Measurement Unit (IMU), the accelerometer tracks changes in motion and position.

Barometer, is a sensor commonly used in almost all UAV applications to assist the quadrotor in maintaining a constant altitude. In fact, this device measures atmospheric pressure, and because air pressure varies with elevation, this sensor can precisely determine the quadrotor's altitude. Although GPS system may also be used to track the altitude of a quadrotor, but barometers give more accurate readings with faster response so long as it has been calibrated properly.

Magnetometer, or an electronic compass or a magnetic sensor, which is widely used in the UAV area, is a device that detects magnetic field induction and is meant to

provide a north direction; as a consequence, the angle information could well be obtained. The magnetometer can be found together with accelerometer and gyroscope in IMU or with GPS module as Ublox NEO GPS.

The global positioning system (GPS), is a radio-base-navigation system that uses satellites in orbit to provide precise worldwide location, timing, and velocities information. The precision of the geolocation will be determined by the strength of the quadrotor's signal of The GPS module and number of satellites within its range.

Rangefinder, is a device that measures the depth of nearby surfaces; sound waves at a particular ultrasonic frequency or laser pulses in a narrow band were utilized to estimate and typically infer the height of the UAV and enhance quadrotor landing behaviour.

Thermal and infrared sensors, or thermopile sensors is kind of vision sensors, this device has infrared wavelength sensitivity and is used to measure temperature from a distance by sensing the emitted infrared (IR) radiation of objects. The IR light of a heating element is interpreted by these sensors as energy emitted in the IR section of the electromagnetic spectrum. UAVs are used for surveillance, pipe leak detection, agricultural, research, and rescue operations, among a variety of other applications.

Vision sensors, the picture is captured by the sensor when light waves pass through the lens and are converted into electric signals by the light-receiving array element CMOS of the sensor (camera). The received data of a picture is then transmitted as a matrix to the computer's processor. Multispectral and hyperspectral camera, RGB camera. The thermal camera are the known vision sensors in the UAV field

Power sensors, current and voltage levels of the quadrotor battery are extremely important when the quadrotor is in flight because power drops directly affect the quadrotor's stability; flight stability is only guaranteed by compensating for this energy drop.

These sensors are not the only ones carried by the quadrotor; there are additional sensors such as radar, a radiometer, chemical sensors, optic-flow sensors, wind sensors, and more

2.2.3 Propulsion Chain Actuators Propeller-Motor-ESC-Batterie

The quadrotor propulsion chain is an integral part of quadrotor, since it transfers power ($P_{battery}$) from battery to the rotors-propellers through the ESC (P_{ESC}), allowing them to create thrust (T) and torque (M), Figure 2.4 illustrates the propulsion chain.

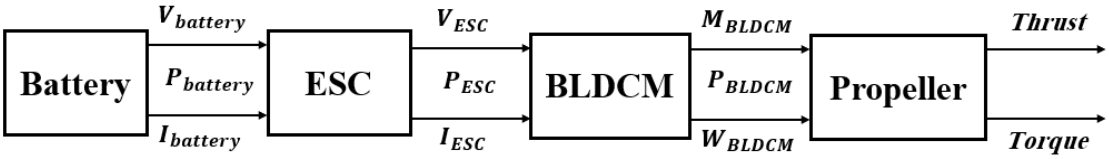


Figure 2.4: Quadrotor propulsion chain

2.2.3.1 Brushless motor

A brushless motor (BLDCM) is an electric actuator powered by direct current (DC) through three wires in electricity phases. Unlike conventional DC motors, the BLDCM runs at high-speed rotation without brushes. BLDCMs are mainly used to create a lift and are more popular than conventional ones due to their efficiency, durability, high speed, precise torques, and low electrical noise. However, its strength and speed cause it to consume a great deal of energy. Figure 2.5 illustrates a common existing motor.

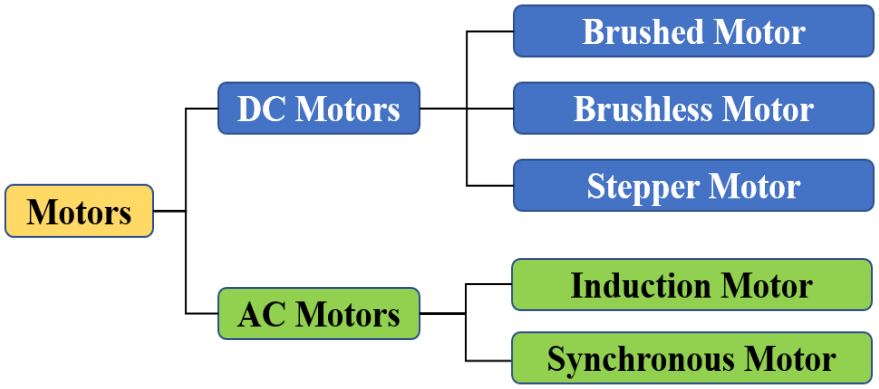


Figure 2.5: Electric motors types

BLDMs comes in various size and designs, but all convert the electricity energy to mechanical energy; this mechanical energy is used in UAV to lift the body and may be used as fun for PC, for the CD drive to turn a CD at high and precise speed, or for battery power driller due to it hight-torque that can be generated. The primary components of the BLDCM in Figure 2.6 are the rotor (permanent magnet) and stator (coils). It operates similarly to conventional DC motors, where there are numerous

resources about its functionality on the internet, where the primary distinctions are the brush option, the wire number, and the velocity adjustment operating principle. Unlike conventional DC, where changing the voltage affects the speed, BLDCM changes the speed by adjusting the pulse timing. There are two types of BLDCM, depending on whether the rotor is inside or outside the motor; inner and outer. The main difference between the two is that the inner is used for applications that require high speed, while the outer is used for applications that require more torque, such as our application, where we need more torque to lift the quadrotor.

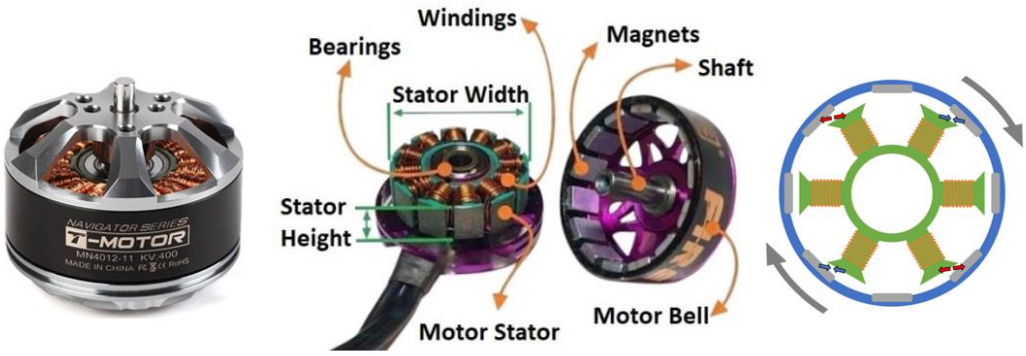
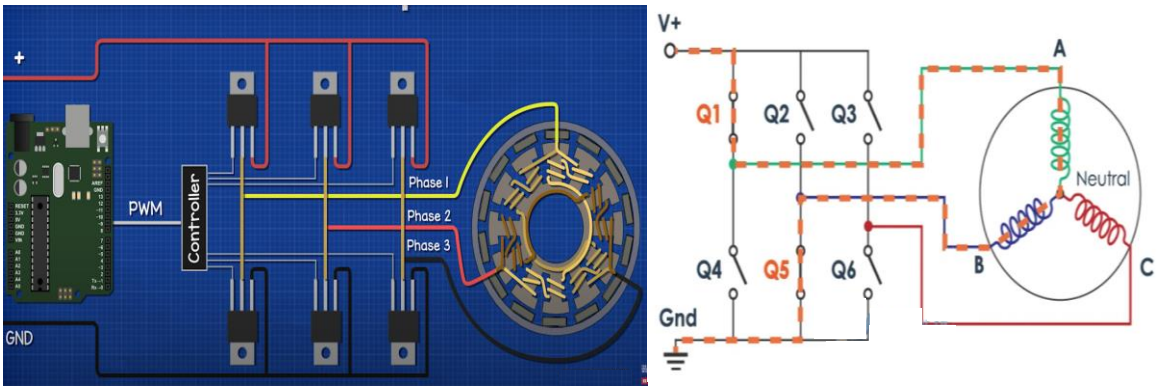


Figure 2.6: BLDC motor primary components

The BLDCM has three wires which is connected the three set of coils (A, B, C) inside the motor Figure 2.7.b and in same time connected to electronic speed controller (ESC) as in Figure 2.7.a noted 'Phase1' phase2, phase3. The flight controller sends a pulse width modulation (PWM) signal to ESC that correspond the motor needed speed, The ESC then pilots the motor from the lowest to highest speed based on the PWM duration, which varies between 1 and 2 ms, the form of PWM signal is illustrated in Figure 2.13.



a **b**

Figure 2.7: BLDC Motor coils and ESC switches

The direction of rotation changes by reversing the polarity or change the two wires of three wires by help of ESC switches (Q_1...$Q6$) [55]. For rotating the BLDCM, the stator's three coils must be activated sequentially. In every commutation sequence, one of the coils is energized positively (Figure 2.7b - coil A), the second is energized negatively (Figure 2.7b - coil B), and the third is kept non-energized or open (Figure 2.7b - coil C), by help of closing the switches Q_1 and Q_5. The sequential control of the switches requires knowledge of the rotor's position, and some BLDCMs use Hall-effect sensors to determine it. Others utilize the third open coil (not energized) of the motor, which carries current induced by the rotation of the armature or back electromotive force (EMF). The application of BLDCM control, including the switching process, back-EMF detection timing for speed calculation, and software implementation, is well presented in [56-58]. An example in Figure 2.8 depicts a switching sequence and the corresponding activated coils. ESC switching 1 and 4 closed to activate the coils A and B, where C used for B-EMF.

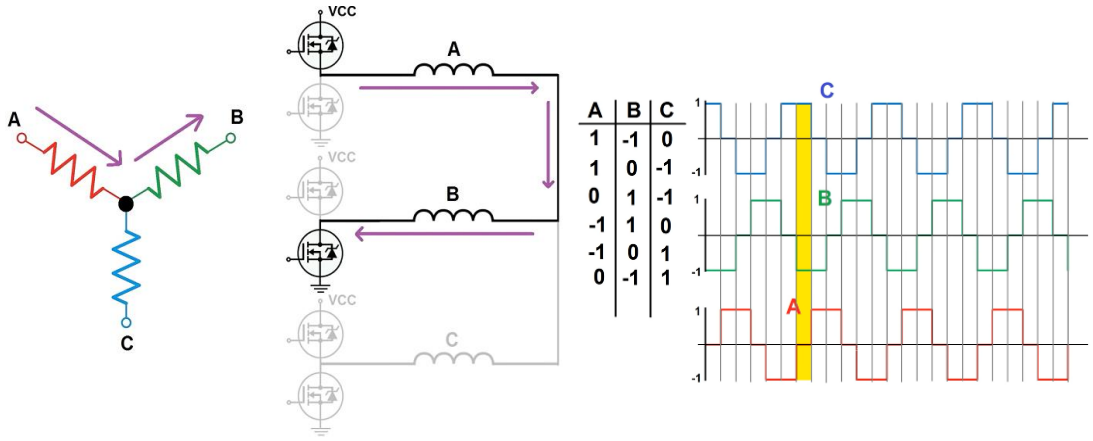


Figure 2.8: ESC switching one state and it correspondent signals and logic

A) BLDCM Modeling

BLDCM is typically powered by three-phase. All the phases are similar; their circuits consist of a resistance R and inductance L as Figure 2.9. The back-EMF voltage, denoted as U_{emf} is caused by the rotor's rotation with angular velocity ω . When a load is attached, it generates a torque "τ" that results in the motor consuming the current "I". The corresponding per phase circuit of a BLDCM is seen in Figure 2.9. Equations 2.1 and 2.2 describe Newton's second for rotational motion and Kirchhoff's laws, explaining the mechanical and electrical behaviors of BLDCM

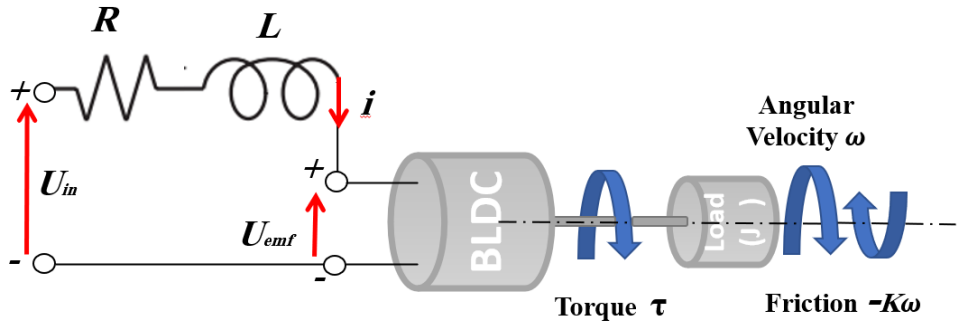


Figure 2.9: BLDCM equivalent circuit

$$U_{in} = \text{Ri} + L\frac{di}{dt} + U_{emf} \quad (2.1)$$

$$J\frac{d\omega}{dt} = \sum \tau_i \quad (2.2)$$

The electromotive force *Uemf* is proportional to the motor's angular velocity (ω), while the current flowing in coil '*i*' is related proportionally to the produced torque (τ), as expressions in equations 2.3 and 2.4.

$$u_{emf} = K_e\omega \quad (2.3)$$

$$\tau = K_m i \quad (2.4)$$

Where K_m [Nm/A] and K_e [V /(rad/s)] are the armature and the emf constants.

2.2.3.2 Propellers

The propeller is a rotating device with a centralized hub and blades arranged such that each produces a helical or spiral shape. It produces a linear push against the air when turned. The determination of the optimal propeller size and shape is performed based on the desired quadrotor frame size and the thrust required by the motors to lift the quadrotor. The propellers in Figure 2.10 are described by two digital numbers, e.g., APC 10X45, the diameter (10 inches) and the pitch (4.5 inches). The diameter is the length of the propeller, and the pitch is the forward travelled distance of a single rotation of the propeller, which is created by the angle of the blades. Figure 2.10 shows the forces and moments acting in propellers blades.

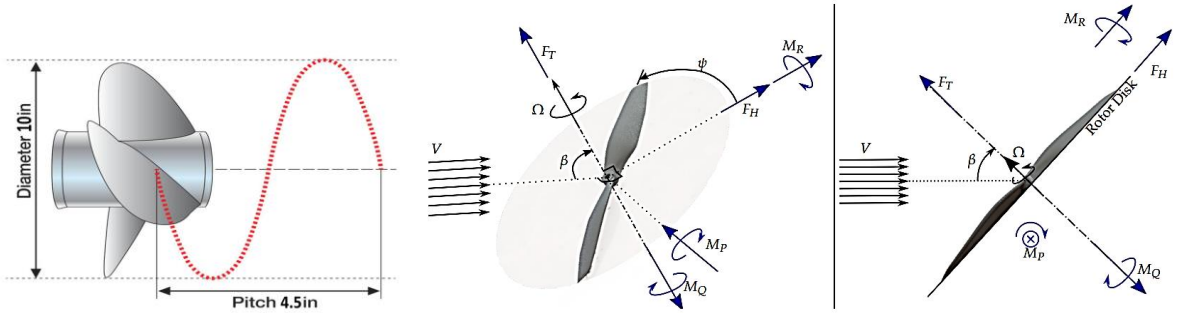


Figure 2.10: Propeller characteristics and applied forces and moments [59]

Where Ω is the rotation rate, which forms and angle β with the incoming winds. The rolling moment M_R and H-force F_H are perpendicular to the pitching moment M_P and the rotor torque M_q, as are the thrust force F_T. all these forces acting in disk rotor center propeller [59], Raffaello D'Andrea has explained in depth the computationally efficient force and moment models for propellers.

2.2.3.3 Motor-Propeller forces (thrust and drag)

While in rotor-propeller rotation, the blades generate aerodynamic forces that have a direct impact on quadrotor dynamics. These forces are governed by the angular velocity and the blade shape. These forces are thrust and drag, where the thrust force equation is:

$$\mathrm{F}_{thrust} = k_T \omega^2 \quad \text{(Newton)} \tag{2.5}$$

$$k_T = C_T \rho A r R^2 \tag{2.6}$$

k_T is the propeller thrust factor, and ω is angular velocity of propeller, C_T is thrust coefficient, ρ (kg/m3) is density of air, A is the propeller's disk area, and R is the propeller radius. While the drag moment of the rotor-propeller is equal to:

$$\tau_i = k_T \omega_i^2 \tag{2.7}$$

$$k_\tau = C_P \rho A r^3 \tag{2.8}$$

Where k_τ is the drag torque factor and C_P is the drag coefficient.

2.2.3.4 Electronic Speed Control (ESC)

ESC is a circuit designed to control and sense the BLDCM's speed by matching the flight controller's PWM references and the BLDCM's speed estimate, typically obtained from back EMF or Hall Effect sensors, or by zero-crossing detection. ESC acts as an intermediary between the BLDCM and the power source (battery). By modifying the switching frequency (duty cycle) of the network of field transistors (FET), the motor velocity will be altered. ESC outputs are a three-phase alternating signal, similar to a variable frequency drive (VFD) [60-61] as presents in Figure 2.11. The ESC consists of three fundamental elements: the Battery Elimination Circuit (BEC) or voltage regulator, a microcontroller, and field effect transistors used for switching.

-Where the BEC/voltage regulator provides 5 volts (1 AH) as a 5 volt power supply to the flight controller or receiver.

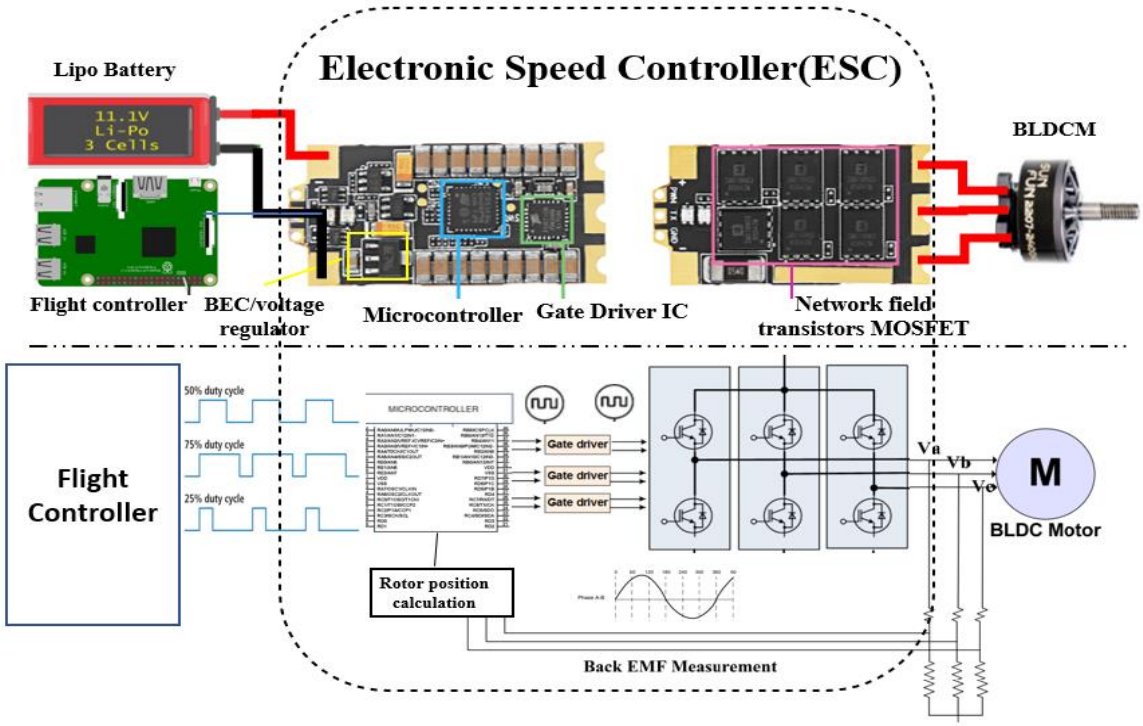


Figure 2.11: ESC main components

-Microcontroller(MCU) for interpreting the information provided by the flight controller or receiver into a properly regulated signal for the motor by switching the FET component in accordance with an internal program (firmware).

-A FET including Power MOSFET network and and Gate Driver IC to sends the whole current and voltage of the battery to the rotor by cutting up the flow of energy to

throttle the motor by switching frequencies to match the needed speeds. To prevent the ESC from overheating and to provide a small amount of flexibility when operating at the highest throttle, the ESC current must be 20% higher than the motor's maximum current. ESC voltage limits are typically specified as battery cell ranges, such as 3S to 4S cells or (11.1 to 14.8 V), where each cell is 3.7 volts. When the voltage falls below 3.4 V, the ESC switches off.

2.2.4 Communication Protocols

The throttle controller signal is employed to adjust the velocity of the BLDCM. Increasing or decreasing the throttle signal will increase or decrease the power output, which alters the switching frequency of the ESC. The throttle control signal needs a means or techniques to be delivery from the remote controller or ground station to the BLDCM. Multiple signal delivery techniques are used to transmit throttle signals from the remote controller or ground station to the ESC via flight controller. Each protocol has somewhat varying functionality, with the most commonly used being PWM protocol, Multishot protocol, Oneshot protocol, and Dshot protocol, which are the most prevalent, the main difference between them is signal frequency (see Figure 2.12). Lower frequencies can accelerate quadrotor signals and responses. Dshot communication sends a digital signal rather than an analog one, providing better reliability when there is less electrical noise and higher resolution. In our project, we used the PWM protocol.

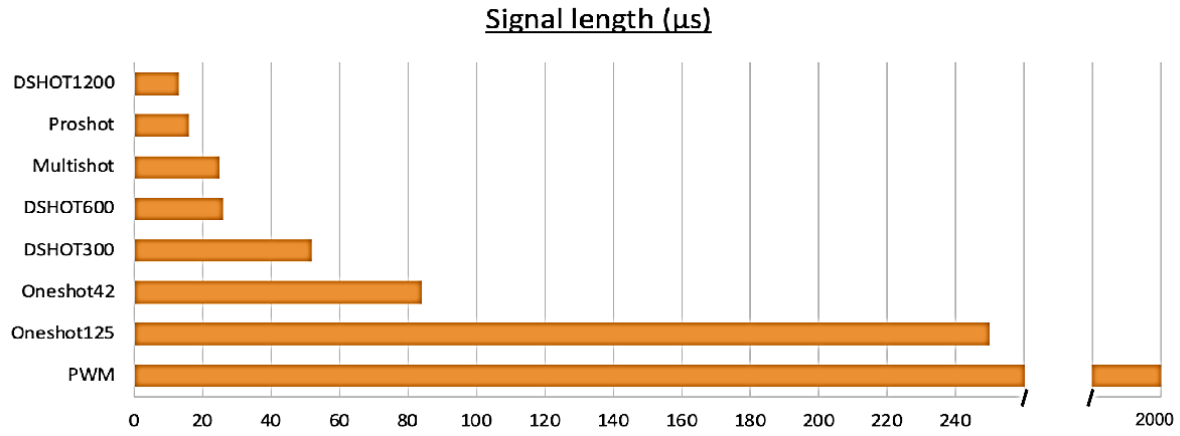


Figure 2.12: ESC and Radio receiver communicate protocols for transmitting the remote controllers' signals (throttle controller signal)

2.2.4.1 PWM Signal

Pulse-width modulation PWM is the first and still-used ESC protocol. PWM sends controlled power pulses to the motor in response to input from the throttle controller. The throttle controller instructs the ESC MCU how much electricity to pull from the battery for the rotor. Pulses determine the duration of the voltage drawn. "on" and "off" voltage pulses are separated by voltage-zero periods. The longer "on" time compared to the shorter "off" time produced more power to deliver and increased the speed of the rotor. Duty cycle is the ratio of on-to-off time. PWM pulses in Figure 2.13 range from 1000 to 2000 microseconds. The gate driver uses the microcontroller's voltage to switch the MOSFETs' three phases. More voltage at the MOSFETs speeds up phase switching and rotor speed.

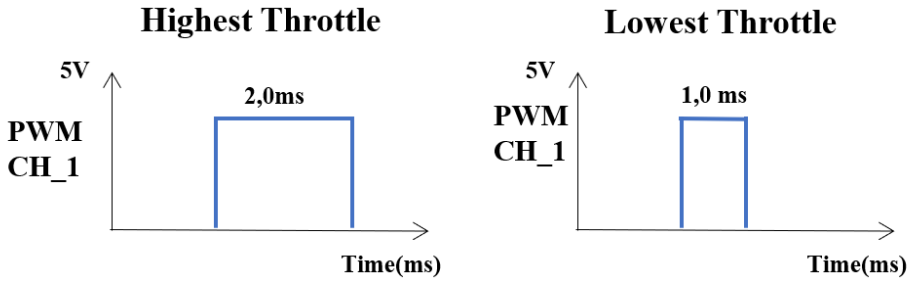


Figure 2.13: PWM signal duration from (1 to 2 ms) for one radio channel one wire

2.2.4.2 PPM Signal

PPM, or Pulse Position Modulation, is another useful modulation that uses a series of fixed-duration pulses to transfer the data (throttle). It is frequently used in transmitters and receivers due to its advantages over the PWM protocol. PPM offers the benefit of obtaining all channel information from a single wire or single port, while PWM needs a wire or port for each channel. The PPM pulses have a duration of 500 microseconds, while its pauses vary from 500 and 1500 micro-seconds. Adding these durations together yields channel values of 1000 and 2000 micro-seconds, which is the standard PWM duration. Figure 2.14 depicts the conversion of three PWM channels to a single PPM channel.

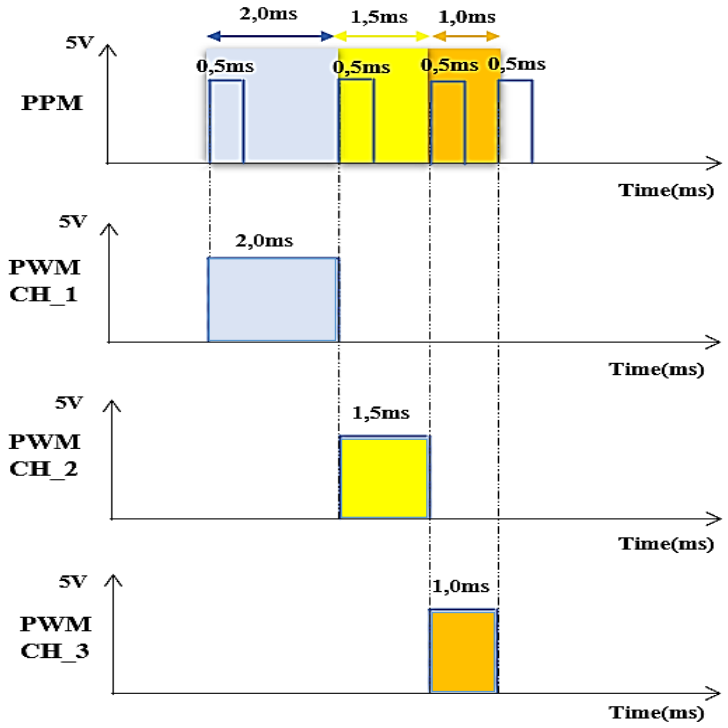


Figure 2.14: PPM Protocol signal duration from for all radio channels at one wire vs PWM three channels at three wires

2.2.5 Battery

Lithium-Polymer (LiPo) battery in Figure 2.15 is increasingly becoming the favoured option for power storage and accumulation, essential in radio control application e.g., drones, rechargeable batteries portable devices, and in many other applications requiring larger amounts of power. A LiPo battery is a suitable power source for UAV applications due to its higher volumetric energy and power densities stored in the battery. It comes in many sizes and shapes, with a lighter weight and higher discharge rate compared to other types of batteries (NiMH, Li-ion batteries). Additionally, it is suitable for greenhouse applications as there are no gas emissions.

Figure 2.15: Lipo Battery three Cell 3S 5000mAh 35C and inside components

On the other side, it is sensitive to overcharging and necessitates careful attention when charging, discharging, and storing. If the battery is broken, it might cause a fire. Lipo batteries have a shorter lifetime (often about 150–250 cycles) compared to NiMH batteries, which have a typical lift time of 1,000 cycles. The lipo battery consists of cells wired in series or parallel; each cell has a nominal voltage of 3.7V and in full charge goes up to 4.2V, the voltage of the battery with two cells or wired in series (2S) is 7.4V, or two times 3.7V, but in charge, it must go up to 8.4V, or multiples of 4.2. Three cells (3S) equal 11.1V, and so on, with the cells wiring in parallel denoted by "P". The battery capacity in milliampere-hours (mAh) is an important factor that indicates the amount of current that the battery can deliver for one hour; it is also a measure of the rate of discharge. Because a battery with a higher capacity is heavier and larger, there's a trade-off between the battery capacity and the battery weight, which impacts flight time and manoeuvrability. While the C-rating factor is the maximum continuous discharging rate of the battery, each "C" is equal to the battery capacity in mAh. Battery 11.1 volts with a capacity of 3000 mAh and a continuous discharge rate of C30 indicates that the battery contains three cells (3S) and can be charged until 12.6 volts with a discharging rate of 3 amps per hour and a maximum continuous discharge rate is 3 amperes times 30 equal to 90 amps per hour. The choice of an adequate battery depends on the application and the maximum load. For example, at 12 volts DC, a quadrotor with a brushless motor, the EMAX XA2212, supports voltages ranging from 7.4 volts to 12.6 volts and consumes current up to 15.1 A to thrust 880 grams for each motor. If we suppose that our quadrotor weighs 1250 grams, then each motor's consummate current is 5.3 A to hover the quadrotor, which needs a battery of three cells (3S) that can handle the delivery of current (4 x 5.3 = 21.2 A). If we choose a

battery with a capacity of 3000 mAh and a C-rate equal to C21 (theoretically), it will give 90 A of continuous current draw, which is more than the needed current for the four motors at full power. The flight time is the ratio between the battery capacity and the average consumed current; 3000 mAh divided by 21.3 gives 8.57 minutes of flight. While the battery cut-off voltage is the battery voltage when it completely discharges, and under this voltage, the battery may be damaged and lose its performance. Generally, the cell cut-off voltage is about 3.3 volts. The following are some battery terminology terms used by researchers in the field [62-64]:

State of Charge (SOC): The current amount of energy stored in a battery as a percentage of its maximum capacity.

Depth of Discharge (DOD): The percentage of the total energy in a battery that has been discharged or used.

Terminal Voltage: The voltage measured across the terminals of a battery or an electrical device.

Open-Circuit Voltage: The voltage presents across the terminals of a battery or a device when no current is flowing.

Internal Resistance: The inherent resistance within a battery or an electrical device, which affects the flow of current and can result in energy loss and voltage drop.

The electrical equivalent simple first-order circuit model for LiPo batteries, which is used for battery monitoring, parameter identification, battery fault diagnostics, etc., is shown in Figure 2.16. Where $V_{OC}(SOC)$ is the Lipo battery voltage in an open-source circuit that changes nonlinearly with the SOC , R_i represents the connection and electrolyte resistance, R_{df} is internal polarized resistance while C_{df} represent the polarization capacitance, the measured current pulled by the load is denotes by I_{bat} which is positive in discharging and negative in charging, and V_{bat} is the apparent voltage that can measured (using a multimeter), the nonlinearity of charge is represented by the parallel RC network.

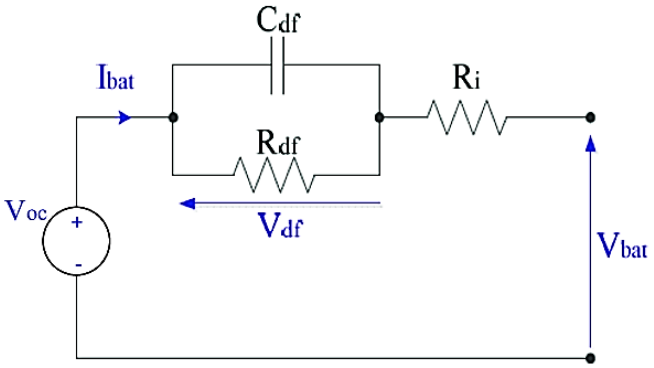


Figure 2.16: Simple First order electrical equivalent circuit model for lipo battery

$$V_{bat} = V_{OC}(SOC) - I_{bat}R_i - \sum_{i=1}^{n} V_{df_{RC}} \quad (2.9)$$

where i is number of RC networks in our case (i=1).

2.3 Exploring UAVs Open-Source Autopilot Hardware and Software

The most popular totally prepared hardware and software are low-cost and dedicated to quadrotors, and they are readily usable by other academics to perform high-level UAV control for educational reasons. They are described below.

2.3.1 Open-source autopilot software

Autopilot software offers a complete toolkit appropriate for practically any UAV; the majority of autopilot softwares used in this field are ArduPilot, Multiwii, Cleanflight, LibrePilot, AutoQuad, Dronecode Community. The most commonly used programming languages for UAV autopilot platforms are C++, Python, Java, and assembly language. Where a high-level software used for high task are, Matlab, ROS, Gazebo, Open Robotics.

2.3.2 Open-source autopilot hardware

A variety of embedded autopilot hardware systems have been utilized in UAVs, but those listed below remain the most prevalent. It is possible to classify based on the embedded platform [65].

2.3.2.1 Atmel family: ArduPilot Mega, FlyMaple.

FlyMaple in Figure 2.17 is an ARM-based multi-rotor and wheel balancing robot autopilot developed by Maple-Project. The project is released under the GPLv3 license.

Figure 2.17: Opensource autopilot flight controller FlyMape Atmel family

2.5.2.2 ARM family: PIXHAWK/PX4, Paparazzi, CC3D and Atom.

CC3D and Atom are flying controllers in Figure 2.18 that have the same functionality and operate on Open-Pilot/Libra-Pilot firmware. designed to fly in any type of aircraft, from fixed-wing to multirotor. The project's software and hardware are accessible under the GPL v3 license.

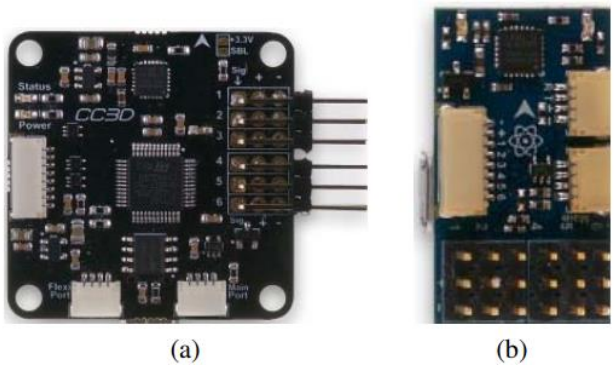


Figure 2.18: Opensource autopilot flight controller (a) CC3D (b) Atom ARM family

2.5.2.3 Raspberry Pi family: Erle Brain 3, PXFmini, Navio2

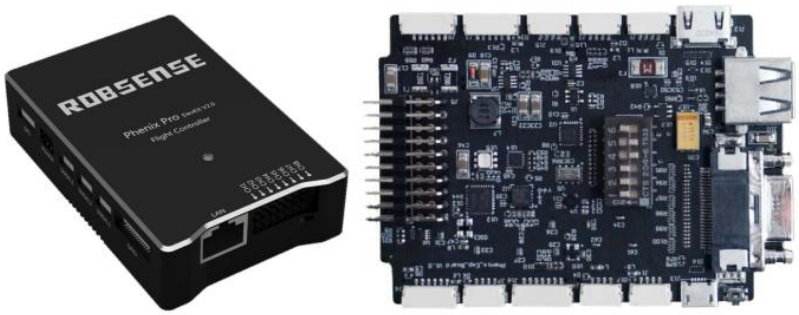
Erle-Brain 3 in Figure 2.19 is an open-source autopilot Linux-based system, embedded computer known as the Raspberry-Pi, designed by Erle-Robotics, that has a number of sensors. It is based on Drone-code Foundation Technologies, and the schematics are available under the Creative Commons license.

(a) (b)

Figure 2.19: Opensource autopilot flight controller (a) Erle-Brain 3 (b) PXFmini Raspberry Pi family

2.5.2.4 FPGA family: Phenix Pro, OcPoC ,

Phenix Pro is powered by a UAV real-time operating system based on FreeRTOS, equipped with onboard sensors, lidar, thermal cameras, radar, ultra-vision, and radio capabilities. It also supports neural networks and vision computers through FPGA acceleration. The system integrates a task scheduler and utilizes ROS for intelligent algorithms and hardware resource management. The software development is available under the GNU General Public License (GPL) v3 license. The Phenix Pro controller is illustrated in Figure 2.20.



(a) (b)

Figure 2.20: Opensource autopilot flight controller (a) Phenix 3 Enclosure (b) Phenix 3 Circuit-board

While these projects have been the most active, other projects become less active lately include AeroQuad, Mikrokopter, MatrixPilot.

The processors that were included in the autopilot platform are: STM32, ATMEGA2560, Xilinx Zynq, FPGA SoC, Cortex-A9/M4F.

The most prevalent quadrotor autopilot projects, as outlined in references [66-68], include Arducopter, Pixhawk, Openpilot, Paparazzi, Mikrokopter, Multiwii, KKmulticopter, and Aeroquad. Additionally, Parrot Mambo, Parrot Bebop 2, and Ryze Tello are quadrotors with support packages in Matlab software. Table 2.1 provides a comparison of several open-source hardware platforms, describing their characteristics, while Table 2.2 outlines the characteristics of open-source software platforms [65].

Table 2.1: Most used open-source hardware platforms and their characteristics

Platform	Processor	Sensors	Interfaces	Weight (g)	URL
Phenix	Xilinx Zynq SoC (ARM Cortex-A9) "Cyclone V" /	HUB, IMU, GPS, LED	CAN, HDMI, Camera Link, LVDS, BT1120-PL	64	www.robsense.com
OcPoC	"Xilinx Zynq" FPGA SoC (ARM	IMU, Barometer, GPS, Bluetooth, WiFi	PWM, I2C, CAN, Ethernet, SPI, JTAG, UART, OTG	70	www.raerotenna.com
PIXHAWK/ PX4	Cortex-A9) ARM Cortex-M4F	IMU, Barometer, LED	PWM, UART, SPI, I2C, CAN, ADC	38	www.pixhawk.org
PIXHWEK2	STM32F427	IMU, Barometer, LED	PWM, UART, SPI, I2C, CAN, ADC XBEE, PWM,	-	www.proficnc.com
Paparazzi (Chimera)	STM32F767	IMU, Barometer	UART, SPI, I2C, CAN, AUX	-	www.paparazziuav.org
CC3D	STM32F	gyro, accelerometer	SBus, I2C, Serial	8	http://opwiki. readthedocs.io/en/latest/ user_manual/cc3d/ http://opwiki.
Atom	STM32F	gyro, accelerometer	SBus, I2C, Serial	4	https://opwiki.readthedocs.io/en/latest/user_manual/cc3d/cc3d.html
APM 2.8	ATMEGA2560	IMU, Barometer, LED	UART, I2C, ADC	31	www.ardupilot.co.uk
FlyMaple	STM32	IMU, Barometer	PWM, UART, 12C	15	www.emlid.com
Erle-Brain: PXFmini	Raspberry Pi shield	IMU, Barometer	PWM, UART, I2C, ADC	15	www.erlerobotics.com

Table 2.2: Most used open-source software platforms and their characteristics

Platform	Running Processor	Programming language	Website	Source Code
ArduPilot	32-bit ARM	C++	www.ardupilot.org	www.github.com/ArduPilot
MultiWii	8-bit ATmega328	C	www.multiwii.org	https://code.google.com/p/multiwii/
AutoQuad	32-bit ARM	C	www.autoquad.org	www.github.com/mpaperno/aq_flight_control
LibrePilot	32-bit ARM	C++	www.librepilot.org	www.bitbucket.org/librepilot& www.github.com/librepilot

2.4 Quadrotor Testbench

A quadrotor testbench is an essential tool for evaluating and refining the design of a quadrotor controller. It provides a controlled environment where various aspects of the quadrotor's performance can be thoroughly tested and analyzed. The goal of such testing is to enhance the quadrotor's stability, maneuverability, and overall flight characteristics. Several types of testbeds [231] have been developed to cater to different aspects of quadrotor performance evaluation. One common focus is on the impact of aerodynamics. These testbeds are designed to assess how the quadrotor interacts with the surrounding air, including factors such as wind resistance, turbulence, and airflow patterns. By examining the aerodynamic behavior, engineers can make adjustments to optimize flight efficiency and reduce energy consumption. Another important aspect of quadrotor testing is altitude and attitude control. Testbeds dedicated to this area allow researchers to assess the quadrotor's ability to maintain a desired altitude and attitude during flight. These testbeds typically include sensors and measurement systems to monitor the quadrotor's position, orientation, and stability. By analyzing the data collected during these tests, engineers can fine-tune the controller's algorithms to improve altitude and attitude control accuracy. Motion control in six degrees of freedom (6DOF) is another area of interest in quadrotor testing [232]. A 6DOF testbed enables evaluation of the quadrotor's ability to move freely in all directions—translation along the x, y, and z axes, as well as rotation around each axis. This type of testing is crucial for assessing the quadrotor's agility, responsiveness, and dynamic stability. Engineers can use the data gathered from 6DOF tests to refine control algorithms, enhance manoeuvrability, and achieve better overall performance.

In addition to 6-DOF, there are also testbeds that focus on four degrees of freedom (4-DOF) [233]. These testbeds typically restrict the quadrotor's motion to a specific plane or axis, allowing for more specialized testing. For example, a 4-DOF testbed might focus on evaluating the quadrotor's performance in terms of attitude (3-DOF: roll, pitch, and yaw) and altitude (1-DOF: vertical position). Alternatively, testbeds with 3-DOF (roll, pitch, and yaw) or 1-DOF (roll or pitch or yaw) can be used to evaluate and validate the quadrotor's behaviour in specific flight scenarios. By isolating specific aspects of flight, engineers can gain insights into how the quadrotor behaves in different flight regimes and optimize the controller accordingly.

It's worth noting that the development of a quadrotor testbench is a complex process that requires expertise in control systems, robotics, and aerospace engineering. The testbed must be carefully designed to ensure accurate and repeatable results while maintaining safety protocols. Furthermore, the testbench should be versatile enough to accommodate various configurations and modifications, allowing for iterative testing and continuous improvement of the quadrotor controller design.

NB: *Our testing platform is currently undergoing the patenting process. This suggests that the testbed incorporates innovative features or technologies that are potentially novel and worthy of intellectual property protection.*

Overall, a well-designed and comprehensive quadrotor testbench plays a crucial role in the iterative design and improvement of quadrotor controllers. Through rigorous testing and analysis, engineers can enhance the quadrotor's performance, safety, and reliability, paving the way for advancements in various applications.

2.5 Conclusion

This chapter has provided a comprehensive overview of the hardware and software components that make up a typical quadrotor UAV system. We have discussed the role of the flight controller and the computer, as well as the fundamental sensors used in UAV systems such as the IMU, barometer, magnetometer, and GPS. We have also examined the propulsion chain actuators, including the propeller, motor, ESC, and battery, and discussed the various UAV communication techniques and protocols. Finally, we have highlighted the importance of open-source autopilot software and hardware in the development of quadrotor technology. These open-source technologies have enabled the creation of low-cost and easily accessible quadrotors for researchers and educators to perform high-level control experiments and educational activities. Overall, this chapter has provided a solid foundation for understanding the hardware and software components of quadrotor systems, and has demonstrated the importance of open-source technologies in driving innovation in the field of UAVs.

CHAPTER 3

SENSOR CALIBRATION AND FILTERING

Contents **Page**

3.1 Introduction

This chapter focuses on the crucial aspect of quadrotor sensors calibration and sensor filtering techniques. Firstly, we delve into the importance of accurate sensor calibration, including the explanation of sensor deterministic errors and the sensor error models. The calibration of the IMU sensor, the compass. Additionally, the barometer is discussed in detail, as well as the integration of these sensors with the GPS and vision navigation systems. Secondly, the chapter explores the various sensor filtering techniques, including the Kalman filter and extended Kalman filter. We examine the implementation of a complementary filter, which combines a low-pass and a high-pass filter to reduce the noise in the IMU accelerometer and gyro sensors and overcome the limitations of these sensors. Finally, the chapter focuses on the impact of vibrations on the stability of quadrotors and the solutions to these problems, including dynamic and static balancing of the motors and propellers. The aim of this chapter is to provide a comprehensive overview of the techniques used to improve the accuracy and reliability of the sensors used in quadrotor systems, and to highlight the importance of accurate sensor calibration and filtering in achieving stable and safe flight.

3.2 Sensors Calibration Overview

A quadrotor is equipped with several sensors, including a three-axis gyroscope, accelerometer, barometer [35] and magnetometer [81-87], GPS [90], and in some applications, ultrasonic and range finders, and a camera. These sensors allow a quadrotor to determine its location and attitude, altitude orientation, and speed. Inaccurate measurement by this sensor leads to unstable flying; in other words, precision measurement implies proper sensor calibration [70].

In this section, the most important works in sensor calibration will be discussed, as well as an overview of sensor errors and noise. Figure 3.1 displays errors and noises related to sensors that can impact the accuracy of measurements.

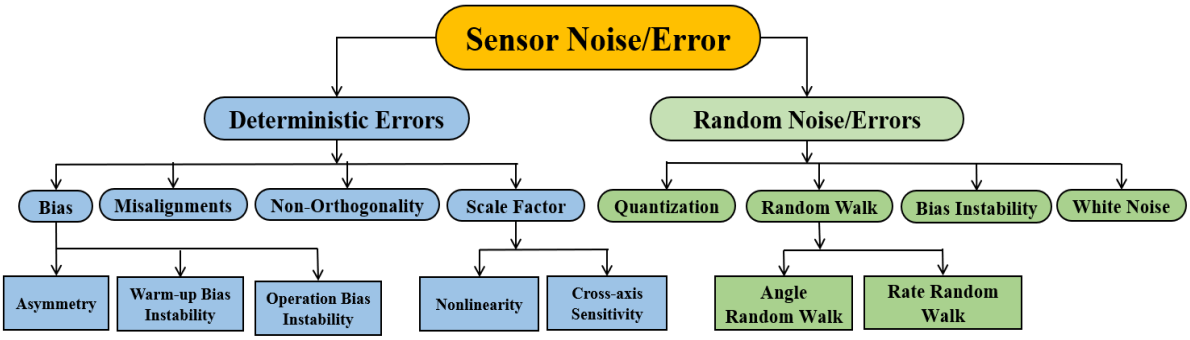


Figure 3.1: Classification of Sensor Noises and Errors

3.2.1 Sensor Deterministic Errors

Sensor deterministic errors encompass a range of inaccuracies and biases in sensor measurements that exhibit consistent and predictable patterns [70]. Some specific examples of sensor deterministic errors include:

3.2.1.1 Sensor Zero-Bias Errors:

Optimal sensor output measurement should be null when the measured physical quantity is zero. In reality, the output is typically not zero, this is termed zero-bias error [11]. A zero-bias error, sometimes called zero-position error, may be split into asymmetry error, warm-up zero-bias instability and operational zero-bias instability. This kind of error often resulting from a manufacturing defect. Figure 3.2 depicts the sensor zero-bias errors.

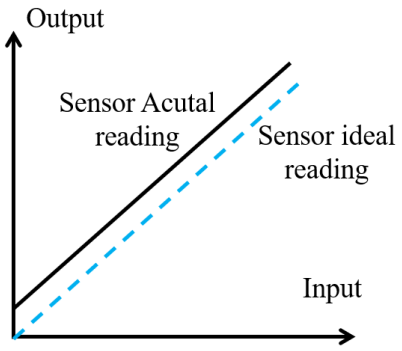


Figure 3.2: Sensor Zero-bias Error

3.2.1.2 Sensor Scale Factor Error

The scale factor, also known as the proportionality factor or sensitivity fault, can be mathematically defined as the difference between the variation in the sensor output and the corresponding change in its input. Ideally, every axis's scale factor should be identical, meaning that input changes and output changes should match perfectly. However, in reality, there may be variations in scale factors. Figure 3.3 illustrates scale factor errors.

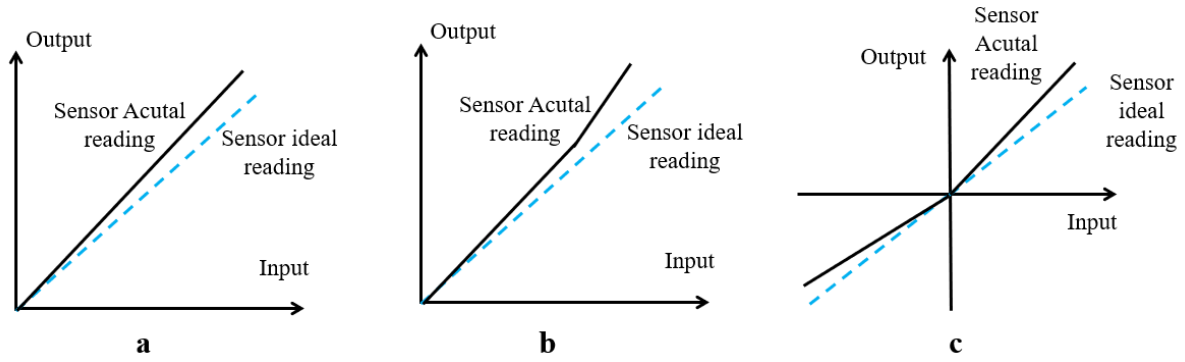


Figure 3.3: Sensor Scale factor: (a) linear errors. (b) nonlinear errors. (c) asymmetry errors

3.2.1.3 Cross-Axis Sensitivity

The cross-axis influence is associated with non-orthogonality and the cross effect among distinct sensor channels. Non-orthogonality error refers to the error resulting from the faulty assembly of IMU chip or the intrinsic non-orthogonality resulting from the fact that the sensitivity axes in the sensors cannot be built correctly owing to real manufacturing limits.

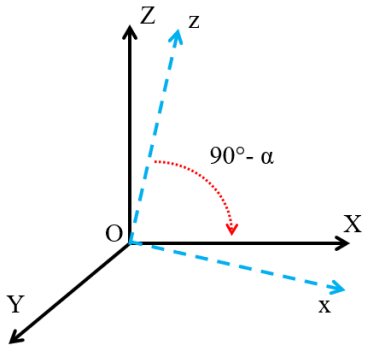


Figure 3.4: Sensor Cross-Axis Sensitivity

3.2.1.4 Misalignment Error

Theoretically, all the orthogonal sensitive axis element of the sensor must be aligned with the host of sensor orthogonal axes, each both two axes must be matched and there is no angle between them. In reality, however, this is not guaranteed, and as a result, there is a mounting error in the angle between the two (sensitive element of the sensor and its host), leading to misalignment inaccuracies in the sensor's measurements. Figure 3.5 depicts the misalignment diagram.

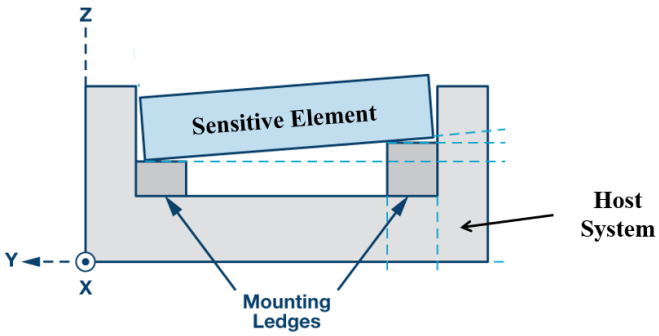


Figure 3.5: Sensor Misalignment Error

3.2.2 Inertial Measurement Unit Sensors Calibration

The inertial measurement unit is a platform with multi-sensor that measures and reports the force of gravity and the angular speed of an associated object. It composed of three accelerometers sensing vibration by help of piezoelectric material, and three gyroscopes use Coriolis effect in double-T structure element and drive arm rotates. In certain situations, three magnetometers, for each axis of roll, pitch, and yaw. many scopes in the area of inertial sensor calibration have been treated by many kinds of literature [69-71]. Static multi-position method based on the direction of gravity and magnetic field of the earth was proposed by [72]. It is unsuitable for high-performance IMUs [72-74]. The least squares method (LSM) is used to estimate the error parameters, where LSM has been used experimentally by [75-76] which demonstrate its feasibility when using a high accuracy equipment (turntable) which provides an accurate control to the orientation during the hole calibration processes, making it accurate and simple to implement. However, a high-precision turntable is not a cost-effective solution due to its high price [72][74]. Unfortunately, the high-precision turntable is not reachable by all searchers because of its high price. Kalman filter and extended Kalman filter have been also widely used for sensor calibration [76-77].

3.2.2.1 IMU Error Model Sensor and Calibration

A) IMU Sensor *Error Model*

Sensor Error Models (SEM) is a depiction of sensors, including uncalibrated primary-element measurement and its offset, misaligned axes and measurement scale factors, and in certain sensors, the magnetization effect, shown as hard iron and soft iron. The IMU calibration process needs pre-determination SEM for both Gyro and ACC. They come in a variety of representations with the same parameters, the only variable in this context is the calibration procedure [78-80]. The SEM in equation 3.1 and 3.2, are comprised of nine unknown values each, including three offsets, three correction scale factors, and three non-orthogonal angles.

$$ACC_f = T_{no} S_f (ACC_n - b) = \begin{bmatrix} ACC_{fx} \\ ACC_{fy} \\ ACC_{fz} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ T_{xy} & 1 & 0 \\ T_{xz} & T_{yz} & 1 \end{bmatrix} \begin{bmatrix} s_{fx} & 0 & 0 \\ 0 & s_{fy} & 0 \\ 0 & 0 & s_{fz} \end{bmatrix} \left(\begin{bmatrix} acc_{nx} \\ acc_{ny} \\ acc_{nz} \end{bmatrix} - \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) \quad (3.1)$$

Where:

$\boldsymbol{ACC_n}$ is the measured acceleration vector.

$\boldsymbol{ACC_f}$ is the vector of compensated ACC final readings.

$\boldsymbol{S_f}$ is the matrix of scale factors

$\boldsymbol{T_{no}}$ is matrix of non-orthogonality, transformation from the non-orthogonal to orthogonal frame.

$\boldsymbol{b}$ is vector corresponding to sensor offset.

The gyro SEM is represented by the equation 3.2:

$$\boldsymbol{Y_g - B_g = T_g M_g S_g U_g} \Rightarrow \begin{bmatrix} Y_{gx} - b_{gx} \\ Y_{gy} - b_{gy} \\ Y_{gz} - b_{gz} \end{bmatrix} = \begin{bmatrix} c_\theta c_\psi & -c_\varphi s_\psi + s_\varphi s_\theta c_\psi & s_\varphi s_\psi + c_\varphi s_\theta c_\psi \\ c_\theta s_\psi & c_\varphi c_\psi + s_\varphi s_\theta s_\psi & -s_\varphi c_\psi + c_\varphi s_\theta s_\psi \\ -s_\theta & s_\varphi c_\theta & c_\varphi c_\theta \end{bmatrix}^T *$$

$$\begin{bmatrix} 1 & 0 & 0 \\ \alpha_{xy} & 1 & 0 \\ \alpha_{xz} & \alpha_{yz} & 1 \end{bmatrix} \begin{bmatrix} S_{gx} & 0 & 0 \\ 0 & S_{gy} & 0 \\ 0 & 0 & S_{gz} \end{bmatrix} \begin{bmatrix} u_{gx} \\ u_{gy} \\ u_{gz} \end{bmatrix}. \quad (3.2)$$

Y_g is measured vector of the angular rates for the gyro undergoing calibration, B_g is offsets vector that prior-determined, S_g is matrix of scale factors, T_g is matrix represent the transformation from the non-orthogonal to orthogonal frame, M_g is the

misalignment matrix between gyro and the referential frames Ug is vector of measured angular rates. c_i=cos i, s_i= $sin\, i$ and θ, ψ, φ represent Euler angles.

B) Calibration Process

The static approach is utilized as an example of one of these common approaches in the calibration field. The calibration of the ACC is based on the ACC data set gathered at different orientations in static conditions with only gravitational impact. It is advised to collect the outputs of ACC at least in 21 orientations according to the Thin-Shell approach [81] and the reliability of this technique analysis has been reported in [82]. Moreover, to reduce the effect of temperature drift, it is recommended to warm up the ACC before starting the calibration process, in which it should hold the ACC in each orientation in static conditions and give some time for raw data averaging, which would minimize the impact of random noise. All averages have been conducted along all axes x, y, and z, with at least eight orientations along each axis Figure 3.6, where exact knowledge of these orientations is not necessary [82]. Using the minimization function approach to estimate the nine unknown coefficient of SEM, as a criteria function for minimization, the Root Mean Square Error (RMSE) presented in equation 3.4 has been utilized. This calibration technique assumes that in a static condition where there is no movement the ACC-measured output should equal acceleration of gravity.

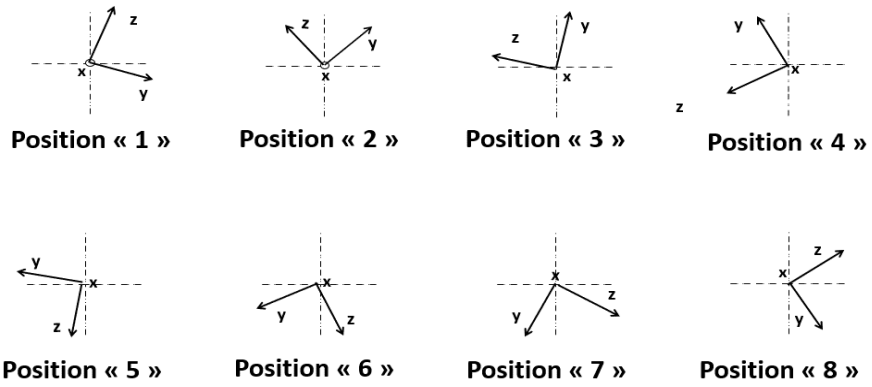


Figure 3.6: ACC calibration around eight orientations along "x" axis

In static conditions, the only force impacting the accelerometer is gravitational forces in the downward direction; this fact results in equation 3.3.

$$g_x^2 + g_y^2 + g_z^2 = |g|^2 \qquad (3.3)$$

Where g_i represents the projection of the gravity vector onto axes i. As a criterion function for minimization, RMSE is used.

$$RMSE(\Theta, g) = \sqrt{\frac{\sum_1^N (|a_i(\Theta)| - g)^2}{N}}, \quad N=8, \tag{3.4}$$

$$\text{Where} \quad |a_i(\Theta)| = \sqrt{acc_x{}^2 + acc_y{}^2 + acc_z{}^2} \tag{3.5}$$

And $\Theta = (\Theta_1, \Theta_2, \dots, \Theta_9)$ is SEM nine coefficient being estimated, the total number of orientations about all axis is N (in our case N=8), m=24 in our case, g=1 is gravity vector, $a_i(\Theta)$ represent the SEM acceleration vector being estimated.

3.2.3 Compass Calibration

Compass sensors, also known as magnetometers, are magnetic sensors that use resistance to detect changes in the strength or direction of a magnetic field. Non-contact rotation or position detection is possible when used in combination with a magnet. There are two distinct error categories for magnetic sensors. The instrumentation errors are categorized under the first heading. The sensor offsets, the scale factor, and the sensor axes' non-orthogonality are all included. The fabricator's restrictions are the cause of these sensor errors. A magnetic divergence produced by host platform hardware onboard falls under the second category, which is dependent only by the kind of magnetometer. Sensor's magnetometer tri-axis sensor, and instrumentation errors can be regarded as constants [80-81] and [83-85]. The compass has been used for determining direction for many years and is still widely used in all aspects of daily life. The issue of compass calibration has been the subject of numerous studies. Astronomer Nathaniel Bowditch published a Dedicated Guide to Sky Navigation [86], which includes a swinging calibration technique that requires the device to be levelled and rotated in a series of known declinations [83]. By utilizing the maximum and minimum values of the measurements obtained during the sensor's rotation in the horizontal plane, it was possible to accurately predict the scale factors and compass biases. While this method is practical, it overlooks certain sensor errors. A more comprehensive approach has been discussed in [84], where a total sensor calibration employs a Least Squares (LS) iterative batch algorithm to determine scale factors and sensor deviations, where the initial conditions are given by a LS nonlinear two step. This method is assigned to those sensors that sense a constant linear

field. The assumptions behind this calibration's limitations are that misalignments can be ignored and that measurements on the sensor's axis that are parallel to the induced magnetic field are the only ones that are negatively impacted by the soft iron. An iterative geometric method based on the Maximum Likelihood Estimator (MLE) is employed in [85]. MLE, in this context, stands for magnetometer measurement ellipsoidal manifold. The misalignment matrix is computed using an additional closed-form optimal algorithm.

3.2.3.1 Magnetometer Errors model and Calibration Process

Magnetometers are vital sensors in UAVs, but they suffer from errors like hard and soft iron disturbances, temperature effects, and sensor noise. This section explores magnetometer error models and calibration processes to enhance UAV accuracy, by delve into mathematical models that describe these errors, and the calibration techniques such as hard iron and soft iron correction, temperature compensation are discussed.

The following equation can be thought of as the SEM for magnetometers:

$$\widehat{\mathbf{H}} = \mathbf{SM}(\mathbf{A}_s\mathbf{h} + \mathbf{b}_h) + \mathbf{b}_s + \boldsymbol{E} \tag{3.6}$$

Become:

$$\widehat{\mathbf{H}} = \mathbf{Ah} + \mathbf{b} + \boldsymbol{E} \tag{3.7}$$

$\mathbf{A} = \mathrm{S\ M\ A_s}$ Soft iron $\mathrm{A_s}$ and non-orthogonality matrix M and scale factor matrix S. $\mathbf{b} = \mathrm{S\ M\ b_h + b_s}$ is a combined biases includes Hard Iron $\mathrm{b_h}$ and the sensor offset $\mathrm{b_s}$ $\mathbf{S} = \mathrm{diag}(s_x \quad s_y \quad s_z)$ is scale factor matrix, $(s_x \quad s_y \quad s_z)$ are ratio constants of between input and output.

$\mathbf{M} = \mathrm{N}^{-1} = [e_x \quad e_y \quad e_z]^{-1}$, M matrix utilized to rectify the effects of non-orthogonality, such as misalignment errors, and $e_x \quad e_y \quad e_z$ are misalignment errors provide the sensor direction of such x, y, and z axis. N is a matrix in which the orientation of each sensor axis is indicated by N column vectors.

Where $h^T = [h_x \quad h_y \quad h_z]$ is the sensor magnetic field error-free.

$\hat{H}^T = [\hat{H}_x \quad \hat{H}_y \quad \hat{H}_z]$ Are the magnetometers, sensor reading. Misalignments, soft iron errors, and the scale factors, are all combined in matrix A (3x3). the bias when combined b is the total sensor bias. E represents Gaussian white noise.

Bias, even when there is no movement, there is often a slight offset in the signal output represented by the term "bias b_s".$b_s = [b_{sx} \quad b_{sy} \quad b_{sz}]^T$

Hard Iron, b_h It is a bias caused by permanent magnets or the magnetization of iron materials that are already magnetized. $b_h = [b_{hx} \quad b_{hy} \quad b_{hz}]^T$

Soft iron, ferromagnetic materials interact with an external field to produce magnetism b_s, This modifies both the intensity and direction of the sensed field. The soft iron effect can be represented as matrix A_s which is (3x3) [87].

$$A_s = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{32} & a_{33} \end{bmatrix}$$

In an environment devoid of perturbations, the standard deviation of the magnetometer's vector measurement should match the earth's magnetic field. Rotating the sensor in space should result in the description of a sphere with a radius equivalent to the Earth's magnetic field [87][84].

The next equation constrains the readings of an ideal magnetometer in surroundings devoid of perturbations:

$$\boldsymbol{H_m^2 - \| \mathbf{h} \|^2 = H_m^2 - \mathbf{h}^T\mathbf{h} = 0} \qquad (3.8)$$

Where H_m represents the geomagnetic field model derived from the International Geomagnetic Reference Field (IGRF), a dataset provided by the International Association of Geomagnetism and Aeronomy (IAGA) [88].

By rewriting equation 3.6 as follows:

$$\mathbf{h} = \mathbf{A}^{-1}(\hat{\mathbf{h}} - \mathbf{b} - \boldsymbol{\varepsilon}). \tag{3.9}$$

Substituting h in equation 3.9 in 3.6 to get

$$(\hat{\mathbf{h}} - \mathbf{b})^{\mathrm{T}}(\mathbf{A}^{-1})^{\mathrm{T}}\mathbf{A}^{-1}(\hat{\mathbf{h}} - \mathbf{b}) - H_m^2 = 0. \tag{3.10}$$

$$\text{suppose} \quad \mathbf{Q} = (\mathbf{A}^{-1})^{\mathrm{T}}\mathbf{A}^{-1}, \tag{3.11}$$

Then equation 3.10 become

$$(\hat{\mathbf{h}} - \mathbf{b})^{\mathrm{T}}\mathbf{Q}(\hat{\mathbf{h}} - \mathbf{b}) = H_m^2 \tag{2.12}$$

Rewriting equation 3.12 to be

$$\hat{\mathbf{h}}^{T}\mathbf{Q}\hat{\mathbf{h}} + (-2\mathbf{Q}^{\mathrm{T}}\mathbf{b})^{T}\hat{\mathbf{h}} + \mathbf{b}^{\mathrm{T}}\mathbf{Q}\mathbf{b} - H_m^2 = 0 \tag{3.13}$$

Suppose $\mathbf{u} = -2\mathbf{Q}^{\mathrm{T}}\mathbf{b}$ and $k = \mathbf{b}^{\mathrm{T}}\mathbf{Q}\mathbf{b} - H_m^2$. Then equation 3.13 become

$$\hat{\mathbf{h}}^{T}\mathbf{Q}\hat{\mathbf{h}} + \mathbf{u}^{T}\hat{\mathbf{h}} + \boldsymbol{k} = \mathbf{0} \tag{3.14}$$

Since the magnetic field of the Earth is positive, consequently, the general formula of an ellipsoid is in equation 3.14, [89] the calibration process is the estimation of unknown parameters in equation 3.14. Along numerous orientations that best characterize the ellipsoid, it is feasible to calibrate magnetometers using the restriction on the norm of the field vector equation 3.14.

The calibration algorithm presented in this paper draws inspiration from the research conducted in [87] and consists of main two steps. Where the first step involves locating the parameters “u" and "k” by finding of "b" and "Q" of the equation 3.14 which a fitting problem involving the fitting of an ellipsoid to a magnetic field measurement collected in random paths or along different orientations. The last step is the calibration of "b" and "A".

Numerous approaches have been devised. By using the least squares method, algebraic fitting techniques attempt to solve optimization issues. In this study, however, the calibration method employs an adaptive least squares estimation to resolve the ellipsoid-fitting issue, which manages the measurements' excessive noise, the details of the used calibration process are in [87].

3.2.4 Barometer Sensor

A barometer is the device to measure the environment's absolute pressure, where the system altitude is estimated from the pressure variations. This study [35] has demonstrated that at altitudes lower than 70 meters, provide the best performances for altimeters of barometric, and due to high level of noise, GPS data utilized for barometer calibration. A polynomial approach has been presented in [36] for determining the optimal calibration for barometric MEMS pressure sensors after they have been calibrated under a range of pressures and temperatures. Pressure sensors are frequently built from piezoelectric materials, which are sensitive to deformation and produce an electric field when being subjected to deformation forces. The pressure barometric sensors give the air pressure values by which the altitude can be calculated using formula in [35].

$$P_h = P_{h_0} * e^{-(\frac{gM}{RT}(h-h_0))} \tag{3.15}$$

Where g = 9.80665 m/s2 is the earth gravitational.

M = 0.0289644 kg/mol is the air molar mass.

R = 8.31432 $JK^{-1}mol^{-1}$ is the gas constant and at T = 273,16° C is Absolute temperature. h is the quadrotor altitude. P_h is the pressure of air of the quadrotor at altitude h. P_{h_0} is pressure taken at the ground h_0. The altitude equation can be succinctly expressed as follows short formula:

$$h = 8453.669 \ln(P_{h_0}/P_h) \tag{3.16}$$

It's important to note that this formula is an approximation and relies on several assumptions, such as a constant temperature lapse rate and no influence of factors like temperature inversions or weather variations. Additionally, local variations in atmospheric conditions can introduce errors in altitude estimation using this formula. To obtain accurate altitude measurements, it is recommended to use calibrated barometric sensors and consider factors such as temperature, humidity, and local weather conditions. Additionally, altitude determination may be improved by incorporating other sensor data, such as GPS or altitude from ground-based reference points.

3.2.5 GPS-IMU-Vision Navigation

For location estimation, an IMU-based localization system employs gyroscope, accelerometer, and magnetometer sensors. This type of localization relies on the quantity of sensor errors generated by external electromagnetic noise or sensor drifts induced by time and distance travelled. Despite efforts to increase the accuracy of IMU-based localization across great distances, it remains limited. Therefore, GPS-based localization, outside localization, or hybrid techniques must be used. The Global Positioning System is a satellite-based navigation system that delivers location and time information where there is a clear line of sight to four or more GPS satellites. When tracking is the primary objective of a navigation system, GPS is employed. GPS survey navigation for UAVs in denied environments is discussed in [90]. Multi-GNSS is utilized in numerous types of literature:

Relative location [91] have been addressed in several attempts to combine GPS and GLONASS satellite systems using Pseudo-range. other adopts the modernism of Relative-GPS approaches when the goal is to find an unknown point's coordinates relative to a known point.

IMU/GPS navigation: The combination of GPS and IMU gyroscopes, accelerometers, and magnetometers has been widely used for position tracking. UKF, Particle Filters, and Sigma Filters are used too for increased precision and complexity, while non-linearity is managed by Extended Kalman Filters [92].

Visual Navigation: In the field of UAV navigation, the use of visual sensors for localization has shown to be very advantageous due to the effectiveness of vision approaches. estimating attitude angles by following the horizon [91]. Survey of UAV navigation based on vision [93], as vision-based IMU sensors supplements, optic flow, stereo-vision and feature tracking have been suggested too.

3.3 Sensor Filtering

Accurate estimating a UAV's altitude and attitude is a crucial control challenge. Because of the fast timescales involved with the angle stabilization loop, deterioration of altitude and attitude estimations may rapidly result in system instability and complete collapse. The stability of the estimator augments the stability of the system. Recently, the emphasis on innovative low-cost aerial vehicles has generated a great

deal of interest in state estimation and low-cost sensor signal filtering. The primary purpose of signal filtering is to decrease and smooth out measurement-related high-frequency noise. The noise and time are uncorrelated, the value of the noise at any given moment does not depend on its past values. For state estimation, filtering algorithms try to minimize the difference between the estimated state variables and their actual values. The optimal Bayesian filtering offers a theoretical solution to the stochastic system's overall state estimation problem. Based on computational approximations of the concept of a Bayesian solution, numerous Bayesian filtering methods have been created. In some special cases, the optimal Bayesian solutions for linear systems with Gaussian noise are provided by the Kalman filter (KF). To expand Kalman filter utilization for those nonlinear systems, two basic suboptimal methods have been suggested to approximate the nonlinear functions. Probability distributions and nonlinear functions are approximated by using power series; other variations of filters are used to approximate the nonlinear functions, e.g., the extended Kalman filter, the divided-difference filter, and the extended information filter (EIF) [94], the Cubature Kalman filter (CKF) [95], the particle filter (PDF) [96], and the unscented Kalman filter (UKF) [97]. For non-gaussian cases, use the gaussian-sum filter, where the non-Gaussian and nonlinear systems would be handled by the particle filter. System uncertainties are well handled using variable-structure filters (VSFs), which are based on a stability theorem to prove their stability. Learning-based filtering [98] suggests a rich literature on learning algorithms and deep learning architectures. Attitude estimation of a quadrotor is experimentally validated in [99] and simulated in [100]. While unscented Kalman filtering based on ultra-wide band (UWB) and IMU data fusion for quadrotor indoor navigation is experimentally validated in [101]. The Cubature filter used in [102] attitude estimation using low-cost sensors based on a particle filter subjected to uncertainties is treated in [103]. In [104] an experimental comparison of Kalman and the complementary filter is performed for attitude estimation. Conventional linear Kalman filter approaches, such as EKF techniques, have proven particularly challenging to implement reliably in applications using sensor systems of poor quality [105]. Due to the intrinsic nonlinearity of the system and the non-Gaussian noise experienced in reality, such filters might exhibit extremely poor performance. Many searches in the UAV field use the complimentary filter technique to estimate orientation states [106]. This kind of filter is widely used in quadrotor projects and proves robust and easy to tune and implement in no powerful calculators. The traditional use of this filter is for a single input integration [107].

The most used filter algorithms in UAV are Kalman filter, extended Kalman filter, and Complementary Filtering.

3.3.1 Kalman Filter

The Kalman Filter is a recursive algorithm used for estimation and prediction of the state of a linear dynamic system, given noisy measurements. It combines measurements from sensors with predictions from a mathematical model to provide an optimal estimate of the system's state.

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ y_k &= Cx_k + v_k \end{aligned} \tag{3.17}$$

$w_k \; and \; v_k$ represent system disturbance and sensor noise respectively, $\hat{x}_{k+1}$ represents estimated states and y_k is the measurement output. The estimated state $\hat{x}_{k+1}$:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K_k[y_k - C\hat{x}_k] \tag{3.18}$$

Where:

$$\begin{aligned} K_k &= P_k C^T (CP_k C^T + R)^{-1} \\ P_{k+1} &= A(I - K_k C)P_k A^T + Q \end{aligned} \tag{3.19}$$

K_k is a Kalman-filter gain, and Q is the covariance of the noise process w_k, where R is the covariance of the noise in the measurement v_k, and $I =$ identity matrix.

The Kalman Filter is widely used in quadrotor systems for state estimation, such as estimating the quadrotor's position, velocity, attitude, and angular rates. It fuses measurements from various sensors, such as accelerometers, gyroscopes, magnetometers, GPS, and barometers, to provide a more accurate and reliable estimation of the quadrotor's state. The filter accounts for measurement noise, process noise, and the dynamics of the quadrotor system, allowing it to handle uncertainties and provide robust estimation even in the presence of sensor noise and external disturbances. By continuously updating the state estimate based on the prediction and measurement update steps, the Kalman Filter enables real-time tracking of the quadrotor's state variables, which is crucial for tasks such as autonomous navigation, control, and localization.

It's worth mentioning that extensions and variations of the Kalman Filter, such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF), are also used in quadrotor systems to handle nonlinear dynamics and improve estimation accuracy in situations where linear models are inadequate. These variants make use of linearization or non-linear transformations to adapt the Kalman Filter framework to non-linear systems.

3.3.2 Extended Kalman Filter

The Extended Kalman Filter (EKF) is an extension of the Kalman Filter that allows for estimation and prediction of states in nonlinear systems. It addresses the limitation of the standard Kalman Filter, which is designed for linear systems. In the EKF, the nonlinear system dynamics are approximated by linearizing them around the current state estimate. This linearization is done using first-order Taylor series expansion. By linearizing the system dynamics, the EKF can use the same prediction and update steps as the standard Kalman Filter, albeit with modified matrices to account for the nonlinearity. The state equation is denoted as:

$$\begin{aligned} x_{k+1} &= f(x_{k-1}, u_{k-1}) + w_{k-1} \\ z_k &= h(x_k) + v_k \end{aligned} \tag{3.20}$$

w_k *and* v_k represent system disturbance and sensor noise respectively, both are assumed to be zero-mean multivariate Gaussian noises, with covariance matrices Q_k and R_k respectively.

Predict:

Predicted state estimate: $\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1})$

Predicted covariance estimate: $\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^T + \boldsymbol{Q}_{k-1}$ (3.21)

Update

Innovation (or residual) covariance: $\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$

Innovation or measurement residual: $\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k$

Near-optimal Kalman gain: $\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T \boldsymbol{S}_k^{-1}$

Updated state estimate: $\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k$

Updated covariance estimate: $\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1}$ (3.22)

The matrices representing state transition and observation are defined as the following Jacobians: $\boldsymbol{F}_k = \left.\frac{\partial f}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k}$ and $\boldsymbol{H}_k = \left.\frac{\partial h}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k|k-1}}$ (3.23)

In the context of quadrotors, the EKF is commonly used for state estimation and control in situations where the dynamics are nonlinear. Quadrotors have complex dynamics due to factors such as rotor aerodynamics, nonlinearities in control inputs, and environmental disturbances. The EKF allows for more accurate estimation of the quadrotor's position, velocity, attitude, and other states by handling the nonlinearities present in the system. By iteratively updating the state estimate using the prediction and measurement update steps, the EKF provides real-time estimation of the quadrotor's states, enabling robust control, navigation, and localization. It is an essential tool for autonomous flight, sensor fusion, and state estimation in quadrotor systems.

3.3.3 Complementary Filtering

Complementary filtering is a sensor fusion technique commonly used in quadrotor systems to combine the outputs of multiple sensors and obtain a more accurate estimate of the system's state. It is particularly effective in situations where there are complementary characteristics among the sensors used. The basic idea behind complementary filtering is to utilize the strengths of different sensors while compensating for their individual weaknesses. The complementary filter is a non-heavy computational and suitable for less powerful calculators (e.g., 8-bit microcontrollers). It is an approach for first order kinematic systems that combines a low-pass and a high-pass filter to reduce sensor noise and overcome sensor drawbacks, such as gyro drift, by complementing with other sensors' measurements.

In our application, the IMU-sensor-base attitude estimation uses multiple sources to estimate stable angles where gyroscope measurements are complementary to the accelerometer measurements [108-109][106]. The estimated state is extracted as following:

Suppose the integrator first order as

$$\dot{x} = U \tag{3.24}$$

Let suppose the measurement model for gyroscope and accelerometer as:

$$x_a = LPF(s)U_a + \mathcal{N}_a \tag{3.25}$$

$$\dot{x}_g = U_g \quad + \mathcal{N}_g + b \tag{3.26}$$

Where $LPF(s) \approx 1$ is low-pass filter equal to one near the frequency range where we pick the measurements $\mathcal{N}_a$, $\mathcal{N}_g$ are Acc and gyro sensor noise, b is bias associated with gyro. x_a complement $\dot{x}_g$ in order to estimate the state $\hat{x}$ through the filters as showed below in Figure 3.7.

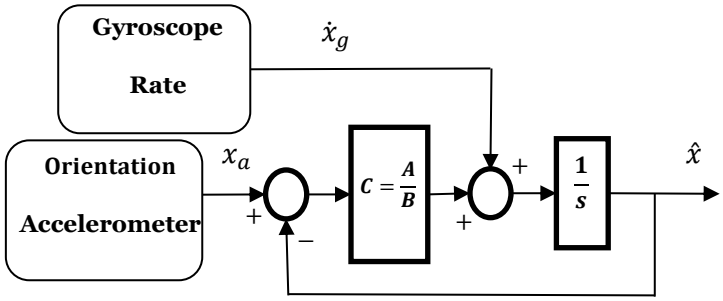


Figure 3.7: General form of simple complementary filtering

$$sB\hat{x} = Bx_g + A(\dot{x}_a \quad - \hat{x}) \tag{3.27}$$

If the compensator $C(s) = 1/\tau$ the equation 3.27 become:

$$\hat{x} = \frac{\tau s}{1+\tau s}\left(\frac{\dot{x}_g}{s}\right) + \frac{1}{1+\tau s}(x_a) \tag{3.28}$$

$$= S(s)\left(\frac{\dot{x}_g}{s}\right) + T(s)(x_a)$$

$S(s)$ is a high pass filter and it is the closed-loop sensitivity transfer function, $T(s)$ is complementary of sensitivity transfer function and it is low pass filter.

Where $S(s) + T(s) = 1$ (3.29)

The gyro measurement $\dot{x}_g$ is fed through the integrator to estimate the state after being filtered by a high pass filter S(s) to maintain only the high-frequency data, because it is unreasonable to integrate the gyroscope's low-frequency component noise and bias $\left(\frac{\dot{x}_g}{s}\right)$. τ (rad/s) is the crossover frequency derived from the trade-off between gyro and ACC measurements, taking into account the low-pass properties of the accelerometer and the noise properties of the gyroscope at its low-frequency. From bode diagram of $S(s)$ (HPF) and $T(s)$ (LPF) the sum their magnitude equals one at all rang of frequencies. The complementary filter used as following:

$$\hat{x} = (1 - \beta)x_a + \beta\left(\int \dot{x_g} dt\right) \quad (3.30)$$

where β is a weighing factor between zero and one.

$$\beta = \frac{\tau}{T_s + \tau} \quad (3.31)$$

Where T_s is the loop time (Sample Rate), τ is the complementary filter time constant (desired bandwidth). The aforementioned classical filter design is one of a variety of complementary filter designs, such as the complementary filter with or without bias compensation and the non-linear complementary filter, etc.

3.4 Vibration Analyses

3.4.1 Vibration Analysis in the Time Domain

Vibration analysis in the time domain is a technique used to analyze the behavior of mechanical systems by examining the characteristics of vibrations in the time-based waveform of a vibrating object or structure. Unlike frequency domain analysis, which decomposes signals into their frequency components, time domain analysis focuses on studying the vibration signal directly in the time axis. In time domain analysis, vibration signals are typically captured using sensors such as accelerometers or velocity transducers. These sensors measure the acceleration or velocity of the vibrating object at specific points, generating a time series of data points. By analyzing the time-domain waveform, various parameters and characteristics of the vibration can be determined. These include amplitude, frequency, duration, and shape of the waveform. The amplitude represents the magnitude of the vibration at a specific point in time, while the frequency indicates the rate at which the object is oscillating or vibrating. The duration helps determine the overall time span of the vibration event, and the waveform shape provides insights into the nature and potential causes of the vibration. Time domain analysis techniques include time waveform analysis, which involves examining the raw vibration signal in the time domain, and various statistical measures such as root mean square (RMS) analysis, peak analysis, crest factor analysis, and envelope analysis. Vibration monitoring often employs v-RMS (velocity root mean square) and a-RMS (acceleration root mean square) as commonly used measurements in the time domain to assess overall machine vibration. These measurements provide valuable insights into resonance, chattering, gear mesh wear, and other factors affecting machine performance [234]. Another technique employed in time domain

vibration analysis is Time Synchronous Averaging (TSA). TSA involves averaging the raw vibration signal in the time domain, ensuring sample rate or sample time synchronization. The classification of time domain vibration analysis techniques has been documented in references [235] and [236]. These techniques encompass various approaches, including statistical functions and advanced methods such as Time Synchronous Averaging (TSA), Time Series Regressive Model, Stochastic Parameter Technique, and Blind Source Separation (BBS). These methods offer valuable insights into the characteristics of vibration signals. Additionally, there are several commonly used equations in vibration analysis in the time domain that aid in the analysis and understanding of vibration behaviour. These techniques provide valuable information about the overall vibration levels, transient events, and the presence of impulsive or cyclical vibrations. Vibration analysis in the time domain is widely used in industries such as mechanical engineering, aerospace, automotive, manufacturing, and structural analysis. It plays a crucial role in condition monitoring, troubleshooting, and maintenance strategies to ensure optimal performance, reduce downtime, and prevent catastrophic failures in mechanical systems.

3.4.2 Vibration Analysis in The Frequency Domain

Vibration analysis in the frequency domain is a technique used to analyse the behaviour of mechanical systems by decomposing time-domain signals into their frequency components. This analysis is carried out using mathematical tools such as the Fourier transform, which converts a signal from the time domain into its frequency domain representation. By analysing a system's vibration characteristics in the frequency domain, it is possible to identify specific frequencies associated with machinery faults, diagnose problems, and design effective maintenance and repair programs. One common method of vibration analysis in the frequency domain is the Fast Fourier Transform (FFT) analysis. This method converts the time waveform of the vibration signal into the frequency domain, allowing for the determination of the frequency content of the vibration signal. This information can be used to identify the natural frequencies of the system and determine if any resonances are present.

FFT, is a powerful mathematical algorithm used in a wide range of applications, including signal and image processing, audio and speech analysis, and data compression. It is used to transform a signal from its time-domain representation to its frequency-domain representation. The FFT algorithm takes a set of time-domain

samples and decomposes them into the constituent frequency components that make up the signal. The FFT works by re-representing the original time-domain signal as a sum of complex sinusoidal functions, where each sinusoidal function corresponds to a specific frequency component in the signal. The result of the FFT is a set of complex coefficients that describe the amplitude and phase of each frequency component in the signal. The FFT is defined by the following equation 3.32:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \tag{3.32}$$

where $X(k)$ is the frequency-domain representation of the signal, $x(n)$ is the time-domain signal, N is the number of samples in the time-domain signal, k is the frequency bin index, and $(e^{-j2\pi kn/N})$ is the complex exponential term. Note that these equations are defined for discrete-time signals, and the FFT and inverse FFT algorithms can be implemented using a variety of numerical methods, including the radix-2 FFT, the radix-4 FFT, and the prime factor FFT. The choice of algorithm depends on the specific requirements of the application and the computational resources available.

3.4.3 Vibration Analysis for Quadrotor

The ACC is being used to estimate position and attitude. The acceleration data is integrated with the GPS and barometer, which is extremely important. However, any vibration would have an immediate impact on the quadrotor's stability. When a propeller spins at a given speed, it produces a downward force, which can cause vibration displacement in the vertical direction. The elasticity of the materials used, along with the imbalance in mass distribution on each blade, leads to vibrational displacement. Because of the centrifugal force created by propeller spinning, the force exerted by each blade must be identical [110]. Typically, there is very minor excess mass that is not detected during static balancing, the effect of this extra will manifest at high speeds by generating vibrations during propeller rotation, The imbalanced spinning motion issues have been extensively examined in the literature. many research has been conducted on dynamic balancing [111] and nonlinear dynamics of imbalanced rotors in vertical and horizontal rotation axes [112-114]. A contact-less vibrometer laser used in [115] to measure the vibration velocity caused by the propeller rotation, where the act of balancing lowered vibration [110], in order to decrease vibration, a study [115] has examines the dynamics of blade movements at its maximum velocity, which corresponds to 118 Hz. With the use of an accelerometer

sensor and the FFT function, the vibration intensity of the motor-prop was lowered significantly in this study using two kinds of balancing, static and dynamic, where acceleration in the third degree of freedom is utilized as a measure of vibration strength according to the following formula:

$$ACC_{xyz} = \sqrt{acc_x{}^2 + acc_y{}^2 + acc_z{}^2} \tag{3.33}$$

ACC_{xyz} Is the strength of the acceleration and acc_x , acc_y, acc_z are the accelerations with respect to the axes x, y, z. The analysis of the strength of mechanical vibrations through monitoring their signal patterns over time is a fundamental practice in condition monitoring and fault diagnosis of motors and propeller. This analysis typically involves two main domains, the time and frequency domain. In the time domain, the vibration strength is analysed by monitoring the signal strength and patterns of vibration over time. In the frequency domain, the signal is decomposed into its constituent frequencies, which allows the identification of specific frequencies associated with motors/propellers faults. The experimental section describes the particulars of this procedure.

3.5 Conclusion

The focus placed on UAVs prompted us to develop our scientific methodology based on in-depth studies of UAV design and control. We paid close attention to the calibration of sensors and filtering methods to ensure accurate sensor readings, which led to effective aircraft control. This was accomplished by calibrating the most important sensors, such as the IMU and magnetometer, after a lengthy discussion of the most important calibration techniques and filtering methods. In this last, we touched in depth on complementary filters and how to select the frequency band for each high-pass and low-pass filter based on quadrotor attitude bandwidth. And because vibration is the greatest threat to aircraft stability, research was conducted to determine how to reduce it scientifically and effectively, the use of vibration analysis techniques in the time and frequency domains was presented, specifically in the context of quadrotor UAVs.

CHAPTER 4

DYNAMIC MODELLING AND CONTROL

Contents **Page**

4.1 Introduction

This chapter provides an in-depth examination of quadrotor dynamic modeling, identification, and control. The first section focuses on the dynamic modeling of quadrotors, with a comprehensive overview of the aerodynamic effects and the various forces acting on these quadrotors. The modeling approaches discussed in this section include Euler-Lagrange, Newton-Euler, and Quaternion methods. The second section covers dynamic model identification, providing an overview of the state-of-the-art quadrotor modelling and identification techniques, including white-box, grey-box, and black-box approaches. The process and steps of system identification are explained in both closed-loop and open-loop configurations, using direct and indirect approaches. The key aspects of dynamic model identification, such as excitation input selection, data preparation, model architecture and structure, order selection and input delay, and model validation, is also discussed in detail. The final section of this chapter explores quadrotor control. The classification of UAV controllers and their advantages and disadvantages are presented, followed by a general quadrotor control diagram. The linear control approaches, including PID controllers with fixed gains and scheduling and the Linear Quadratic Regulator (LQR) are discussed. In the non-linear control section, the focus is on the sliding mode technique, with a particular emphasis on the adaptive sliding mode controller and the second-order sliding-mode super-twisting observer.

4.2 Quadrotor Dynamic Modelling

4.2.1 Modelling Overview

Modelling UAVs, specifically quadrotor dynamics, is a challenging endeavor due to several factors that contribute to their complexity. These factors involve the determination of aerodynamic effects, the forces acting on quadrotors, and various considerations associated with quadrotor modelling. Futher illustrating the intricacies of this task, the assumptions, effects, and forces presented in reference [155] are considered:

- **Gyroscopic effect:** The tendency of a rotating quadrotor's angular momentum to resist changes in orientation.

- **Ground effect:** The change in aerodynamic forces experienced by a quadrotor when it is close to the ground, resulting in increased lift and decreased drag.
- **Drag and lift force of propeller rotation:** The forces generated by the rotation of the quadrotor's propellers, which include drag acting against the motion and lift supporting the vehicle.
- **Structure:** The physical framework and components that make up the quadrotor, affecting its overall stability and performance.
- **Propeller shaft:** The rotating shaft that connects the motor to the propeller, transmitting power and torque.
- **Friction force and moment:** The resistance encountered by the quadrotor due to internal and external factors, leading to losses in motion and rotational energy.
- **Coincidence of center of mass (COM) and coordinate system origin:** The assumption that the quadrotor's center of mass is aligned with the coordinate system's origin, simplifying calculations.
- **Blade flapping:** The motion of the rotor blades in response to aerodynamic forces, causing changes in their pitch angle.
- **Critical conditions in taking-off and landing:** Specific situations during takeoff and landing where the quadrotor's dynamics may be more challenging due to factors such as reduced airspeed and proximity to the ground.
- **Wind:** The effect of air currents on the quadrotor's motion and stability.
- **Aerodynamic Effects:** Various phenomena related to the interaction between the quadrotor and the surrounding air, including lift, drag, and other forces influencing its flight characteristics.

The dynamics of a quadrotor can be modelled using different mathematical frameworks, such as Euler-Lagrange and Newton-Euler formulations, as well as the Quaternion approach. These approaches provide distinct mathematical frameworks for analyzing and understanding the dynamics of quadrotors and other rigid body systems. Here's are an overview of these approaches:

4.2.1.1 Euler-Lagrange and Newton-Euler

Research on aircraft dynamics using Euler-Lagrange and Newton-Euler has been widely conducted [156] despite Euler-concise Lagrange's formulation and extension,

the Newton-Euler technique is physically straightforward to understand. The Lagrange formula is a variant of the Newton Second Law [157]. The difference between the Newton-Euler result model and the one derived via the Euler-Lagrange technique is that the expressions of "I" (pseudo inertia matrix) and "C" (Coriolis vector) are more complicated to compute in Euler-Lagrange. It's essential to understand that the 'model 13' in [157] is commonly used for quadrotor.

4.2.1.2 Quaternion approach

Is effective when roll or pitch angle approaches to 90°, the angles lose significance and a so-called gimbal lock occurs. Quaternions representation approach is the mean to solve this issue. This approach provides a mathematical language for representing 3D rotations in 4D space. The quaternion dynamics, a quaternion-based control method for quadrotor aircraft attitude stabilization, is well described in [158-159]. For model analysis and control design applications to take use of control system theories and traditional design methods like bode plot and root locus design, the linearization of non-linear systems becomes crucial. However, linear models are commonly used in quadrotor control and identification. In contrast, a quadrotor controlled by a linear controller is only stable and effective around the operation point, where the linearization condition is met [157].

4.2.2 Quadrotor Dynamic Modelling

A quadrotor is a rigid-body frame with four arms installed in an X configuration equipped with four rotors mounted at the end of each arm, independently controlled as seem in Figure 4.1. Where the quadrotor motion results from changes in the rotor's speed, the variation in the rotor speeds affects the thrust forces and produces motions. Thus, the vertical motion is the result of the increasing or decreasing by the same amount in propellers speed. As shown in Figure 4.1, the positive pitch rotation is generated by decreasing the speeds of the front motors (motors 3 and 1) compared to the rear motors (4 and 2). The positive roll rotation is adjusted by increasing the speed of the left motors (motors 1 and 4) while decreasing the speed of motors (2 and 3). The positive yaw rotation results from the difference between the speeds of the two pairs of motors, increasing motor speed (1 and 2) and decreasing the motor speed (4 and 3). All these operations should be performed while maintaining the same total thrust to

maintain the same altitude. The six degrees of freedom (DOF) characterize the location and the orientation of a quadrotor body in terms of six coordinates grouped into two reference frames are depicted in Figure 4.1.

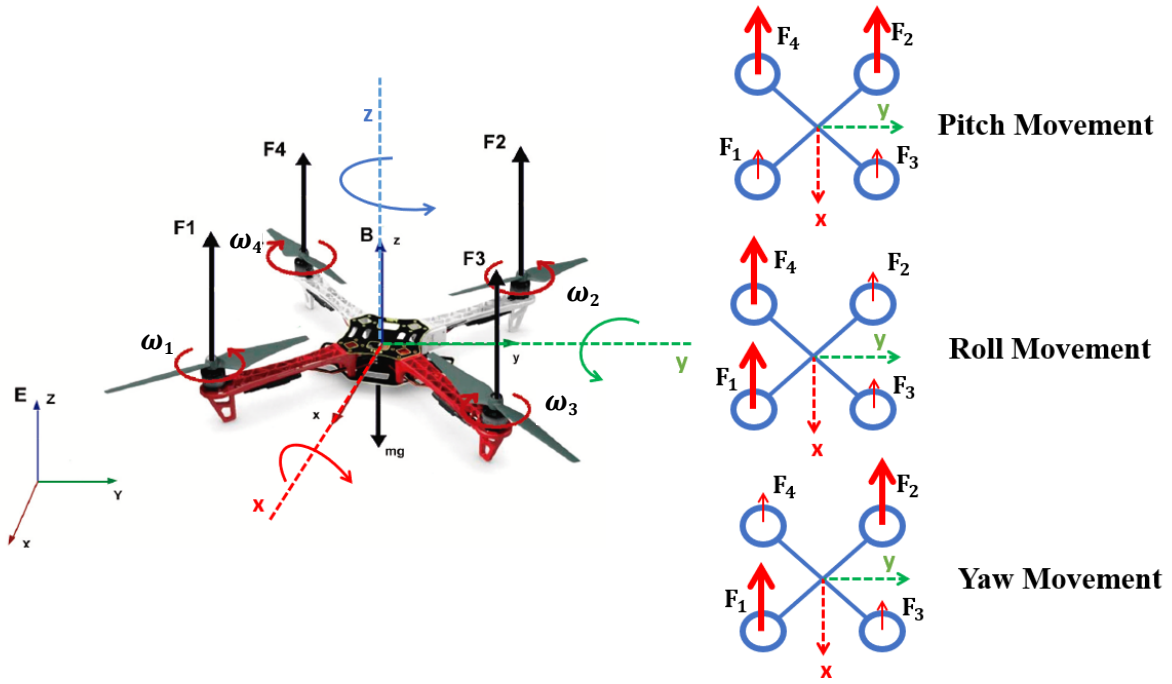


Figure 4.1: Quadrotor configuration and applied forces with Pitch, Roll, Yaw movements

The dynamic of the quadrotor is an unstable nonlinear multivariable and underactuated system. Despite having only four actuators, the quadrotor possesses six degrees of freedom. Figure 4.1, the {E} is an earth-fixed frame, and {B} is a body inertial frame supposed to be fixed on the gravity centre of the quadrotor, and it shows the applied forces. Let us consider the quadrotor structure and propellers rigid and symmetrical. From [160-163], The transformation matrix from the earth fixed frame {E} to body inertial frame {B} can be described as:

$$R = \begin{bmatrix} c\theta c\psi & c\theta s\psi + s\phi s\theta s\psi & s\phi s\psi + s\phi s\theta c\psi \\ c\theta s\psi & c\theta c\psi + s\phi s\theta s\psi & s\phi c\psi + c\phi s\theta s\psi \\ s\theta & s\phi s\theta & c\phi c\theta \end{bmatrix} \quad (4.1)$$

Where the terms s and c are for trigonometric functions *sin and cos* respectively. The angular body rates $W = [p \quad q \quad r]^T$ and Euler rates are $\dot{\xi} = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]^T$.

$$W = R_r \dot{\xi} \quad (4.2)$$

The Newton-Euler equation states:

$$\begin{cases} M = I_B \dot{W} + W \times I_B W \\ F = ma \end{cases} \tag{4.4}$$

F is the total thrust created by all four motor-propellers, while "M" is the net-total momentum in quadrotor, the acceleration vector of quadrotor represented by "a" with respect to the ground, " W *and* $\dot{W}$ " are the angular speed and acceleration vector around the gravity center, and "I_B" is the rotating moment of inertia.

$$F = \begin{bmatrix} \sin\phi \sin\psi + \cos\phi \cos\psi \sin\theta \\ \sin\psi \sin\theta \cos\phi - \sin\phi \cos\psi \\ \cos\phi \cos\theta \end{bmatrix} * \sum_{i=1}^{4} F_i \tag{4.5}$$

Where $i = 1,2,3,4$. F_i is the thrust force created by each motor-prop. Where F_i is given by the following equation:

$$F_i = K_T \omega_i^2 \quad \text{and} \quad K_T = \frac{1}{2} C_T \rho A_r r^2 \tag{4.6}$$

K_T is the aerodynamic thrust coefficient; A_r density of air in the propeller, r^2 is square propeller radius, C_T aerodynamic constant, while ω_i represents i-th motor-prop the angular speed. I_B can be categorized by (I_x, I_y, I_z) as the following rotational inertia matrix about the body frame of quadrotor:

$$I_B = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{4.7}$$

The quadrotor dynamics expressed as equation 4.8, as while, $\phi, \theta, \psi \in \mathbb{R}$ and $(-\frac{\pi}{2} \leq \phi \leq \frac{\pi}{2}, -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$ (for controllability limitation [164-165]) and $-\pi \leq \psi \leq \pi)$ is true, then:

$$\ddot{x} = \frac{1}{m} (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) U_1$$

$$\ddot{y} = \frac{1}{m} (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) U_1$$

$$\ddot{z} = \frac{1}{m}(\cos\phi\cos\theta)U_1 - g \tag{4.8}$$

$$\ddot{\phi} = \frac{U_2}{I_x} + \left(\frac{I_y - I_z}{I_x}\right)\dot{\psi}\dot{\theta} - \frac{J_r}{I_x}\bar{\Omega}\dot{\theta}$$

$$\ddot{\theta} = \frac{U_3}{I_y} + \left(\frac{I_z - I_x}{I_y}\right)\dot{\psi}\dot{\phi} - \frac{J_r}{I_y}\bar{\Omega}\dot{\phi}$$

$$\ddot{\psi} = \frac{U_4}{I_z} + \frac{(I_x - I_y)}{I_z}\dot{\phi}\dot{\theta}$$

Which may be expressed by $\dot{X} = f(X, U)$

Where :

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K_T & K_T & K_T & K_T \\ -h & h & -h & h \\ h & -h & -h & h \\ K_\tau & K_\tau & -K_\tau & -K_\tau \end{bmatrix} \begin{bmatrix} {\omega_1}^2 \\ {\omega_2}^2 \\ {\omega_3}^2 \\ {\omega_4}^2 \end{bmatrix} \tag{4.9}$$

Where h= $\frac{\sqrt{2}}{2} * l * K_T$ and *"l"* is the distance between the gravity center to the end of rotor arm, K_τ is the drag coefficient.

$$\bar{\Omega} = \omega_1 - \omega_3 + \omega_2 - \omega_4 \tag{4.10}$$

Where ϕ, θ, ψ are the angular coordinates, x , y, z are the space translational coordinates, ω_1 , ω_2 , ω_3 and ω_4 are the rotor angular speeds and U_1, U_2, U_3 and U_4 are the control inputs. I_x, I_y, I_z, m are respectively the moments of inertia and the mass of quadrotor; J_r is propeller-rotor inertia moments, and $\bar{\Omega}$ is the angular velocity, the earth gravity is g. The details of the quadrotor modeling, as well as the general forces and moments resulting from various physical effects, are well explained in [164-165]. The altitude command, or "throttle," (U_1), roll command (U_2), pitch command (U_3), and yaw control (U_4) are the quadrotor system inputs. Represented by the input vector $U = [U_1, U_2, U_3, U_4]^T$, and the state vector is given by $X = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]^T$. There are only four control inputs (U_1, U_2, U_3 and U_4) in the dynamic model of a quadrotor, while the opposite side has six outputs (x , y, z , ϕ, θ, ψ). Obviously, not all outputs can be independently controlled, Therefore, it is challenging to properly control both position and orientation at once, this is the under-actuated constraint which well figured in the equations 4.11.

$$\phi = \arcsin\left[\frac{m}{U_1}\left(\ddot{x}\sin(\psi) - \ddot{y}\cos(\psi)\right)\right] \text{ and } = \arcsin\left(\frac{\frac{m\ddot{x}}{U_1} - \sin(\phi)\sin(\psi)}{\cos(\phi)\cos(\psi)}\right) \quad (4.11)$$

Because the quadrotor normally works fairly close to the hovering point, where all angles do not exceed 20° (assumption) or all the angles and their derivative are around the zero, this implies that $(\theta = \phi = \dot{\phi} = \dot{\theta} = \dot{x} = \dot{y} = \dot{z} = 0)$ and at such a particular coordinates location $(x = x_0,\ y = y_0,\ z = z_0)$ and at certain heading 'yaw' $(\psi = \psi_0)$ are the initial conditions, and the trigonometric *cosine* and *sine* approximated to be as $(\cos\theta = 1\ ,\ \sin\theta = \theta)$. While for control $U_2 = U_3 = U_4 = 0$ but $U_1 = m.g$ are the nominal input. The control and observer designing is coming next after checking the system controllability and observability. The controllable and observable properties of the quadrotor linearized model are proved at system equilibrium points [166-168], while [169] overcame the limitation of controllability and observability validation near the hovering point or equilibrium point (in order to perform a small angle approximation) for the homogeneous representation UAV.

The linearized model [170] is determined by derivation $f(X,U)$ with respect to states for the matrix "A" and with respect to control inputs for the matrix "B", around equilibrium point and nominal controls respectively. We may perform an angular approximation, in hovering conditions $u_z = mg$, yielding linear dynamics.

$$\begin{Bmatrix} \ddot{\phi} = \frac{1}{I_x}U_2 \\ \ddot{\theta} = \frac{1}{I_y}U_3 \\ \ddot{\psi} = \frac{1}{I_z}U_4 \\ \ddot{x} = u_x\frac{1}{m}U_1 \\ \ddot{y} = u_y\frac{1}{m}U_1 \\ \ddot{z} = u_z \end{Bmatrix} \text{ where } \mathrm{X} = \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \\ x \\ \dot{x} \\ y \\ \dot{y} \\ z \\ \dot{z} \end{bmatrix} = \begin{bmatrix} x1 \\ x2 \\ x3 \\ x4 \\ x6 \\ x7 \\ x8 \\ x9 \\ x10 \\ x11 \\ x12 \end{bmatrix} = (\text{ Attitude, Position, Altitude })^{\mathrm{T}} \quad (4.12)$$

And

$$\begin{Bmatrix} u_x = \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ u_y = \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ u_z = g - (\cos\phi\cos\theta)\frac{1}{m}U_1 \end{Bmatrix} \quad (4.13)$$

Quadrotor Dynamic Modelling is an essential step in understanding and characterizing the behaviour of quadrotors. It involves creating mathematical models that capture the complex dynamics and forces acting on the quadrotor during flight. These models consider factors such as aerodynamic effects, structural considerations, and the interactions between various forces. Once a dynamic model is developed, it can serve as a foundation for further analysis and control design. However, to truly leverage the potential of quadrotors in various applications, it becomes crucial to accurately identify the parameters of the dynamic model. This process is known as Dynamic Identification.

4.3 Quadrotor Dynamic Model Identification

4.3.1 Overview

A significant proportion of research papers on quadrotor control rely heavily on mathematical models for its dynamics. Obviously, the characterization of aerodynamic effects and other additional hidden dynamics is far from trivial when this proposed mathematical formulation becomes described by the unwanted complexity and strong non-linearity that are considered a nightmare for controller design. These resulting difficulties resulted in the development of numerous quadrotor experimental characterization-based approaches [116-118]. Several decades, beginning in 1965, have seen the evolution of an alternative solution. When system identification is successful at constructing mathematical models for dynamic systems based on observed input and output data, open problems include nonlinearity and closed-loop identification, treated in [119-120]. There are also some special features in the modelling and identification process as [121-122],[116], [123]. In aeronautics applications, a continuous-time model is typically more employed and more popular than the discrete-time one, mainly because they are more intuitive [121]. In addition to that, a closed-loop identification for the quadrotor is necessary when the open loop is highly unstable, Consequently, the identification experiments must be conducted in a closed loop under a controller that maintains the minimal stability of the quadrotor or under human-operator feedback. The phenomenon of dynamic cross-coupling may greatly affect the model identification, subsequently the separation of dynamic modes now is indispensable, to get the best model estimation result. Where each input channel of a quadrotor is excited separately. Where the individual axes for roll and pitch and yaw

can be identified separately. Many pieces of literature [119],[116] proposed some schemes and classifications for quadrotor model identification, so-called white-grey-black box approaches, identification, and these methods were introduced sequentially from white to grey to a black box identification.

White Models is the first classification of system model identification where the parameters of model estimation based on the first principal model, where physical parameters of quadrotor dynamic models are extracted from direct measurements, like mass, a moment of inertia or motor coefficients, sometimes with the help of software like SOLIDWORKS, and other times the parameters are extracted from experimental attempts [124-126]. In brief, quadrotor masse, radius, aerodynamic coefficients of the rotor blades, and rotational inertia are obtained either from measurement or computation or from references [127]. As for the motors' dynamics, they can be considered as a system of first order where its constants can be identified from the experimental data [128-129].

Grey Models is the second classification of model identification that makes use of prior knowledge about the system dynamic representation and experimental response data in order to complete the model by estimating the unknown coefficients of system representation [130-132]. [133] demonstrated that the grey-box model provides improved forecasting capability in terms of thrust and moment models for the physical model. While in [130], aerodynamic coefficients were estimated using the Blade Element Momentum Theory and a Grey Box iterative parameter identification approach, the experimental show a very good correlation between the model used to find parameters and the real data from the quadrotor.

Black Box Models is the third class of system model identification, which aims to modelling directly the dynamics the system from collected input & output data [134-135]. In [136] a black-box technique is used instead of conventional mathematical modelling. Despite that, it may be appropriate to understand the influences on the quadrotor motion by giving physical meaning to the model coefficients. This comprehension would aid in system analysis and controller design or re-design in order to achieve the desired dynamic performance. The black-box approach is suitable for modelling a class of unconventional aircraft whose dynamics are difficult to model from first principles or not well understood. Much of the literature pays attention to

this area of experimental identification [137]. The state-space discrete model in identification, which overcomes the problem of closed-loop identification by adding a Pseudo Random Binary Sequence signal PRBS directly to a controller output signal to overcome the correlation of the input signal with the feedback noise that is present in the output [138]. The research, as described in [139], identified the quadrotor's nonlinear attitude subsystems as an auto-regressive exogenous input (ARX) model. The quality of the ARX model was evaluated and determined as pretty excellent. The research, [121] has interpreted the quadrotor local-dynamics and giving a meaningful information about uncertainty is evidence of the success of this method. The utilizes of the artificial neural network to learn and model the quadrotor dynamics has been addressed by both [136] and [140], these studies demonstrate that the identified model resulted from black-box neural network training also learnt the noise and the dynamics of the trends.

4.3.2 Quadrotor Dynamics Model Identification Techniques

Model identification theory seeks to discover mathematical models that explain the dynamics of systems based on gathered datasets [141]. Selecting appropriate approaches for UAV platform experimental testing has been examined broadly in the pieces of literature [122],[142-144]. But when a process is too hard and complex to be described by physical laws or there isn't enough available information about hidden dynamics, it would make sense to start working with a black-box model identification approach [137]. Measurements of input-output data are required for this approach to obtain the mathematical model. In addition, if prior knowledge, e.g., the bandwidth of this model is available, it will greatly help in the identification process. Moreover, the open-loop identification provides an unbiased estimation of the quadrotor model where the bias term is zero due to a lack of feedback, where there is uncorrelation between the noise the control surface, and because rotorcraft systems are unstable in open-loop, the identification must be performed in in closed-loop where the feedback regulation is active for roll and pitch dynamics and open-loop for yaw. The undesirable correlation of feeded noise and control surface in a close-loop leads to bias errors in the estimation response. The system to be identified in the closed-loop approach is depicted in Figure 4.2. The closed-loop identification approach is divided into the following main approaches:

The direct approach consists of collecting data from the controller u and response outputs y to identify the dynamic system model $G(s)$ as if the dynamic system is in an open-loop system. The system can be identified by using any prediction method [145].

The indirect approach identifies a closed-loop system by using collected data from reference r and the output y, than determines the dynamic model based on a previously known controller $C(s)$ of the closed-loop model that has been identified before [146].

Joint input output approach considers the control and the response $[u, y]$ signals as a cascaded system output. Where the reference and the noise $[r, n]$ signal considered are jointly perturbing the system. The dynamic model $G(s)$ identified from this joint input-output system.

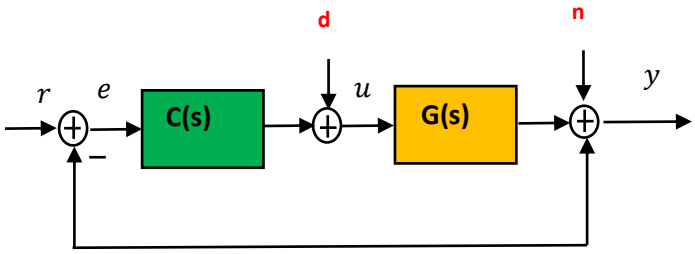


Figure 4.2: Close-loop system identification framework

According to the prior knowledge about the system e.g. (stability, bandwidth ...) and the picked signals from the experiments, the suitable identification approach is on Table 4.1, that summarizes the information we must have for each identification approach.

Table 4.1: Identification approaches

Signal and Knowledge	Control Signal	Response Signal	Reference Signal	Controller Prior-Knowledge
Direct	Yes	Yes	—	—
Indirect	—	Yes	Yes	Yes
Joint Input-Output	Yes	Yes	Yes	—

Each modelling result from system identification will be associated with an applicable frequency range, defined as the range over which the frequency response can be accurately identified (i.e., good coherence) or as the range that the identified model is expected to be accurate [122]. The acceptable identified model result from the identification process is expected to be precise close to the desired closed-loop bandwidth frequency (natural frequency). Therefore, the excitation signal input must be carefully chosen. As long as the dynamic models are not well excited by the test input signal, those models will remain hidden and not appear in the experimental data. Consequently, they will not be included in the final identified model. In identification literature terms, the input signal should be continuously stimulating or persistently exciting [122]. The excitation inputs for identification have been broadly examined by a large literature on optimal input design. In sum, the well-selected inputs mean excellent starting in system-identification [147-149] and [122].

In our work here, and owing to insufficient information about quadrotor parameters, e.g., inertia moment and aerodynamics coefficient, make it imperative for us to trend over using the Black-Box model identification direct approach. Where the system identification is carried in the time domain and each input channel of the quadrotor is excited separately as in Figure 4.3.

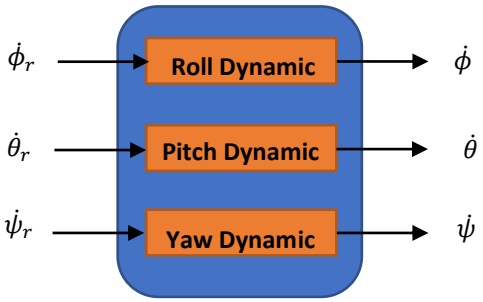


Figure 4.3: Input–output dynamic model identification channels

The following steps must be completed with respect to using prior knowledge or experience. Figure 4.4 depicts the identification loop that must be followed to obtain a perfect model of any system versus the use of prior-acknowledgements versus experiments [150].

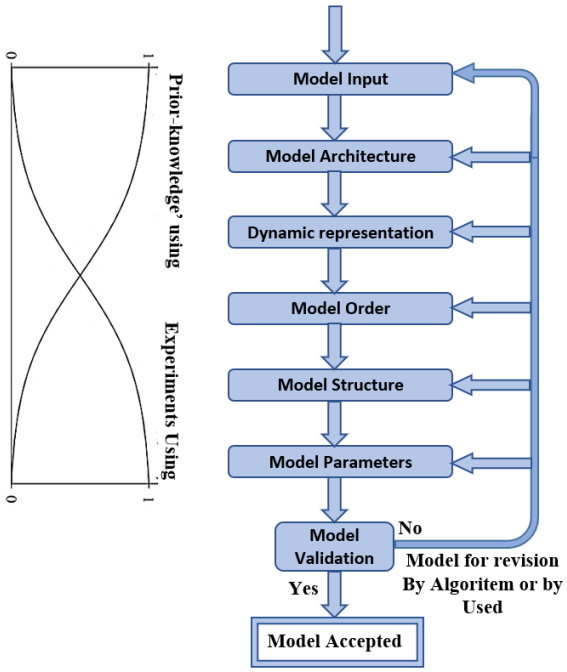


Figure 4.4: Identification loop steps [150]

4.3.2.1 Techniques for Input Selection

This step often involves a trial-and-error approach with the use of previous knowledge about systems, considering the current understanding of system physics. Fundamentally, four distinct techniques may be used [150]:

- May use all possible input separately.
- All input combinations at one.
- Unsupervised input chosen.
- Supervised input selection.

4.3.2.2 Excitation Input Selection

This stage needs prior knowledge about the system and the objective of modeling. For instance, in black box modeling, measurements data sets, and bandwidth are the essential sources of information. This is likely the stage that requires the most engineering skill. The type of excitation input used can have a significant impact on the quality of the resulting model and the accuracy of the identification process. The accuracy of model estimates is often impacted by the presence of noise in experimental data. As a result, it is crucial to design input signals that ensure a specific level of

accuracy in the model. The issue of model accuracy in system identification is closely tied to the design of input signals.

Figure 4.5 illustrate some of the most commonly used excitation inputs in system dynamic identification [212-213]:

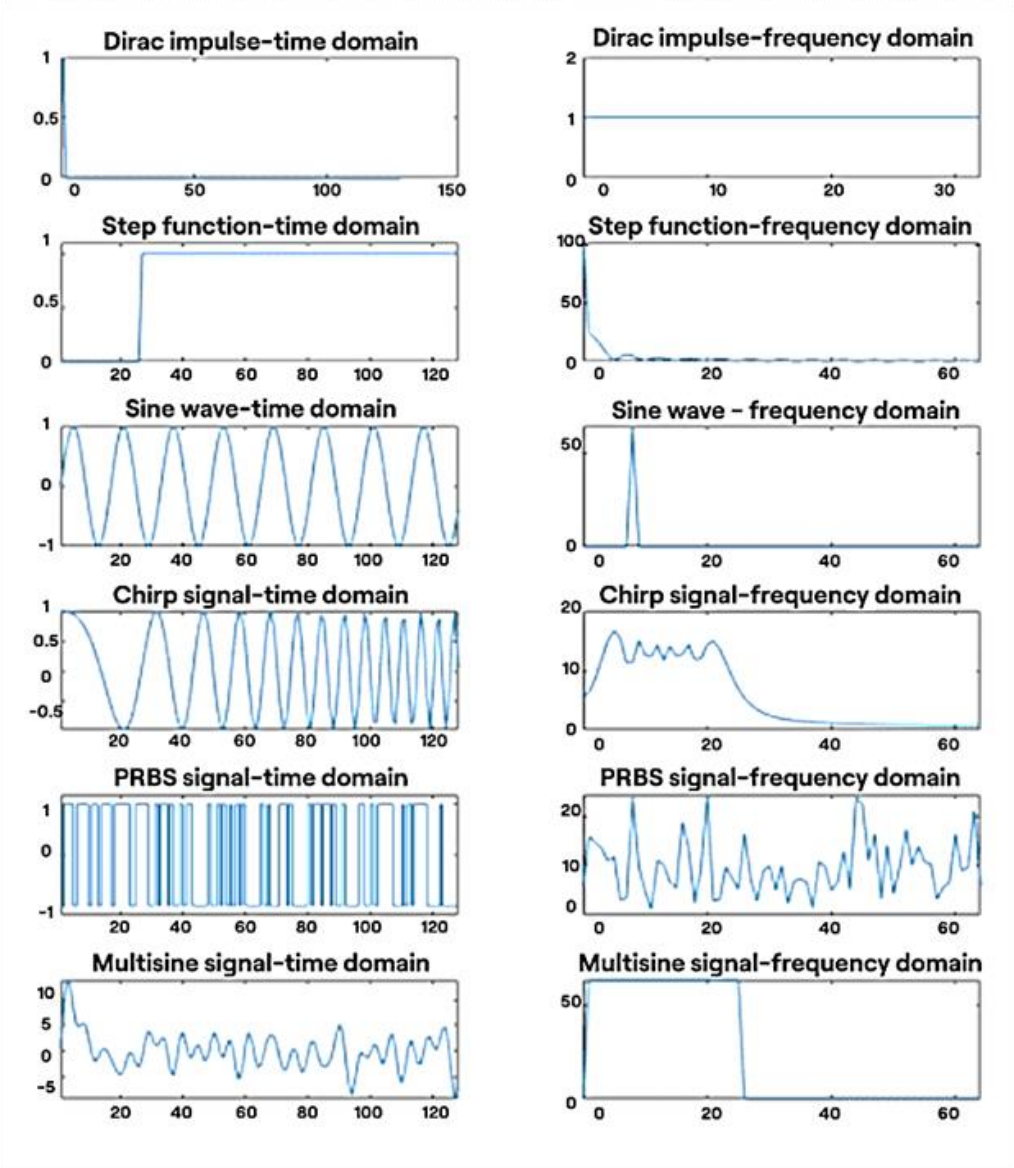


Figure 4.5: Commonly used excitation inputs in time/frequency domain

Impulse Input: An impulse input is a short-duration and a high magnitude input signal that represents an instantaneous change in the system's input. It provides information about the system's response to sudden disturbances or shocks.

Step function input: Step inputs are simple, abrupt changes in the input signal that can provide a clear and well-defined response from the system. The selection of amplitude is the only variable here. The static gain and Rise time, overshoot is directly related to the step response.

Sum of sinusoids inputs: it involves injecting a combination of sinusoidal signals with different frequencies and amplitudes into a system. The use of a sum of sinusoids as an excitation input allows for a precise and controlled way to excite a system and obtain its response for system identification purposes. The frequencies and amplitudes of the sinusoids can be chosen to excite specific modes of the system and to obtain a more complete response. This type of excitation input is often used in frequency response analysis and modal analysis. In practice, a combination of these excitation inputs may be used in order to obtain a more comprehensive representation of the system's dynamic properties. The choice of excitation input will depend on the specific requirements of the identification process and the properties of the system being modelled.

Chirp Input: A chirp input is a type of signal that varies its frequency continuously over time, usually in a linear fashion. It starts at a specific frequency and sweeps or ramps up or down to another frequency over a defined period. Chirp inputs are commonly utilized in dynamic identification to probe and characterize the frequency response of a system across a wide frequency range. By applying a chirp input and analysing the system's output, valuable information about the system's dynamics, resonances, and frequency-dependent behavior can be obtained. Chirp inputs enable the identification of system parameters and the determination of its frequency response characteristics.

Pseudorandom Binary Sequence (PRBS): is a type of excitation input used in system identification to excite a system and obtain its response. PRBS is a digital sequence of binary values (0 or 1) that appears to be random but is actually generated by a deterministic algorithm. The sequence is designed to have specific statistical properties such as uniform distribution, low autocorrelation, and good spectral properties. The use of PRBS as an excitation input allows for a precise and controlled way to excite a system and obtain a response for system identification purposes.

Additionally, PRBS sequences can be used to test and validate the performance of digital systems.

In System identification theory, one of the most used excitation input is the PRBS, is used as input for each attitude dynamics Figure 4.6, PRBS signal has proved extraordinarily effective in conjunction with time-domain identification, It is a periodical deterministic signal with Gaussian noise-like features [150]. As a prior knowledge gained from analyses of quadrotor in open-loop responses, the dominant dynamics are located at a range of 0.1 to 20 rad/s [151] [122], where the excitation PRBS is formed based on the pre-defined range of interest [149-153]. The PRBS is a two-state signal with a magnitude of a, which is produced with the aid of a feedback shift register. Where registers bits number of n, and the highest possible length is : $N = 2_n - 1$, Consider the clock period to be Δt (switching time) send with frequency of $(1/\Delta t)$, the PRBS period time becomes $T = N.\Delta t$. Represents the highest length of a signal where the lowest length is Δt . Due to the fact that the PRBS length is an odd number in a period, it takes two state or values $[-a$ or $+a]$, where the number of $+a$ is less than the number of values $-a$ by one. See Figure 4.6.

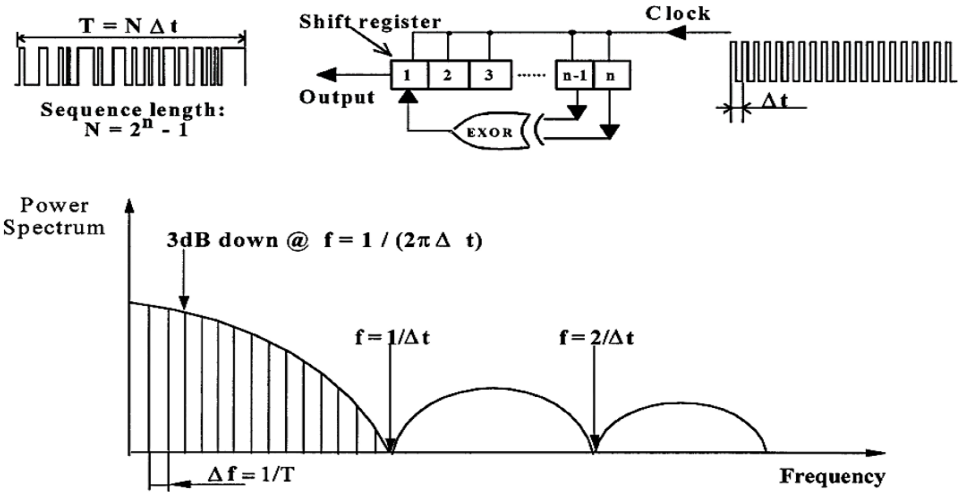


Figure 4.6: PRBS Signal Generator and Its Power Spectrum

Selection of the N (therefore T) value and Δt value is a compromise between:

- a good identification of the static gain.
- a good excitation on the system frequency band

Let τ_{max} τ_{min} respectively be the largest and the smallest constant time of the system, the good identification of the static gain leads to choosing:

$$N.\Delta t = 3 \text{ to } 5\ \tau_{max} \quad \text{and} \quad 0.3\frac{1}{\Delta t} = \frac{1}{2\,\pi\,\tau_{min}} \tag{4.14}$$

In the context of system identification, a persistent excitation input signal is an input signal that is used to excite the system for a sufficient amount of time so that the response of the system can be accurately measured and used to estimate its dynamic behavior. Persistent excitation refers to the requirement that the input signal excite all the modes of the system for a long enough time such that the response can be accurately captured. Each excitation input has order of exitation which used to assess the quality of the excitation input signal and to determine if it is sufficient to obtain a complete and accurate response from the system. There are several methods to measure the order of excitation, including , Power Spectral Density (PSD) , Singular Value Decomposition (SVD) , Excitation Energy , Input Entropy. These methods can be used to measure the order of excitation and determine if the excitation input signal is sufficient to obtain a complete and accurate response from the system. It is important to use a persistent excitation input signal in system identification because if the input signal is not persistent enough, the system response may not contain all the necessary information to accurately estimate its behavior. This can result in inaccurate or incomplete models of the system. Therefore, selecting an appropriate persistent excitation input signal is an important aspect of the system identification process.

4.3.2.3 Effective Data Preparation for System Identification

Good data preparation allows for efficient data analysis, The preparation of well-founded and robust data should begin with the experiment itself. If the experiment bench test was well-prepared by eliminating all external effects that could be injecting into our data and all sensors were well calibrated, here we can say that there is a big chance of getting an accurate model representing the identified dynamics. All this is true only after performing another task in data preparation for identification, which comes after collecting the data from experiments as follows:

Data Collection: The first step in data preparation is to collect input-output data from the system under study. This data should be collected under various operating

conditions and should cover a sufficient range of the input-output space to accurately capture the system's behaviour.

Data Cleaning: Data collected from real systems often contains noise, outliers, or other anomalies. These should be removed or corrected before the data is used for system identification.

Data Pre-processing: Data collected from the system may need to be pre-processed before it can be used for system identification. This may include normalizing the data, filtering the data to remove high-frequency noise, or transforming the data to a suitable form for system identification.

Data Segmentation: The collected data should be divided into appropriate segments for use in system identification. This may involve selecting a portion of the data for use in model estimation, and reserving another portion for validation of the model.

Data Re-sampling: If the collected data is sampled at a high frequency, it may need to be resampled to a lower frequency to reduce the computational complexity of the system identification process. This should be done carefully to ensure that important dynamics of the system are not lost.

By considering these key steps in data preparation, the system identification process can be more effective and the resulting model can be more accurate and representative of the system under study.

4.3.2.4 Selecting Model Architecture for System Identification

Selecting an appropriate model architecture for system identification can be challenging and complex. It involves a thorough evaluation of various factors that impact the effectiveness of the identified model. These factors encompass the purpose of the identification problem, which can range from modeling dynamic systems to approximating static systems or even classifying system behavior. Moreover, model architecture selection may be driven by specific goals, such as optimization, fault detection, control, or optimization. These objectives influence the choice of model architecture and its adaptability to the intended application.

Additionally, the number of input/output variables and the quality of the dataset play crucial roles in imposing constraints on the class of possible models and the approach to identification, particularly in the presence of noise and disturbances within the dataset.

Furthermore, aspects like memory limitations, user experience, the choice between online or offline training, and the availability of suitable identification tools are other important considerations that must be taken into account when selecting the appropriate model architecture.

4.3.2.5 Model Structure in System Identification

In system identification, the model structure refers to the form or architecture of the mathematical model that is used to represent the behaviour of the system under study including, linear models which represent the system as a linear combination of inputs and system parameters. Auto-Regressive (AR) models and Moving Average (MA) models, and ARMA models. For nonlinear models which represent the system as a nonlinear function of inputs and system parameters, include Hammerstein models, Wiener models, and Nonlinear Autoregressive with Exogenous Input (NARX) models. For state-space structure which represent the system as a set of first-order differential equations, which is widely used in control systems and can represent both linear and nonlinear systems. For Volterra structure which represent the system as a set of nonlinear kernels that describe the system's impulse response, is particularly useful for nonlinear systems with nonlinear memory effects. While for transfer function structure which is the ratio of the Laplace transforms of the output and input signals. Transfer function structure is widely used for linear systems and can be represented in either time-domain or frequency-domain forms.

The choice of model structure will depend on the characteristics of the system under study and the goals of the system identification process. It is important to choose a model structure that accurately captures the system behaviour while also being computationally tractable. The NARMAX or NARX may be a suitable representation for the nonlinear system dynamics if one-step prediction is used. In this work the direct approach was selected by directly applying a prediction error-method (PE) to the input-output signals, given that the method can be applied to systems either stable or

unstable as long as the predictor is stable. The ARX and ARMAX models are compliant with this requirement. Moreover, it is guaranteed for output-error and Box-Jenkins model structures [154] if special precautions are taken. Since the instrumental variable (IV) method provides the same level of precision as the direct PE method, it is also utilized. The following equation 4.15 and is a representation of the general linear model structure in Figure 4.7.

$$\begin{aligned} y(t) &= G(\theta, q)u(t) + v(t) \\ &= G(\theta, q)u(t) + (H(q)e(t)) \end{aligned} \tag{4.15}$$

$G(q, \theta)$ is the real system and $H(q)$ is the noise model. where white noise is assumed to be the signal $e(t)$ Auto-Regressive with eXogenous-Input, ARMAX (Autoregressive Moving-Average-Model with Exogenous-inputs), and BJ (Box-Jenkins) are the possible models structures for $\hat{G}(q, \theta)$

$$\begin{aligned} &\textbf{ARX :} && \hat{G}(q,\theta) = \tfrac{B(q,\theta)}{A(q,\theta)}, && \hat{H}(q,\theta) = \tfrac{1}{A(q,\theta)} \\ &\textbf{ARMAX :} && \hat{G}(q,\theta) = \tfrac{B(q,\theta)}{A(q,\theta)}, && \hat{H}(q,\theta) = \tfrac{C(q,\theta)}{A(q,\theta)} \\ &\textbf{BJ:} && \hat{G}(q,\theta) = \tfrac{B(q,\theta)}{F(q,\theta)}, && \hat{H}(q,\theta) = \tfrac{C(q,\theta)}{D(q,\theta)} \end{aligned} \tag{4.16}$$

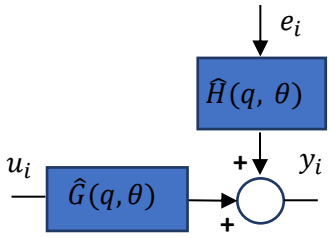


Figure 4.7: General Linear Model Structure

ARX model

$$y(k) + a_1 y(k-1) + \cdots + a_n y(k-n) = b_1 u(k-1) + \cdots + b_m y(k-m) + e(k) \Rightarrow$$

$$y(k) = \underbrace{-a_1 y(k-1) - \cdots - a_n y(k-n)}_{\text{Auto-Regressive}} + \underbrace{b_1 u(k-1) + \cdots + b_m y(k-m)}_{\text{eXogenous-Input}} + \underbrace{e(k)}_{\text{Noise at Step k}}$$

Where $a_i\,(i = 1{:}\,n)$ and $b_j(j = 1{:}\,m)$ are linear parameters to be determined by least squares methods.

4.3.2.6 Order Selection and Input Delay

This step is usually accomplished through a combination of trial & error and prior knowledge of the system. Choosing a model with a greater degree of dynamics increases the problem's dimensions and complexities, which may reduce the precision of a static approximation. Following Akaike's information criteria (AIC), which guide enhanced model selection, the lowest achievable AIC represents the optimal balance between model fit and generalizability. This helps in finding the right trade-off between complexity and precision.

AIC in order selection involves choosing the optimal order or complexity of a model based on the AIC values computed for different orders. The lower the AIC value, the better the model is considered to be in terms of balancing goodness of fit and model complexity.

Here is a general step-by-step process for order selection using AIC:

1-Define a range of potential orders: Determine a range of orders or complexities for the model we are working with. For example, if we are fitting an autoregressive (AR) model, consider different lag lengths as potential orders.

2-Fit models for each order: Fit models for each order in the defined range. This involves estimating the parameters and computing the likelihood function for each model.

3-Calculate AIC values: Compute the AIC value for each model using the estimated likelihood and the number of parameters in the model. The AIC value is typically calculated as *AIC = -2(log-likelihood) + 2k*, where *k* represents the number of model parameters.

4-Select the model with the lowest AIC: Compare the AIC values and select the model with the lowest AIC value. This model is considered to strike the best balance between goodness of fit and model complexity.

5-Perform model diagnostics: Once the model with the lowest AIC is selected, it is important to perform diagnostics to ensure that the chosen model adequately captures the underlying patterns in the data and meets any assumptions associated with the model. This may involve examining residuals, conducting hypothesis tests, or other relevant diagnostic procedures.

4.3.2.7 Model Validation

Model validation is an essential step in the system identification process as it allows us to assess the quality of the identified model. The objective of model validation is to determine if the model meets the desired requirements and specifications. In the case of system identification, the performance of the model can be evaluated using metrics such as the goodness of fit, residual analysis, and prediction error. These metrics allow we to determine the accuracy of the model and its ability to capture the dynamics of the system under study. In control systems, the criteria for model validation become more focused on the behaviour and performance of the system in a closed-loop setting. For example, the closed-loop response of the system can be evaluated for stability, steady-state error, and robustness to disturbances and measurement noise. Additionally, the sensitivity of the model to fault detection is also an important aspect of model validation in control systems. This involves evaluating the ability of the model to detect and isolate faults in the system and to provide appropriate responses [137].

4.3.3 Adaptive Neuro-Fuzzy Inference System

Adaptive Neuro-Fuzzy Inference Systems (ANFIS) is a type of hybrid intelligent system that combines the strengths of artificial neural networks and fuzzy logic systems. They are designed to model complex, non-linear relationships between input and output variables [237]. ANFIS integrates the flexibility of fuzzy logic in handling linguistic rules and the learning capabilities of neural networks [238]. The model adapts and refines its parameters based on input-output data to optimize its performance and accuracy.

4.3.3.1 ANFIS architecture

The ANFIS architecture comprises five layers, as depicted in Figure 4.8. Within each layer, the nodes are categorized into two types: adaptable nodes and fixed nodes. In this system, layers 2, 3, and 5 consist of fixed nodes (represented by circular nodes), while layers 1 and 4 contain adaptive nodes (represented by square nodes). Adaptive nodes, unlike fixed nodes, have the capability to learn and adjust their parameters.

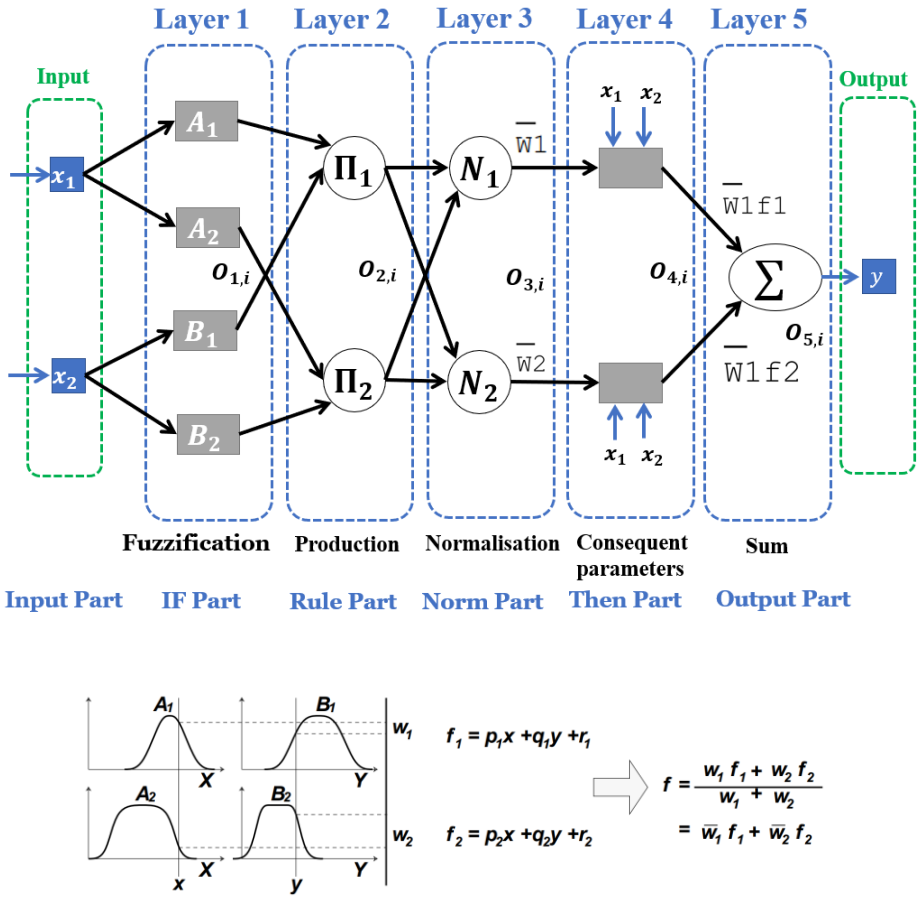


Figure 4.8: ANFIS Structure

To illustrate the rules of each layer, let's consider two fuzzy if-then rules:

Rule1: if x_1 is A_1 and x_2 is B_1 then $f = p_1 x_1 + q_1 x_2 + r_1$

Rule2: if x_1 is A_2 and x_2 is B_2 then $f = p_2 x_1 + q_2 x_2 + r_2$

The ANFIS architecture involves input variables x_1 and x_2, fuzzy sets A_i and B_i, and the output variable f. The consequent parameters $\{p_i, q_i, r_i\}$ are crucial and need to be

determined through the ANFIS training process. The functionality of each layer can be described as follows [215]:

Layer 1: Within this layer, each node, i, is characterized by a membership function. In fuzzy logic, variables are made fuzzy through the utilization of membership functions. These membership functions, represented by curves, determine the mapping of a point in the input space to a membership value within the range of [0 to 1]. Membership functions can take on various forms, with the most commonly used ones being Triangular, Trapezoidal, and Gaussian membership functions.

$$O_{1.i} = \sigma_{A_i(x)} \quad \text{and} \quad O_{1.i} = \sigma_{B_i(x)} \quad i = 1.2. \tag{4.17}$$

In this context, x represents the input of node i, while $O_{1.i}$ refers to the membership function of A_i. Typically, the membership function $O_{1.i}$ for A_i is defined using the Gaussian function, expressed as:

$$\sigma_{A_i(x)} = \exp\left(\frac{-(x-c)^2}{\sigma^2}\right) \tag{4.18}$$

In the given equation, σ represents the standard deviation, and C represents the center of the Gaussian membership function. These parameters, known as antecedent parameters, play a crucial role in defining the membership functions. Their values are determined through the optimization algorithm.

Layer 2: The firing strength of a rule is determined by the following relationship

$$w_i = \sigma_{A_i(x)} * \sigma_{B_i(x)} \quad \text{Where } i=1.2 \text{ (in example case)} \tag{4.19}$$

Layer 3: The firing strength of each rule is normalized by dividing the firing strength of the i^{th} – rule by the total firing strength of all rules.

$$O_{3_i} = \overline{w_i} = \left(\frac{w_i}{w_1+w_2}\right) \quad i = 1.2. \tag{4.20}$$

Layer 4: The outcome of the fuzzy rule is determined by the following equation:

$$O_{4_i}=\overline{w_i}f_i=\overline{w_i}(p_i\, x_i+q_i x_i + r_i) \qquad i = 1.2. \tag{4.21}$$

$\{p_i, q_i, r_i\}$ involving the set of consequent parameters which are computed using the optimization algorithm.

Layer 5: In this layer, the outputs from Layer 4 are summed together:

$$O_{5_i} = \sum_{i=1}^{R} \overline{w_i} f_i \qquad i=1.2. \tag{4.22}$$

4.3.3.2 ANFIS training

The training algorithm for ANFIS typically involves an iterative process that adjusts the parameters of the model to minimize the error between the predicted output and the actual output. One common algorithm used for training ANFIS is the hybrid learning algorithm, which combines gradient descent optimization and least squares estimation. Here are the steps involved in the ANFIS training algorithm [218]:

1-Initialize the consequent parameters: Start by initializing the consequent parameters of the ANFIS model. This can be done randomly or using some initial values based on prior knowledge.

2-Forward pass: Pass the training data through the network to compute the firing strengths of each rule and the consequent values for each data point.

3-Compute the error: Calculate the error between the predicted output and the actual output for each data point in the training set.

4-Parameter update - Part 1: Update the premise parameters (antecedent membership function parameters) using a gradient descent optimization method. This involves computing the gradients of the error with respect to the premise parameters and updating them to minimize the error.

5-Parameter update - Part 2: Update the consequent parameters using a least squares estimation method. This involves solving a set of linear equations to find the optimal values for the consequent parameters that minimize the error.

6-Repeat steps 2-5: Repeat the forward pass, error calculation, and parameter updates iteratively until the error converges to a satisfactory level or a maximum number of

7-Testing and validation: Evaluate the trained ANFIS model using the testing set to assess its performance. Calculate performance metrics such as mean squared error (MSE) or coefficient of determination (R-squared) to measure the accuracy and generalization capability of the model.

8-Model refinement: If the performance is not satisfactory, you can refine the model by adjusting parameters such as the number of rules, the type of membership functions, or the learning rate to improve the model's accuracy and generalization capability.

By repeating these steps iteratively, the ANFIS model gradually learns to approximate the underlying system's behaviour and can make accurate predictions for unseen input data.

The adaptive nature of ANFIS allows it to learn from data, refining its fuzzy rules and consequent parameters, and improve its modeling accuracy. It can handle complex nonlinear systems, provide interpretable models, and adapt to changing system dynamics. ANFIS has found applications in various fields, including system identification, control systems, pattern recognition, and data analysis, where the combination of fuzzy logic and neural networks offers advantages in modeling and inference tasks.

4.3.3.3 ANFIS Rules

When it comes to acquiring linguistic rules for decision-making within an ANFIS, there are generally two approaches that can be employed. two fundamental approaches are commonly employed. These approaches, often referred to as the "indirect" and "direct" methods, play a crucial role in shaping the decision-making process of ANFIS models. Let's explore each of them:

The first approach or direct approach, relies on consulting human experts who provide rules and membership functions for fuzzy subsets. However, these initial rules and functions are often too simplistic for practical application. The effectiveness of linguistic rules in developing a fuzzy model is contingent upon the quality and accuracy of the rules and membership functions [230].

The second approach, known as the indirect approach, is data-driven and involves using supervised learning algorithms to train the ANFIS model based on input-output data. This method entails dividing the premise and consequence space and establishing a mapping between them. By employing training algorithms, the structure and parameters of the fuzzy model can be adjusted using numerical information. As a result, this approach generates grid-like fuzzy partitions [230]. An example of grid fuzzy partitions is depicted in Figure 4.9, when the input variables exhibit a high level of correlation, the number of partitions expands to encompass these correlations.

4.3.3.4 Clustering Techniques for ANFIS Input Space Analysis

Clustering is involves grouping a set of data points into clusters based on their similarities or proximity in the data space. The goal of clustering is to partition the data into meaningful subgroups (clusters) such that the data points within a cluster are more similar to each other than to those in other clusters. Clustering is used for a variety of purposes, such as data segmentation, pattern recognition, outlier detection. In the context of system identification, clustering can be used to identify different modes of operation in a dynamic system. For example, a system may exhibit different behaviour patterns in different operating conditions, and clustering can be used to group these different modes into separate clusters. This information can then be used to develop more accurate models of the system, as well as to improve control and monitoring of the system. Clustering can be used to form the input membership functions when using Takagi-Sugeno (TS) fuzzy or ANFIS. The idea is to use clustering to segment the input space into different regions. The membership functions for each mode can then be used as inputs to a ANFIS controller for example, which can adjust its control strategy based on the current mode of operation or provide an accurate and interpretable output of the nonlinear system if ANFIS is for modelling. Creating a fuzzy rule is closely related to how the input space is divided.

There are several types of clustering algorithms, each with its own strengths and weaknesses, and they can generally be classified into the following categories [219-220]:

Centroid-based Clustering: In this type of clustering, the cluster center is represented by the mean or median of the data points in the cluster. Examples of centroid-based clustering algorithms include K-Means, K-Medoids, and Fuzzy C-Means.

Partitioning-based Clustering: algorithms work by dividing the data space into a number of partitions or clusters Figure 4.9, and then assigning each data point to the nearest partition. The objective is to minimize the sum of distances between data points and the corresponding cluster center. The process of grid partitioning involves dividing a space into a grid-like structure to prevent overlapping in the input space this partitioning has fast processing time and not require the distances calculation. When applying the grid partitioning method, the partitioned areas, which represent specific regions associated with fuzzy rules, are generated uniformly to ensure smooth analysis of the fuzzy rules. Grid partitioning is typically employed when the number of input variables is limited, resulting in a lower-dimensional input space. To illustrate, consider a scenario where there are 10 input variables, each divided into two membership functions. In this case, the space is divided into 2^10 = 1024 specific areas. Consequently, a rule is established for each of these areas, resulting in a highly intricate structure comprising 1024 rules. As a result, the grid partitioning method is primarily utilized when the number of input variables is relatively small.

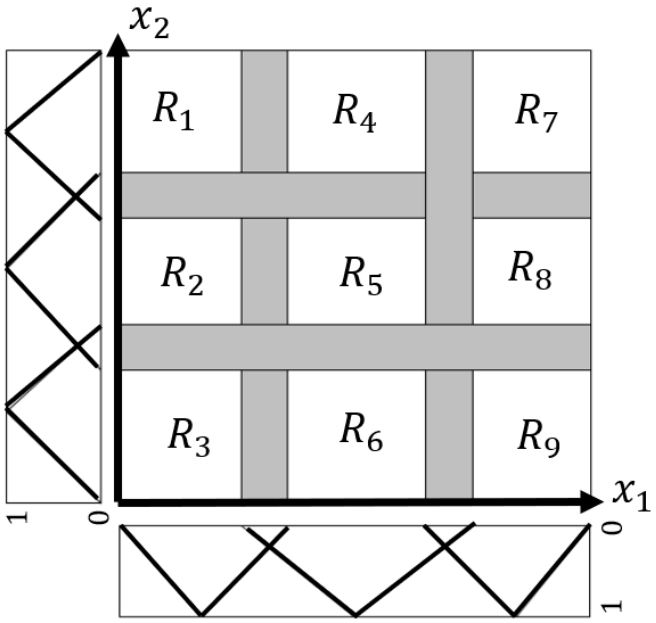


Figure 4.9: Grid Clustering

Hierarchical Clustering: In hierarchical clustering, the data is organized into a hierarchy of clusters, with the smallest clusters merging into larger ones. Hierarchical clustering algorithms can be either agglomerative (bottom-up) or divisive (top-down). Examples of hierarchical clustering algorithms include Agglomerative Hierarchical Clustering and Divisive Hierarchical Clustering.

Density-based Clustering: In this type of clustering, clusters are formed around dense regions of the data space. Examples of density-based clustering algorithms include DBSCAN and OPTICS.

Distribution-based Clustering: In this type of clustering, the data is modelled as a probability distribution, and clusters are formed based on the estimated distribution. Examples of distribution-based clustering algorithms include Gaussian Mixture Models (GMM) and Latent Dirichlet Allocation (LDA).

Graph-based Clustering: In graph-based clustering, the data is represented as a graph, and clusters are formed based on the connections between the data points. Examples of graph-based clustering algorithms include Spectral Clustering and Normalized Cuts.

Subtractive Clustering: Subtractive clustering is a type of data clustering algorithm that works by successively removing the centres (prototypes) of clusters in the data space until a stopping criterion is reached. The algorithm starts by assuming that all data points belong to one big cluster and gradually splits it into smaller clusters by iteratively identifying the densest area in the data space and treating it as a new cluster center. The process is repeated for the remaining data points until a stopping criterion is met, such as a certain number of clusters being formed or a minimum cluster size being reached. The advantage of subtractive clustering is that it can identify clusters of arbitrary shapes and sizes in high-dimensional data spaces, which is often a challenging task for other clustering algorithms. However, the choice of stopping criterion can have a significant impact on the results of the algorithm, and it can be difficult to determine an appropriate criterion. Additionally, subtractive clustering can be computationally expensive, especially for large data sets, as the algorithm requires a dense search of the data space to identify cluster centres [221-223].

Choosing the right type of clustering algorithm depends on the characteristics of the data, such as the number of clusters, the shape of the clusters, and the presence of noise and outliers in the data.

4.3.3.5 Model Identitication Using ANFIS

The significance of the ANFIS model lies in its ability to serve as a potent tool for model identification and state prediction in complex systems, eliminating the need for a separate identification form for nonlinear systems. The adjustable parameters within the network weight values enable ANFIS to effectively identify nonlinear systems and approximate the input and output data of the system. ANFIS combines the advantages of fuzzy identification and neural network identification. It requires fewer computational epochs compared to neural networks when dealing with highly complex nonlinear systems. It effectively balances structure knowledge with adaptability, offering both speed and robustness [216].

For systems with multiple inputs and multiple outputs (MIMO), such as a quadrotor, it is possible to decompose them into multiple Multi-Input Single-Output (MISO) systems, simplifying the analysis of the system's Roll, Pitch, and Yaw dynamics. In this study, the focus is on modeling a MISO system, where each attitude channel is identified separately. One of the challenges in MISO system identification using ANFIS is the selection of appropriate input variables. The performance and accuracy of the ANFIS model heavily rely on selecting the most relevant input variables that capture the essential characteristics of the system. The wrong choice of input variables can lead to poor model performance and inaccurate system identification.

The process of input selection in ANFIS involves determining which variables have the most significant impact on the system's behaviour and output. This selection process is crucial for obtaining a compact and effective model that accurately represents the underlying system dynamics. Various methods and techniques can be employed for input selection in ANFIS, such as domain knowledge, statistical analysis, exploratory data analysis, correlation analysis, and feature selection algorithms. These approaches aim to identify the most informative and relevant input variables while reducing the dimensionality of the input space [217]. Such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA) to identify the most

important input variables. Other option is to use information criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC) to determine the optimal number of input variables for the ANFIS model. It is essential to carefully consider the input selection process in ANFIS to ensure that the chosen input variables capture the essential system dynamics and result in an accurate and reliable model [224-225]. The Exhaustive Search (ES) algorithm utilizes to select the optimal inputs by systematically evaluating and checking all potential combinations. The Sequential Forward Search (SFS) algorithm is a feature selection technique commonly used in machine learning and pattern recognition tasks. Its purpose is to identify the most informative subset of features from a given feature space. The SFS algorithm starts with an empty set of selected features and iteratively adds one feature at a time, evaluating the performance of the model at each step. The feature that improves the model's performance is selected and added to the selected features. This process continues until a predetermined number of features are chosen or until the performance improvement becomes negligible.

In the context of ANFIS, the Sequential Forward Search algorithm can be applied to select the most relevant input features for building an ANFIS model. By iteratively adding features based on their impact on the model's performance, SFS helps to identify the most significant variables and reduce the dimensionality of the input space, improving the efficiency and interpretability of the ANFIS model [225-226-227]. Overall, ANFIS combined with the Sequential Forward Search algorithm provides a powerful framework for data modeling, inference, and feature selection in complex systems, leveraging the strengths of both fuzzy logic and neural networks.

4.4 Quadrotor Control

4.4.1 Overview

The UAV system's main component is the control system. The quadrotor, as a kind of UAV system, needs a control system with excellent characteristics. notably in a critical environment or when task exigencies are present. Owing to the quadrotor's high nonlinearity and multivariable input-output, and strong coupling between channels;

input uncertainty; time delay, uncertainties, and hidden dynamics, makes control systems become more complex. With the evolution of classical control theory, several novel control approaches have emerged. Many review and survey articles outline the quadrotor control [12] [171-173], concludes that the current UAV controllers fall into three primary groups as Figure 4.10.

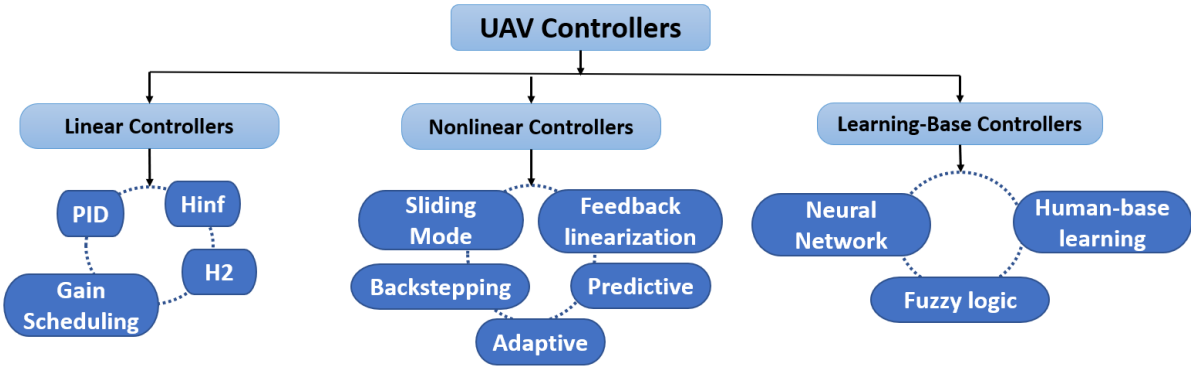


Figure 4.10: Categorizing UAV Control Systems

Commencing with linear control and advancing through nonlinear control, variable structure controllers, adaptive control, and intelligent control, various classifications exist within the field. Another categorization depends on the utilization of dynamic models, controllers designed with a reliance on the model are termed "model-based controllers," while those engineered without knowledge of the dynamic model in use are denoted as "model-free controllers." It's worth noting that certain controllers can be developed using both approaches. For instance, the PID controller can be designed based on a model using MATLAB's control design tools. However, in cases where the model is unavailable, parameter adjustments can be made through trial-and-error or heuristic techniques such as the Ziegler–Nichol's method. Feedforward and feedback controllers are another classification, which can divide to other categorize e.g., classical controllers, robust controllers, optimal controllers, centralized and decentralized controllers, etc. [174].

Table 4.2 highlights the use of numerous common control techniques in the area of UAV control and evaluates their pros and cons.

Table 4.2: Control techniques used for quadrotor stabilization and their advantages and disadvantages

Control Technique	Advantages	Disadvantages
PID	Simple implementation; model or model-free technique; low cost of computation	Not optimal, trial and error tuning
H2	Simple in implementation. Optimal control if model available	Need precise modelling without uncertainties
Hinf	Robust, uncertainties-support. implimentation simple	Calculation and adjustment are difficult.
Variable Structure Sliding mode	Simple implementation, strong stability, dealing with uncertainties	Chattering effect in control
Backstepping	Global stability, robustness, and fast response.	Necessity to full state, large control magnitude
Adaptative control	Robustness to parameter varying	Necessity to complax adaption laws
Feedback linearization	Control signal smooth	Need accurat model without parameter changing
Intelligent control	Robustness, model-free, disturbance, and uncertainties handling and learning	Heavier computing effort offline. Hard to tuning

4.4.2 Quadrotor Control Architecture

The quadrotor model can be partitioned into two primary sub-models governed by dynamic equations. The first sub-model, referred to as the outer-loop, encompasses translation variables, including latitude, longitude, and altitude (x, y, and z), and is closed by position controllers. The second sub-model, known as the inner-loop, encompasses attitude variables, namely roll, pitch, and yaw (ϕ, θ, ψ), and is closed by attitude controllers. It is essential to note that the inner-loop should exhibit a higher control bandwidth than the outer loop. The asymptotic stability of two interconnected inner and outer-loops is assured if each loop possesses asymptotic stability [175]. Figure 4.11 and 4.12 depict a schematic diagram of

quadrotor controller architecture, generally for linear controllers.

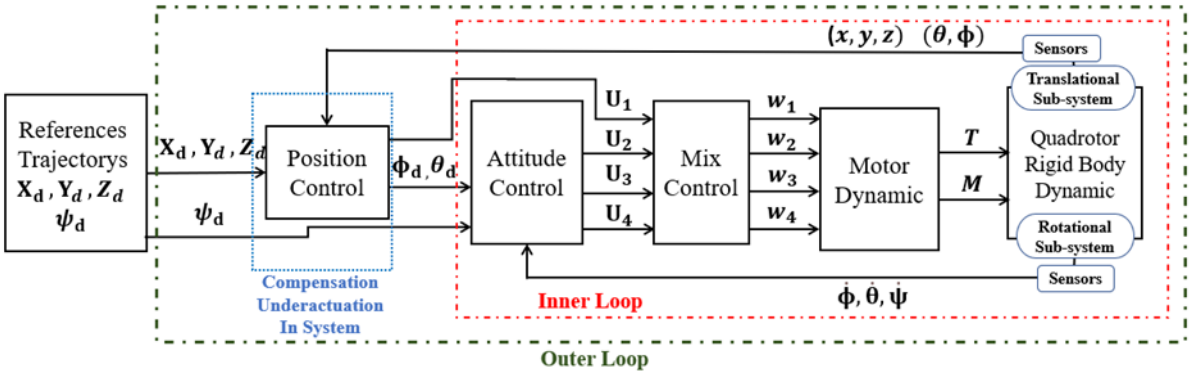


Figure 4.11: General quadrotor block diagram with two loops (inner and outer) and cascade controllers and underactuated system compensation

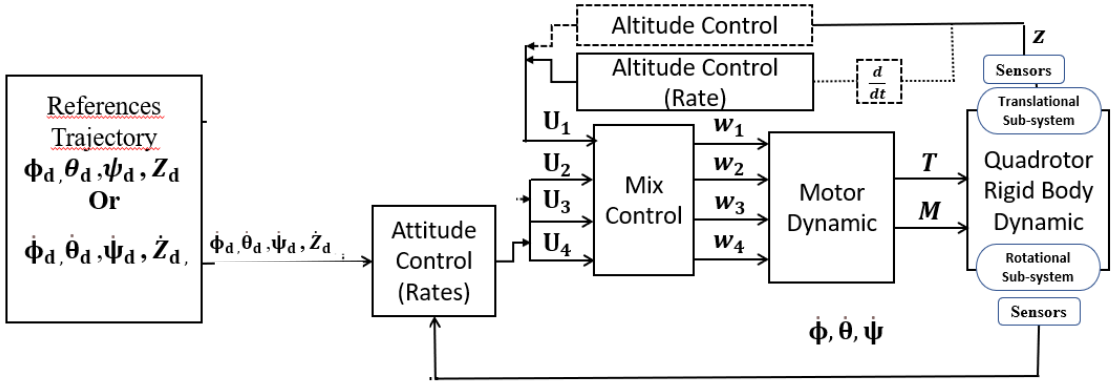


Figure 4.12: General quadrotor block diagram with a single inner loop and two control options (state control or derivative control), without underactuated constraint.

These two previous schematics show the most common architecture in quadrotor linear controllers. Whereas in the first architecture, two loops are used for quadrotor stabilization in position and velocity while tracking the desired trajectory and named "time-scale separation" and the overall system remains stable as long as the inner loop operates faster than the outer loop. This architecture is more suitable for high level tasks such waypoint, tracking etc. The position control block shown in Figure 4.11 consists of three PID controllers for x, y, and z axes. The outputs of these controllers, denoted as (U_x, U_y), are processed through an additional compensatory block. This compensatory block employs Equation 4.23 to address the Underactuation constraints within the system, specifically in relation to roll and pitch control. The result is a new

set of outputs, (ϕ_{d}, θ_{d}), which serve as the desired attitude values for the attitude control system. These desired attitude values, (ϕ_{d}, θ_{d}), then act as set-points for the controller responsible for regulating the system states $(\dot{\phi}, \dot{\theta})$ and generating the control inputs U2 and U3. The attitude controller block is utilized to maintain the orientation of the quadrotor or its respective velocities at the desired attitude (ϕ_{d}, θ_{d}). Concurrently, U1 is produced by the altitude controller, which is tasked with stabilizing the quadrotor's vertical position (variable z) in accordance with the desired trajectory. U1 is directly fed into the mixing control block.

$$\begin{cases} \varphi_d = \sin^{-1}\left(U_x \sin\,\psi - U_y \cos\,\psi\right) \\ \theta_d = \sin^{-1}\left(\frac{U_x \cos\,\psi + U_y \sin\,\psi}{\cos\,\varphi}\right) \end{cases} \tag{4.23}$$

Meanwhile, the desired Yaw (ψ_{d}) comes from the reference block. The outputs from the controller are transferred directly to the mix control block as U4. The mix control block takes the desired commands signals (U_1, U_2, U_3, U_4) and determines how to distribute these to the individual motors of the quadrotor, through a translation matrix in equation 4.9, in order to generates the respective angular velocities for all the quadrotor motors (w_1, w_2, w_3, w_4). The quadrotor motors usually represent by first order system, the perspective of the changes in quadrotor motor velocities result in thrust and moments acting on the quadrotor's rigid body. These forces directly influence the quadrotor's translational and rotational movements, which are subsequently detected by onboard sensors and filtered to derive state variables $(\ddot{x}, \ddot{y}, \ddot{z}, and\ z)$ and $(\dot{\phi}, \dot{\theta}, \dot{\psi},$ and $\theta,\ \phi,\ \psi)$ these states are depend by the kind of used sensors. All those states will be feed back to its respective controllers as shown if figures 4.11 and 4.12, in order to generate new control loop. The quadrotor system exhibits two primary strong interactions within its dynamics. The first is characterized in the inner loop, involving the interplay between altitude and roll and pitch angles. The second occurs in the outer loop, where roll and pitch angles influence the positional coordinates (x, y). Model-free approach was used, where two PID architectures were implemented to handle the stability of the whole system. Later, a gain-scheduling technique was introduced to improve the response accuracy and the quadrotor's performance. Model-based nonlinear robust control (adaptive sliding mode controller with super-twist sliding mode observer) and simulation were developed with the goal of using them in our real quadrotor once the accurate quadrotor parameters, such as

moment inertia, aerodynamic coefficients, and so on, were determined. There were attempts to identify the quadrotor dynamics using a black-box approach as an intermediary step for control design, the outcome models were used to tune the PID and LQR controllers, which gives an acceptable result. A white box identification was also used in the search for quadrotor parameters, which we planned to optimize in order to use it in model-based control approaches.

NB-1: *In our case, the accelerometer measures the acceleration states, which are* $(\ddot{x}, \ddot{y}, \ddot{z})$*, and the barometer measures the altitude* (z) *since the gyroscopes measured the rate of orientations* $(\dot{\phi}, \dot{\theta}, \dot{\psi})$*, while the magnetometer measures the headings* (ψ) *and the angels* θ, ϕ*. The other not-sensed states will be extracted in as the application required either by simple using a derivative (which not preferred) or by using integrative operation if needed. While in other applications (model-based), using state observers such as (Kalman filters, sliding mode observers, high-gain observers, etc.).*

NB-2: *Because the purpose of this study is the design, build, and control of an unmanned quadrotor aircraft, it was crucial to concentrate on the control strategies that can be implemented to overcome the problem of accuracy in quadrotor modeling and that provide the best control in the presence of parameter uncertainties, hidden dynamics, sensor noise, and other disturbances, like adaptive sliding mode (which estimated the quadrotor parameters during the control). Or it was crucial to concentrate on the so-called "model-free" strategies like PID. The third option was to focus on the control design based on the identified model, which helped in controller tuning when using Simulink tools. Taking into account the application of this designed controller to low-power processors and low-cost boards such as Arduino, STM32, and so on.*

4.4.3 Quadrotor Control Design

4.4.3.1 Linear Control

A) PID controller fixed gains and scheduling

The proportional integral derivative controller (PID) is a simple structure that is widely used in many fields to control many dynamics by calculating the error $e(t)$ between the system and the given setpoint, which is later used for the correction with the help of its structure in equation 4.24, that based on the fine-tunned of proportional K_{p} , integral K_{i} , and derivative K_{d} gains.

$$u(t) = K_{\mathrm{p}}e(t) + K_{\mathrm{i}}\int_0^t e(\tau)\mathrm{d}\tau + K_{\mathrm{d}}\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{4.24}$$

Where the simple application of this continuous-time PID in discrete application (embedded systems) is based on by the approximation of error (such as forward Euler, Backward Euler or trapezoidal etc.) a simple standard form discrete-time PID in equation 4.25.

$$U(k) = K_{\mathrm{pd}}\, e(k) + K_{\mathrm{id}} \sum_{n=0}^{k} e(n) + K_{\mathrm{dd}}(e(k) - e(k-1)) \tag{4.25}$$

Where: $K_{\mathrm{pd}} = K_{\mathrm{p}}$, $K_{\mathrm{id}} = {}^{T_s}/_{K_{\mathrm{id}}}$ and $K_{\mathrm{dd}} = {}^{K_{\mathrm{d}}}/_{T_s}$ (T_s is the loop-time, and k is the discrete step). Other important factors must take into consideration in PID design include the integral windup, the derivative low-pass filter, and the saturations [176]. In fact, the PID controller is easy to use and performs well when designed for linear systems or when operating near the system's equilibrium point, as the controller is specifically designed to work and is guaranteed to operate effectively in the vicinity of a particular operating point (with the limitation of local stabilization). The necessity to switch to different operating points requires subsequent redesigns following a series of dynamic linearizations for each point to ensure local stability. Gain scheduling is a solution that extends stability across various operating points. However, the nonlinear properties of dynamic systems make PID design and tuning more complicated, especially when the processes have to work outside their original point of equilibrium, which make it incapable of ensuring closed-loop performance in a variety of flying situations. Many solutions have been proposed to overcome this drawback, including a novel strategy-based mechanism of gain-scheduling that involves varying the PID

gains to stabilize the entire system's altitude and position. Some of these solutions are simple and effective, while others are robust and capable of handling uncertainties and external disturbances during trajectory tracking. Innovative techniques for stabilizing the UAV's attitude and enhancing its position and tracking are also explored. The vast majority of these studies, among others, are based on fuzzy logic [177-180], The hybrid approach involves adjusting PID gains in conjunction with the use of fuzzy resonance. Other used Self-tuning PID based in neural network [181]. The deferent between the Fixed-Gain Control and the Gain-Scheduled control applied for quadrotor is well-illustrated in [182]. Metaheuristic inspired algorithms have been used in various works for optimal PID controller design, where in [183] genetic algorithm (GA) has been used to achieve desired objectives such as overshoot, rise time, and settling time. Due to it adaptability as a universal optimization technique. GA has been used too in [184] to optimal tuning of backstepping controller. A comprehensive exploration of the quadrotor's PID gains and optimization was conducted through the utilization of various nature-inspired algorithms, including cuckoo search, genetic algorithms, differential evolution, and evolutionary strategies. This investigation was followed by a comparative study in which the quadrotor tracked a trajectory to demonstrate the effectiveness of these techniques. The subtle differences among these approaches have been discussed in further detail in [185].

The lookup table for gain scheduling in our simulation for quadrotor was first tuned using the trail end error method at different operation points, and then optimized using the powerful genetic algorithm to find the global solution to achieve desired objectives (overshoot and rise time).

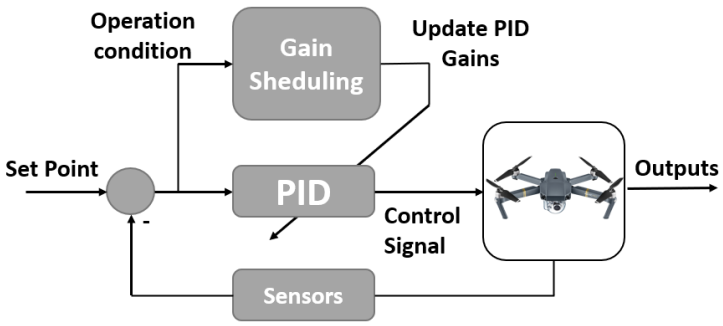


Figure 4.13: Gain scheduling structure of control

Basic PID controller tuning can be achieved either through empirical trial-and-error methods or by utilizing transfer functions that relate the control signal to the state being controlled.

B) States Feedback Control

State feedback control aims to design a control law, often represented as $U(t) = -Kx(t)$, where K is a matrix of gains, in order to control the controllable system in equation 4.25 by manipulating the state vector $x(t)$ which are fully observable. The purpose of the state feedback controller is to determine an appropriate K that stabilizes the system and achieves desired performance in close-loop. Where the closed-loop system with state feedback control can be represented by the equations $\dot{x}(t) = (A - Bk)x(t)$. To achieve stability, the eigenvalues of the matrix $(A - Bk)$ must all have negative real parts. The choice of gains in the matrix K is crucial and depends on the specific requirements and constraints of the control system.

If the A, B, C, and D matrices of the system are available, we use the pole placement technique. This technique involves selecting appropriate gains for the matrix K to place the closed-loop poles (eigenvalues of $(A–BK)$) at desired locations in the complex plane. This is done to achieve the desired closed-loop performance, such as a faster response and minimal overshoot. However, if the system's A, B, C, and D matrices are not available, we may resort to a trial-and-error technique for tuning the feedback gains.

C) Linear Quadratic Regulator LQR

Multiple efforts have been made in quadrotor to employ LQR [186-187], as optimal controller, despite the fact that this controller provides an optimal control only when the quadrotor is at hovering condition (where the system has been linearized). When constructing the LQR controller, the state space of the quadrotor model must be determined before. Experimentally validated in trajectory tracking in [188-189] using benchmark quadrotor Qball-X4, demonstrating its effectiveness compared to traditional PID in quadrotor stabilization while tracking desired trajectories using optimal feedback control law.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \\ J = \frac{1}{2}\int_0^{\infty}(x^T Q x + u^T R u)dt \end{cases} \quad (4.25)$$

Where, A, B and C are the state space matrixes, for states, input and output respectively. (*Q* and *R),* are two weight matrices that depend on the system's states and control performance. The optimal control feedback gains, represented as the feedback gain matrix (K), minimize the cost function (J). The matrix K is defined as follows:

$$K = R^{-1}B \times P \tag{4.26}$$

With *(P)* being the solution matrix to the following Riccati equation:

$$A^{\mathrm{T}}P + PA - PBR^{-1}B^{\mathrm{T}}P + Q = 0 \tag{4.27}$$

The closed-loop control system with the LQR controller becomes:

$$U = -Kx \tag{4.28}$$

In practice, obtaining an accurate representation of a quadrotor rotor is a difficult task, due to model errors and hidden dynamics, which will have an impact on the determination of optimal control gains for LQR, the disturbance in the system and sensor noises are other factors that may degrade the robustness of the LQR solution. The switching from one operation point to another is another factor that impacts LQR response. All of these issues have piqued the interest of the research community. Determining the weighting matrices Q and R of the cost function using empirical techniques for stabilizing a quadrotor has been treated in [170] using Bryson's method, where used only the prior knowledge about maximum acceptable values of states and control inputs, and meet requested the performance from LQR. Other [190] overcomes this drawback by utilizes an adaptive control LQR with model reference and has improved tracking accuracy and stability against parametric uncertainties.

C-1) Weighting Matrices Q, R Tuning

The LQR controller can be an optimal solution when an accurate system model is available. In this scenario, the cost function is minimized through the adjustment of weighting matrices Q and R, resulting in the desired outcome. However, the reality is that the system models obtained through identification processes are often far from perfect and may contain inaccuracies and be unable to account for all possible system

behaviours. As a result, and to overcome this, there are several techniques used for tuning the LQR weighting matrices Q and R, such as:

Integrating the dynamic model into a closed-loop system using the LQR and simulating the system, often utilizing software such as MATLAB. This process centres around adjusting the weighting matrices Q and R until the desired control responses are achieved. This adjustment can be carried out manually, where the values of Q and R, are tailored to meet the specific requirements of the control system and states, a method known as "Bryson's Rule," which will be elaborated on later. Alternatively, the adjustment can be accomplished through a trial-and-error technique. Or the using both of them. While this approach is straightforward, it may consume time and not consistently yield optimal performance.

Another option is to employ optimization algorithms, such as genetic algorithm, Particle Swarm Optimization (PSO), and others, to optimize the weighting matrices Q and R. This optimization process entails iteratively searching for the best values through random mutations and selections.

The feedback gain matrix K is fine-tuned using simulation tools like MATLAB, which incorporate the system's dynamic model into a closed-loop LQR control simulation. This process assesses the controller's performance and system response in various scenarios. Through iterative adjustments, the feedback gain matrix K is optimized until the desired control objectives are met. Once determined, this K matrix is applied to the real system (quadrotor) to enable real-time control, allowing the system to respond to external disturbances and maintain stable flight.

C-2) Tunning Weighting Matrices Q, R by Bryson's Rule

Bryson's Rule is a heuristic that is commonly used in control systems to estimate the required damping ratio of a control system in order to achieve satisfactory performance. As per this principle, the matrices Q and R are represented as diagonal matrices, with their diagonal elements expressed as the inverse of the maximum permissible values of the "x" state variable and the "u" input control variable, squared. The diagonal elements of matrixes $\bar{Q}_i$, $\bar{R}_j$ of the matrixes Q and R can therefore be represented as following:

$$\bar{Q}_i = \frac{1}{\text{maximum acceptable value of } (x_i^2)} \text{ and } \bar{R}_j = \frac{1}{\text{maximum acceptable value of } (u_j^2)} \quad (4.29)$$

$i = (1,2,3 \ldots n)$ and $j = (1,2,3 \ldots m)$. Where (n, m) are the states and control dimensions. corresponding then to the following cost function

$$J_{LQR} = \int_0^{\infty} \left(\sum_{i=1}^{n} \bar{Q}_i x_i(t)^2 + \rho \sum_{j=1}^{m} \bar{R}_j u(t)^2 \right) dt \quad (4.30)$$

The representation provided earlier is one of the commonly used forms of Bryson's Rule. Below, we present the general form of Bryson's Rule for tuning the weighting matrices Q and R as follows:

$$\mathbf{Q} = \text{diag} \left\{ \frac{\alpha_1^2}{x_{1\max}^2}, \frac{\alpha_2^2}{x_{2\max}^2}, \cdots, \frac{\alpha_n^2}{x_{n\max}^2} \right\},$$
$$\mathbf{R} = \rho \ \text{diag} \left\{ \frac{\beta_1^2}{u_{1\max}^2}, \ \frac{\beta_2^2}{u_{2\max}^2}, \ \cdots \frac{\beta_m^2}{u_{m\max}^2} \right\} \quad (4.31)$$

The $x_{i_{\max}}$ and $u_{j_{\max}}$ represent the largest state response and control input respectively. To add an additional relative normalisation on the various components of the state or control, one can add the constraints:

$$\sum_i \alpha_i^2 = 1 \text{ and } \sum_j \beta_j^2 = 1 \quad (4.32)$$

ρ used for relative weighting between the control and the state influence. Some authors simply set all α_i and β_i and ρ to 1.

Note that Bryson's Rule is just a rough estimate and that more detailed analysis may be required to determine the optimal damping ratio for a particular system.

4.5 Conclusion

In conclusion, the dynamic modelling of a quadrotor can be modelling through Euler-Lagrange and Newton-Euler methods, including the use of quaternions for representing orientations. These models capture the nonlinear dynamics of the quadrotor and incorporate the under-actuated constraint. Additionally, angle approximation techniques can be used to linearize the model for control purposes. Dynamic model identification plays a crucial role in understanding and characterizing the quadrotor's behaviour. Various model identification approaches such as black box direct methods and adaptive neuro-fuzzy inference systems can be employed for system identification. It is important to carefully select inputs, consider the system bandwidth, and design a suitable model structure to achieve accurate identification. Quadrotor control techniques involve stabilization during tracking to ensure safe and reliable flight. Control architectures are designed to manage the quadrotor's behaviour, and schematic diagrams can be used to depict the controller's structure. Linear control techniques such as PID controllers and linear quadratic regulators (LQR) and states feedback control are commonly employed. Tuning of the LQR weighting matrices Q and R can be done using different techniques. Nonlinear control methods, such as sliding mode control, offer robustness and accuracy in quadrotor control. Various types of sliding mode controllers exist, providing versatility and flexibility in control design. A study case of quadrotor altitude control using sliding mode control can be explored, considering the design of a sliding mode observer such as the proposed super-twisting observer. Overall, the understanding and application of dynamic modelling, identification, and control techniques are essential for achieving stable and accurate control of quadrotors. These approaches offer the potential to address the complexities and challenges associated with quadrotor dynamics, enabling safe and efficient operations in various applications.

CHAPTER 5

EXPERIMENTAL RESULTS AND DISCUSSION

Contents **Page**

5.1 Introduction

In this section, we concluded our research by exploring the experimental and validation aspects of this study, where we explored the entire process of quadrotor designing in-depth, testing, and evaluating a quadrotor system. Commencing with the process of quadrotor sizing, we employed stress and deformation analysis, quadrotor sizing using cluster technique, and other flight evaluation tools. Subsequently, our pursuit led us to make informed choices in quadrotor components, encompassing the global quadrotor system design, including various critical components such as flight control, sensors, actuators, power supply, communication devices, telemetry systems, and the intricate global interconnection of these elements and the algorithm used. To reduce the impact of vibrations, we employed the FFT technique as an analyser tool. This experimental phase comprised the balancer of the propellers and BLDM motors in static and dynamic conditions, leading to a significant decrease in the intensity of vibrations. Furthermore, the calibration of sensors yielded notable enhancements in measurement accuracy for IMU and compass. The filtering process improved the filter accelerometer and gyroscope data using complementary filter coefficients. A comparison between raw and filtered sensor measurements offered a clear perspective on the advantages of filtration. Moving forward, we delved into identification techniques, encompassing White/Grey-Box and Black-Box methods, to extract essential parameters governing the quadrotor's behavior. Within these results, the successful design and re-tuning of PID controllers, as well as full-state feedback controllers, were apparent, effectively tracking a reference. Finally, the validation of quadrotor attitude control was demonstrated through stable hovering.

5.2 Quadrotor Sizing

The experimental chapter builds upon the theoretical foundation established in the theoretical chapters, where in the first phase of quadrotor design, careful consideration is given to the specific requirements and objectives of the project. This includes identifying the essential features and functionalities that the quadrotor should possess, as well as any limitations or challenges that may arise during the design and construction process. The second phase of quadrotor design follows the determination of sizing requirements. In this phase, sizing and optimization tools such as linear regression and cluster representations are employed. These tools aid in identifying the

combination of quadrotor components that meet the predetermined design goals. By leveraging these techniques, the quadrotor's components can be carefully selected and configured to ensure performance and efficiency.

Here we will build upon these theoretical insights, providing a practical implementation of the design process. Through experimentation and data collection, we will demonstrate how the chosen design approach aligns with the theoretical framework. The experimental results will validate the effectiveness of the design choices made in terms of achieving the desired features, meeting the design goals, and overcoming the identified constraints.

5.2.1 Quadrotor Design Initialisation

As mentioned in the theories chapter that the building of quadrotor is begin by, determine the required features and design goals "*xi*". Includes specifying the payload capacity, flight time, maximum flight speed, and wind resistance, among other factors. In addition, based on our methodology design flowcharts for quadrotor in Figure 1.9 and 1.10 we present the following:

Target goals:

- Open-source platform for educational purposes. *x1*
- Support tasks that necessitate high processing (fuzzy logic PDC, etc). *x2*
- Assist with communication, data transfer, and saving. *x3*
- The ability to detect its location, heading, altitude, attitude, etc. *x4*
- Support for future additional devices, such as a flight computer, a camera, a hand, etc. *x4*
- Indoor and Outdoor flighting x*5*
- Non-adaptive morphologies. *x6*
- Good shock resistance. *x7*
- Stability and control of the quadrotor, it can maintain stability in the air and respond to control inputs from the operator. First task is hovering ability Second is trajectory follows, third is obstacle avoiding. *x8*
- Operating conditions, the quadrotor will be flown in, such as wind, temperature, and altitude, which can affect its performance. *x9*
- The ability for remotely controlled and autonomous flight (flight computer) x*10*

Constraints:

- Only components that meet the budget can be used. C1
- Legal requirements, guidelines of governing the use of drones and unmanned aerial vehicles, depending on the location and mission and necessities an authorisation. C2
- Almost of the quadrotor components are not available in Algeria. C3
- Many other constraints. C4

Target components:

- Low-cost. T1
- Those available in Algeria is preferred. T2

Initial Design:

- Quadrotor with legs and protectors. I1

Target features:

- Payload capacity about: 1.5 kg. *x11*
- Flight time about: 10 minutes. *x12*
- Maximum flight speed about: 20 m/s. *x13*
- Range about: 100 m. *x14*
- Altitude about: 30m. *x15*
- Dimensions about: 50 cm. *x16*

These are just some of the many features that we might consider when sizing a quadrotor, based on these requirements, we can start sizing the quadrotor components beginning by initial estimate for quadrotor components.

5.2.2Quadrotor Components Sizing Using Cluster Technique

Based on the stated requirements **target goals** {*x1 … x10*} and **initial Design** aim {*I1*} and **target features** {*x11…x16*}, we can initiate the process of determining the appropriate sizes for the initial quadrotor components. However, it is important to keep in mind that these results are just an “initial estimate” and the final selection of

components and their characteristics will be based on other factors such as constraints and specific target goals. Let give some examples of estimation of quadrotor components using cluster technique.

To illustrate the selection of motor variables discussed in the theoretical part, we can examine the second design trend related to motor variables. Specifically, we consider motor weight, max-thrust (motor/prop), motor diameter and height, motor KV value, and quadrotor frame size, as presented in Figure 5.2. Through our analysis, we have discovered a significant correlation between these variables, and we have formulated a relationship model, which is outlined in Table 1.5 in the theoretical section and presented in the flowchart Figure 5.2. The model displays the following insights:

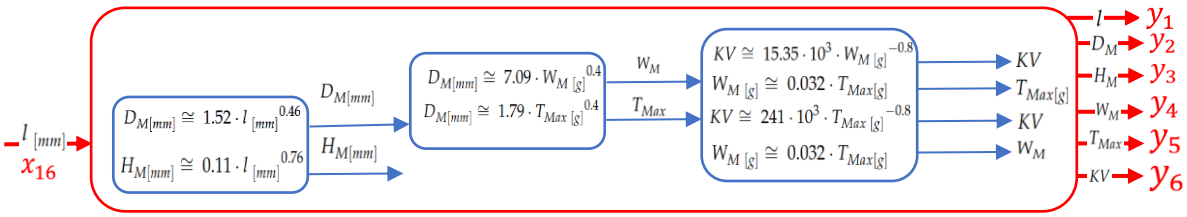


Figure 5.1: Sizing box for second design trend related to motor variables.

Table 5.1: Numerical Solution of Motor Variables second design trend

$x_{16} = y_1(mm)$	y_2 (mm)	y_3 (mm)	y_4 (g)	$y_2 = y_3$ (g)	y_5 (g)	y_6 (kv)
500	26.50	12.37	27.00	26.98	845.30	1099

In our analysis of target features, our primary focus was on the input variable x_{16}, which represents the dimensions of the quadrotor frame size. Specifically, this dimension is approximately 50 cm, denoted as x_{16}=50 cm and referred to as $l_{[mm]}$ in the equation. We utilized this value as an input for the sizing box depicted in Figure 5.2. This particular box encompasses all the equations associated with the "second design trend," which are directly related to motor variables. The resulting outputs from this box are $(y_1 \ldots y_6)$ presented in Table 5.2 represent various other quadrotor features. These outputs are subsequently utilized as inputs for subsequent boxes, such

as the third design trend, which deals with the operational parameters of the motor and propeller, as illustrated in Figure 5.2.

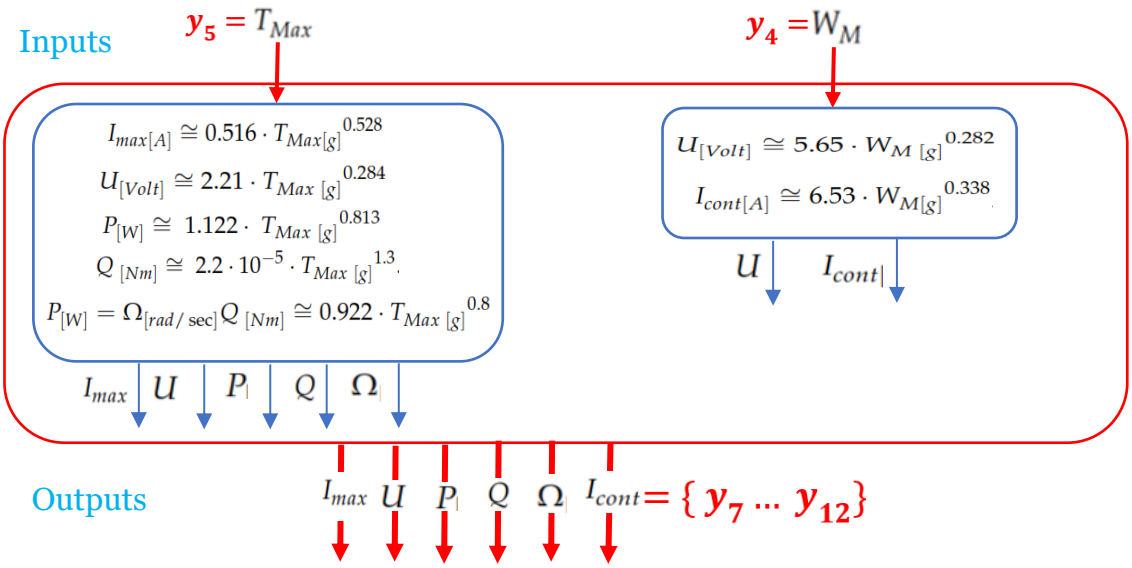


Figure 5.2: Sizing box for third design trend related to motor/prop operational parameters.

Table 5.2: Numerical Solution of motor/prop operational parameters

$y_7(A)$	y_8 (V)	y_9 (W)	y_{10} (Nm)	y_{11} (rad/s)	y_{12} (A)
18.09	14.97-14.31	268.42	0.14	1917-1443	19.89

Based on the data presented in Table 5.1 and Table 5.2, we can draw conclusions about the appropriate motors for a quadrotor with a size of 50cm and a weight exceeding 1.5 kg. The ideal motors would have a diameter of approximately 26.5mm, a height of around 15.37mm, and a weight of approximately 27 grams. Their KV value should be around 1099 rotations per volt, enabling them to generate a thrust of 843 grams. Furthermore, based on these motor characteristics, we can estimate the power consumption. At maximum thrust, we anticipate a motor-prop current consumption of about 18A, and the maximum continuous current trend-line for all motors is expected to be around 19.89A. Additionally, the voltage consumption is projected to be approximately 14V, considering either the motor's maximum thrust or its weight. The expected power output of each motor is around 268W. These estimations provide a general overview of the suitable motor specifications for our quadrotor.

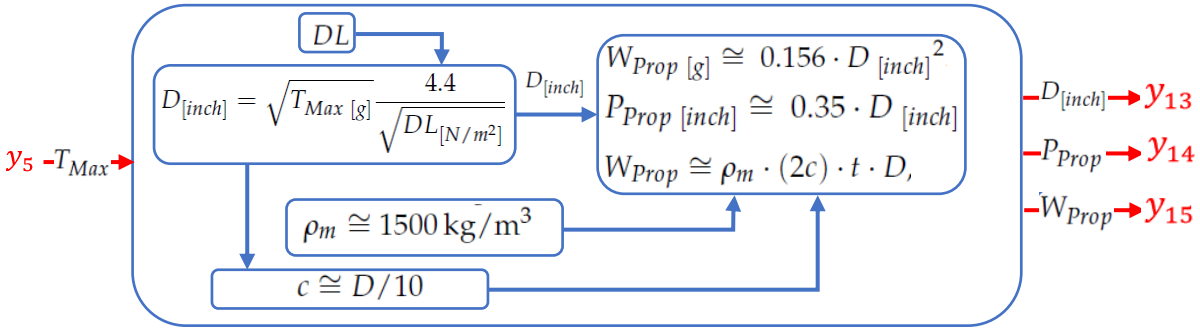


Figure 5.3: Sizing box for design trend "Propeller Diameter" and "Weight and Pitch"

Table 5.3: Solution of first design trend "Propeller Diameter" and "Weight and Pitch"

$y_{13}(inch)$	y_{14} *(inch)*	y_{15} *(g)*	c*(inch)*	**DL** *(N/m^2)*
11.206	5.922	17.48	1.12	130

Based on the requirements and specifications of the quadrotor, the appropriate propeller would have a diameter of approximately 11.2inches and a pitch of around 5.9 inches. This propeller should weigh approximately 1.75 grams. These dimensions and weight are selected to achieve the desired thrust, stability, and efficiency for the quadrotor's performance.

The process of selecting other quadrotor parameters, including propeller rotational velocity, battery characteristics, payload, propeller diameter, weight, and pitch, follows a similar methodology. By applying this process, we can envision an appropriately designed quadrotor that aligns with our target features and goals.

5.2.3 Quadrotor Component Selection Process and Analysis

The final selection of quadrotor components is a result of a comprehensive process that takes into account the characteristics obtained from the previous step, which involved sizing based on clustering or design trends. Additionally, the selection process considers the constraints (*C1…C4*) and the target components (*T1, T2*) outlined earlier. The process involves carefully evaluating the available components that meet the required specifications and align with the quadrotor's target goals and constraints. It takes into consideration factors such as budget limitations, legal requirements governing drone use, component availability in Algeria, and other specific constraints identified during the design process.

The final selection is made by comparing and assessing various component options, weighing their compatibility with the quadrotor's requirements and the constraints imposed. Components are chosen based on their ability to meet the desired payload capacity, flight time, maximum flight speed, range, altitude, and dimensions.

Frame: To accommodate the payload and other components, we would need a rigid frame non-adaptive (*x6*) with dimensions of around 50 cm (*x16-x4*). A quadrotor frame made of carbon fibre or aluminium or adequate plastics would provide the necessary and stability (*x7*). The frame of a quadrotor is indeed an essential component and plays a crucial role in providing structural integrity and stability to the overall system. There are two main options for acquiring a quadrotor frame: building it using available materials such as carbon-fiber or aluminium, or purchasing a commercially available frame.

The commercially available frames have the advantage of being well studied and designed by manufacturers who specialize in quadrotor components. These frames undergo rigorous testing and optimization to ensure their performance and reliability. Commercial frames come in various sizes and designs, catering to different quadrotor configurations and payload capacities.

The following Table 5.4 showcases three well-known commercial quadrotor frames as candidates, highlighting their characteristics such as width, weight, and cost. Additionally, a radar chart in Figure 5.4 illustrates a comparison between these frames based on their performance against target goals.

Table 5.4: Commercial quadrotor frames

F450	**F550**	**Tarot TL65S01 650**
		

• Width: 450mm • Weight: 395g • Super strong & smooth • Price: 3,86 € • Shipping: 14.27 € • Total Price: 18.13 €	• Width: 550mm • Weight: 630 ±2 g • Full carbon fiber • Price: 62,80€ • Shipping: 10.90 € • Total Price: 73.7€	• Width: 600mm • Weight: 1000g • Full carbon fiber • Price: 119,16 € • Shipping: 32.96 € • Total Price: 152.12 €

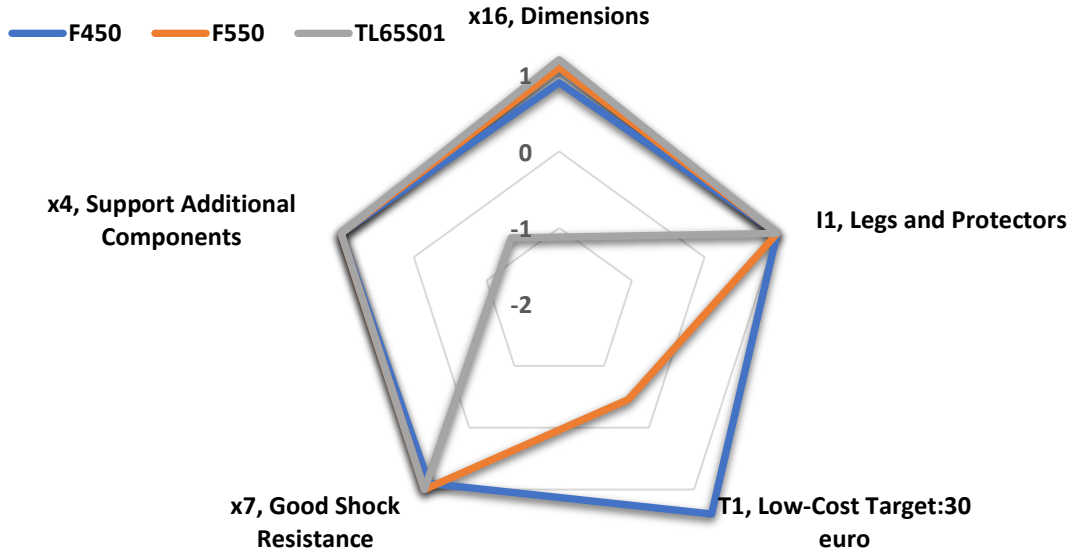


Figure 5.4: Comparison of the commercial quadrotor frames

Based on the comparison of the commercial quadrotor frames, the analysis indicates that the F450 frame emerges as the best choice. This conclusion is drawn from evaluating several factors, including its cost (*T1*), shock resistance (*x7*), dimensions (*x16),* leg and protectors (*I1*), and support for additional components (*x4*). The F450 frame exhibits favourable performance in these areas, making it the optimal selection among the available options.

Motors: Because the payload target capacity is about 1.5 kg (x11). The total thrust of the select motors is preferred to be twist the payload target capacity (3 kg), which make each motor have at minimum thrust of 750 g of each (we found y5=845.30 g). It is important to consider the desired payload capacity. If the payload capacity is high (3 kg), larger and more powerful motors may be required to lift the weight and maintain stability. In this case, a lower KV rating may be necessary to generate the required torque. In addition, this can be achieved by using brushless DC motors with a KV value around 800-1000 KV (we found y6=1099 KV). These motors would have a power rating of around 1000W each.

Propellers: The propeller diameter (y13) calculated using the formula for propeller thrust efficiency. The propellers should be able to generate enough thrust (y5) to carry the payload (x11) and maintain stability during flight (x8). A larger diameter propeller (y13) will typically generate more thrust (y5), which is necessary to carry a heavy payload of 1.5 kg at less. Propellers with a larger diameter tend to be more efficient, which results in longer flight times (x12). In this case, the goal is to achieve a flight time of 10 minutes, so choosing propellers with a larger diameter can help to achieve this goal. In addition, the maximum flight speed (x13) of 20 m/s requires adequate propellers (y13, y14, y15), which can be achieved with larger diameter and adequate pitch (y14) propellers. In general, propellers with an advance ratio of around 0.5 are considered to be the most efficient, as they provide a good balance between thrust and efficiency. The propellers around 11-inch (y13) with 4-inch (y14) of pitch propellers chosen should have a high pitch, which will provide the necessary thrust to maintain stability during flight (x8). The material of the propellers is also important, as lightweight materials (y15) such as carbon fibre or plastic provide better efficiency.

Battery: To achieve the desired flight time of 10 minutes, we would need a battery with a capacity of around 5000mAh. To ensure the safety of the system, we could use a LiPo battery with a C rate of at least 25C. Or battery of 3700 mAh with CC of 80CC.

Electronic Speed Controllers (ESCs): The ESCs should have a continuous current rating of at least 30A and a burst current rating of 40A or higher. They should also have a built-in voltage regulator to ensure stable operation.

Flight Controller: A flight controller should have a fast-processing speed, multiple inputs and outputs, with advanced features such as GPS, altitude hold, and position hold would be suitable for this system. The flight controller should be compatible with the other components and have a sufficient processing power to handle the required calculations.

Radio Control System: A radio control system with a range of at least 100 m would be required to control the quadrotor. We could use a 2.4GHz radio control system with a minimum of 8 channels to provide enough control options. We can choose between a traditional analog radio controller or a digital system such as the FrSky Taranis or the Spektrum DX9.

These are the main components and specifications of a quadrotor system designed to meet the target features outlined previously. Keep in mind that the specific

components and specifications may vary depending on the exact requirements or new constraints or specific target goals.

Table 5.5 presents a range of options that are in line with the quadrotor sizing results. These options showcase a selection of commercialized quadrotors, each consisting of a combination of components carefully chosen to meet the required specifications.

Table 5.5: Some commercial quadrotors components combination meets the required specifications

N	Motor	ESC	Propeller	Battery	Frame Size	Hovering Time	Load Weight	Total Weight
1	EMAX XA2212-980KV	EMAX Simonk 20A	APC 10x4.7	LiPo 2S-7.4V-35C-3400mAh	450mm	17.2min.	0.5kg	1kg
2	EMAX MT2208II-2000KV	EMAX BLHeli 20A	HQ 5x4	LiPo 4S-14.8V-40C-	230mm	17.1min.	0.3kg	1kg
3	EMAX MT2206II-1900KV	EMAX BLHeli 20A	HQ 5x4.5	LiPo 4S-14.8V-35C-	230mm	16.8min.	0.3kg	1kg
4	SunnySky X2204-1800KV	Hobbywing XRotor 20A	GWS 8x4	LiPo 3S-11.1V-45C-3900mAh	360mm	17min.	0.4kg	1kg
5	JFRC U2204 KV1800	Hobbywing XRotor 20A	GWS 8x4	LiPo 3S-11.1V-40C-3900mAh	360mm	17.1min.	0.4kg	1kg
1	T-MOTOR F40 KV2300	**T-MOTOR AIR 40A**	4*4.5 DAL	LiPo 4S-14.8V-35C-	180mm	17min.	0kg	1.3kg
2	**EMAX XA2212-980KV**	EMAX Simonk 20A	APC 9x6	LiPo 3S-11.1V-40C-4000mAh	400mm	17min.	0.6kg	1.3kg
3	EMAX MT2208II-2000KV	EMAX Simonk 25A	HQ 6x4.5	LiPo 3S-11.1V-40C-5200mAh	270mm	17min.	0.5kg	1.3kg
4	JFRC U2212 KV750	Hobbywing XRotor 10A	APC 9x4.7	LiPo 4S-14.8V-40C-	400mm	17min.	0.5kg	1.3kg
1	DJI E310-2312 KV960	DJI E310-20A	DJI 9.4x5	LiPo 3S-11.1V-35C-3100mAh	420mm	17.1min.	0.7kg	1.5kg
2	JFRC U2208 KV1100	Hobbywing XRotor 10A	APC 8x5.8	LiPo 4S-14.8V-35C-	360mm	17min.	0.7kg	1.5kg
3	EMAX XA2212-980KV	EMAX Simonk 25A	APC 10x4.7	LiPo 3S-11.1V-45C-4400mAh	450mm	17.1min.	0.7kg	1.5kg
4	JFRC U2216 KV1100	Hobbywing XRotor 20A	APC 9x4.5	LiPo 3S-11.1V-45C-3800mAh	400mm	17min.	0.6kg	1.5kg
5	EMAX XA2212-820KV	EMAX Simonk 20A	APC 11x4.7	LiPo 3S-11.1V-40C-4000mAh	490mm	17.1min.	0.7kg	1.5kg
15	EMAX MT3110-700KV	EMAX Simonk 30A OPTO	Carbon 11x4.7	LiPo 4S-14.8V-55C-4000mAh	490mm	10.3min.	1kg	2.31kg

The selection process for the quadrotor components is completed by identifying a low-cost component (T1) that aligns with the budget constraint (C1), overcomes all other constraints (C4), and preferred available in Algeria (T2). Our executable application that summarizes the sizing calculations is our GitHub. This crucial component will be further discussed in the following section, starting with the quadrotor frame selection.

5.2.4 Quadrotor Configuration and the Frame Analysis

Our study examines conventional quadrotors with non-adaptive morphologies, handheld micro-UAVs, for educational purposes. Classed under rotor-wing UAV, rotorcraft motorized and heavies than the air with cross configuration "X" one, build from strong plastic frame, rectangular beam, medium size, as showed in Figure 5.5 the frame used for this study, “F450”, is a strong plastic and glass-fiber frame appropriate for rapid prototyping purposes. It has two layers for retaining components and is employed by several experimental researchers. The 50 cm diameter of the frame's arms provides an extensive range until 13-inch propeller diameters since the recommended the ratio of the distance between the propellers divided by the propeller radius be greater than 1.357 as mentioned before [26].

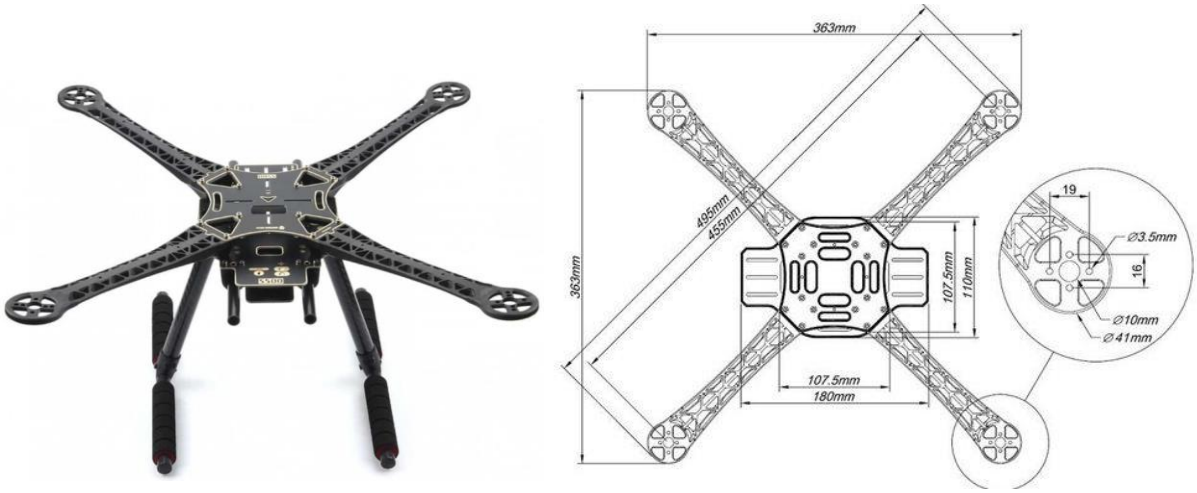


Figure 5.5: Our Quadrotor frame F450 dimensions

5.2.4.1 Frame Stress Analysis

The frame of the quadrotor prototype, including the plates and arms, is analysed structurally in [206] using ANSYS 17.0 software and the Von-Mises stress approach. showing that 160 MPa is the maximum tensile strength of lower plate which is made

from material PA66GF30 “Reinforced Polyamide” 66% and 30% “Reinforced glass fiber” its characteristic is presented in Table 5.6.

Table 5.6: PA66GF30 Reinforced Polyamide- glass fiber characteristic

Frame plates Material	Density (Kg/m3)	Tensile Strength (M Pascal)	Weng Modulus (M Pascal)	Poisson Ratio
PA66GF30	1370	160	15000	0.36

As indicated in Figure 5.6, the greatest equivalent stress attained is 5.99 M-Pascal. Therefore, the foundation plate can safely support the loads. As illustrated in Figure 5.6, the highest equivalent stress produced for the upper plate is 2.255 M-Pascal. Therefore, the higher plate can safely support the loads.

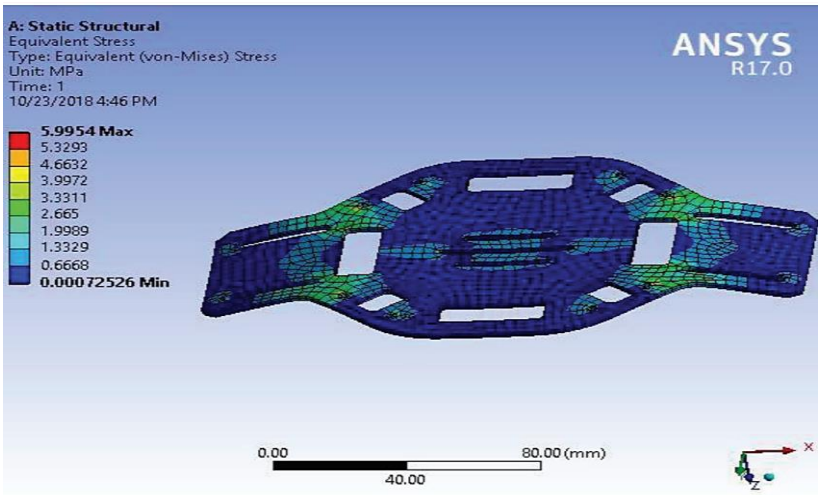


Figure 5.6: Center lower plate von-mises stress analysis with tensile-load of 20 Newton

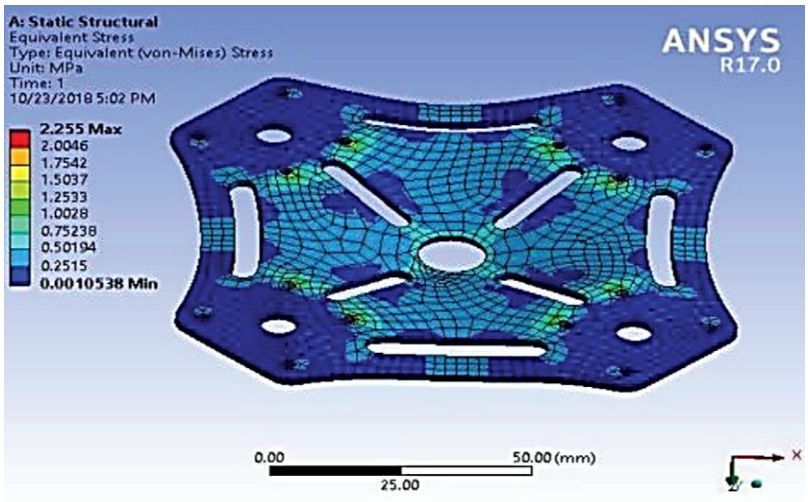


Figure 5.7: Center upper plate von-mises stress analysis with tensile-load of 20 Newton

As indicated in Figure 5.8, the greatest equivalent stress attained is 25.2 MPa. Therefore, the F450 frame can safely support the loads.

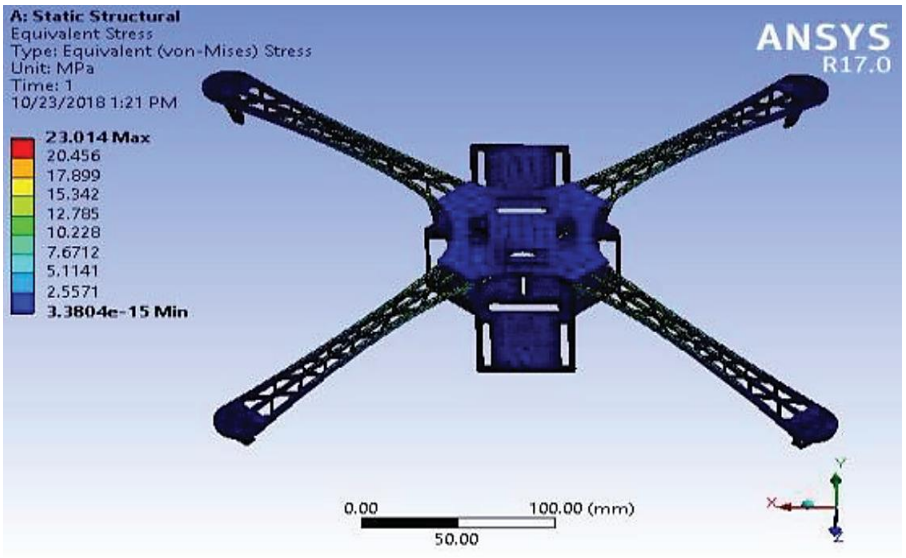


Figure 5.8: Quadrotor frame F450 von-mises stress analysis with tensile-load of 20 Newton

5.4.1.2 Frame Deformation Analysis

As seen in Figure 5.9, 5.20, and 5.21, the maximum total deformation in the lower plate, upper plate, and strut is 0.14025 mm, 0.0097 mm, and 4.1335 mm, respectively.

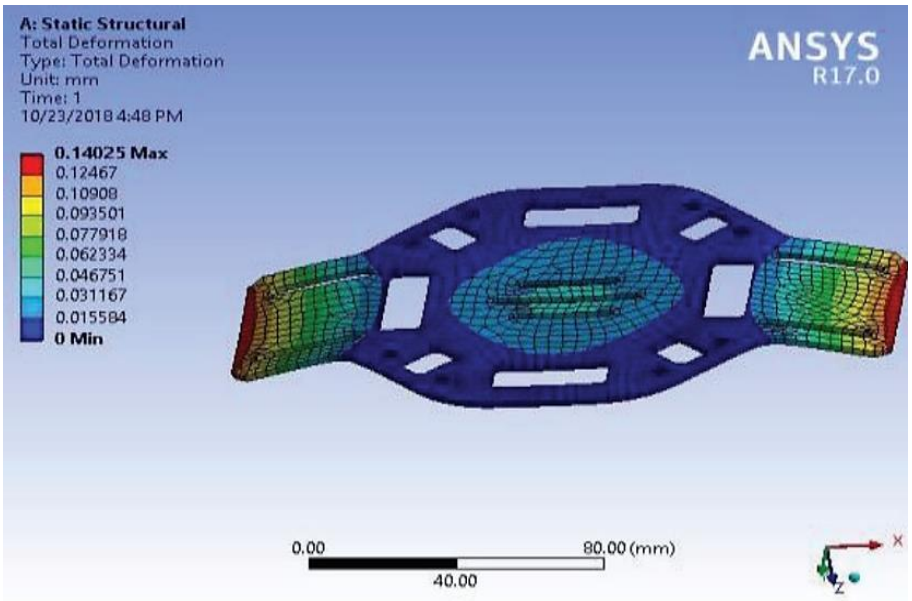


Figure 5.9: Center lower plate deformation analysis to tensile-load of 20 Newton

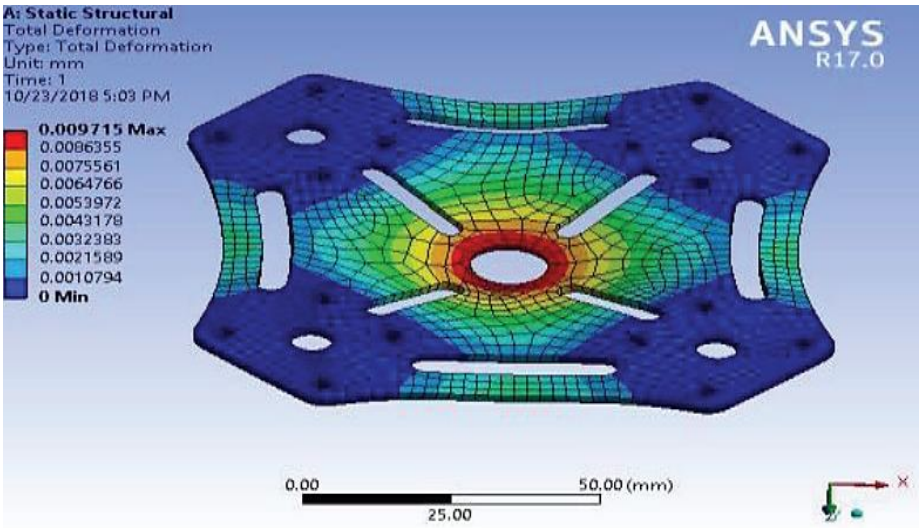


Figure 5.10: Center upper plate deformation analysis to tensile-load of 20 Newton

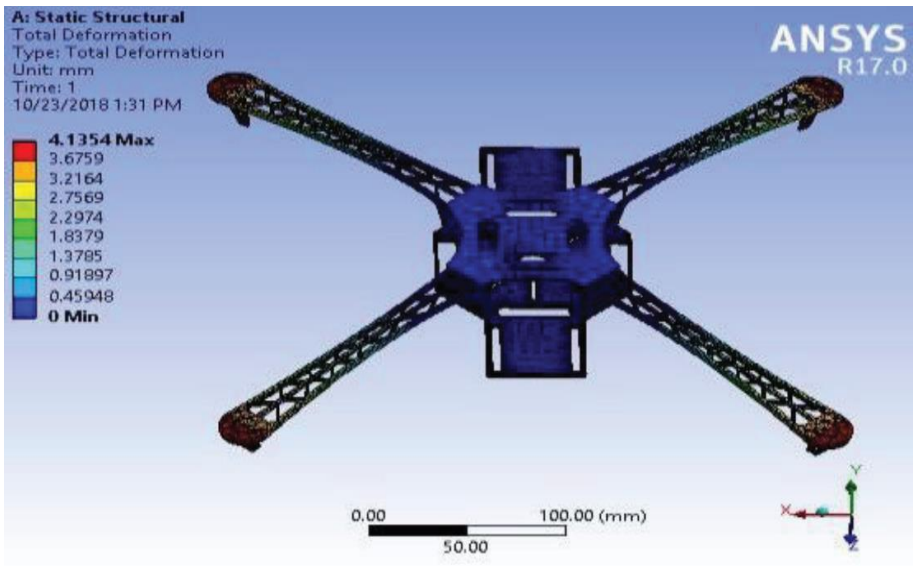


Figure 5.11: Quadrotor frame F450 deformation analysis to tensile-load of 20 Newton

A static structural study is performed in this analysis to determine the deformation of the frame when a load is applied. where the findings fall within the acceptable range when compared to the PA66GF30 materials. Therefore, the quadrotor frame is acceptable for use, and it can resist shocks very well.

5.3 Quadrotor Hardware Design and Configuration

Taking the aforementioned requirements into account, we recognized a reasonable solution in the use of microcontrollers, actuators, IMU including accelerometers and gyroscopes, barometers, magnetometers, and voltage-current level sensors, considering the introduction of new elements for enhancing the sensing/actuator resources.

Our quadrotor comprised of three key systems, namely the flight-controller, propulsion system, and ground station as presented in Figure 5.12. The component selection process involves prioritizing between the flight-controller part and the propulsion system to ensure accurate component selection. This prioritization ensures that the chosen components for both the flight-controller and propulsion system are carefully evaluated and selected with precision.

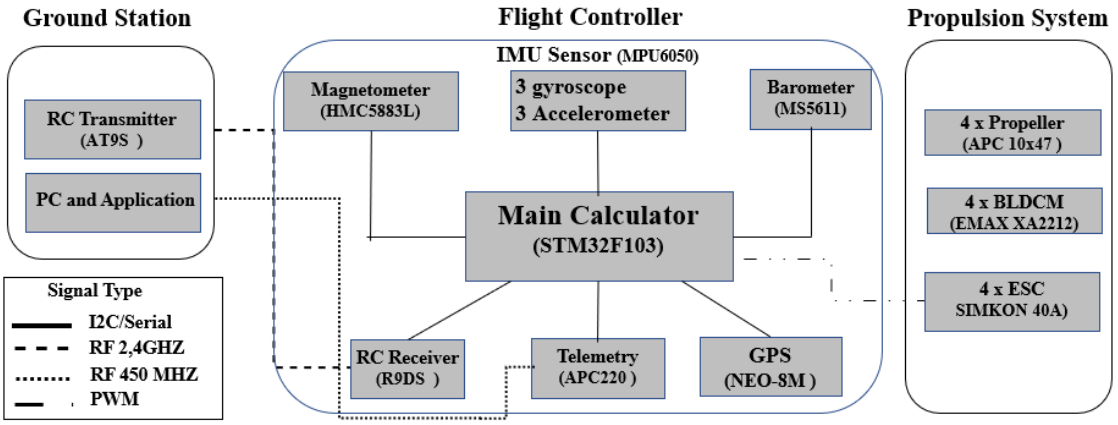


Figure 5.12: Quadrotor Component and Architecture Design Overview

The logical sequence dictates starting with the selection of flight controller components, including sensors, microcontrollers, and batteries. Subsequently, the components for the propulsion system are chosen. This approach allows for the determination of the total weight of the quadrotor prior to selecting the motors and propellers, which are responsible for carrying this weight. It is important to consider the weight of the motors and propellers themselves in this assessment. While the ground station is not included in our optimization process, its selection is based on the system requirements. The ground station, which serves as the control and communication hub for the quadrotor, is chosen based on our requirements and functionalities. such as the desired range of control, these requirements are taken into

consideration when we selected the appropriate ground station for the quadrotor system.

The Flight-Controller is a critical component of the UAV and is equipped with an Inertial Measurement Unit (IMU) MPU6050, which integrates three accelerometers and three gyroscopes. This IMU is complemented by a barometer MS5611, a magnetometer HMC5883L, and voltage-level sensors to provide the necessary measurements for navigation and control. The UAV is also equipped with a Global Positioning System (GPS) module NEO-8M and a telemetry system APC220 for communication and monitoring. An RC Receiver R9DS is also integrated to receive control signals from the ground station.

The Propulsion System of the quadrotor consists of four BLDC motors EMAX XA2212, which are coupled with four propellers APC10X47. The motors are managed by four Electronic Speed Controllers (ESCs) SIMKON of 40A, ensuring the efficient operation of the motors and propellers. The entire propulsion system and flight-controller are powered by a Lithium Polymer (LiPo) battery.

The Ground Station comprises an RC Transmitter AT9S and a personal computer, which provide the operator with the necessary control and monitoring capabilities. The RC Transmitter AT9S sends control signals to the RC Receiver R9DS onboard the UAV, while the personal computer allows for the collection and analysis of telemetry data. The technical specifications and all the detail about these components are in next section.

5.3.1 Quadrotor Components and Configuration

5.3.1.1 Flight-Controller (Microcontroller)

The flight controller consists of the main microcontroller and off-board sensors. It is mainly responsible for stabilizing the quadrotor and executing the tasks specified by the user. In our quadrotor, we used an STM32F103 in Figure 5.13, manufactured by STMicroelectronics. It is a 32-bit ARM architecture microcontroller using the ARM Cortex M3 microprocessor core, with a maximum CPU speed of 72 MHz, operating at voltages of 2.7V to 5.6V, with 37 GPIO pins, 10 analogue input pins of 12-bit resolution,

15 digital pins with 12 PWM pins, 2 for I2C and the other 2 for SPI, it has a big flash memory of 64KB. and 20 kB of RAM. Due to roger-Clark et al effort STM32 become compatible with the Arduino development platform (IDE), it is suitable for quadrotor.

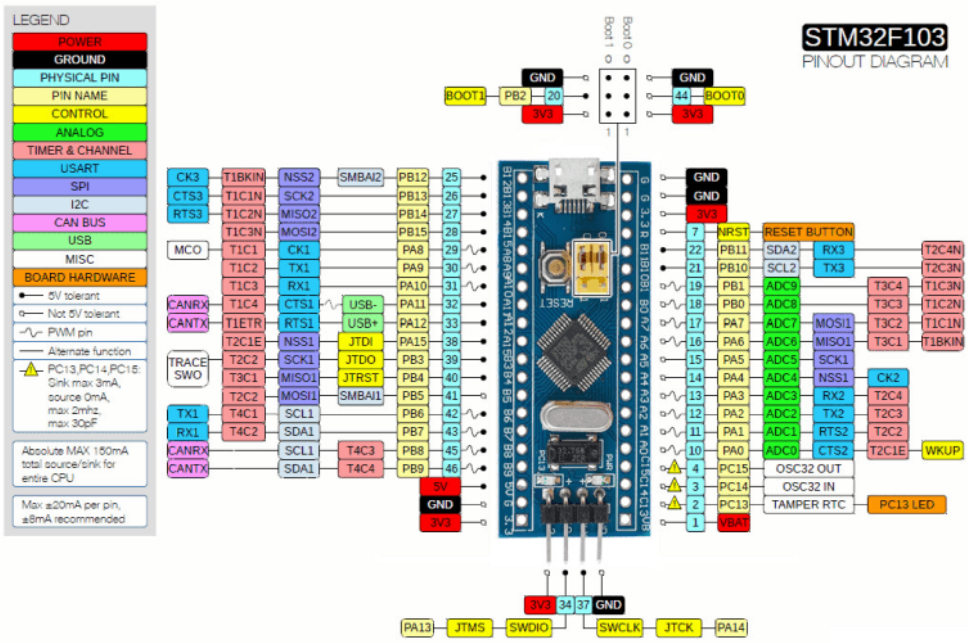


Figure 5.13: Our quadrotor flight-controller microcontroller STM32F103

5.3.1.2Flight-controller (Sensors)

A) IMU

The Inertial Measurement Unit MPU6050 Figure 5.14 is the main sensors used in UAV application, is comprised of MEMS three-accelerometers, three-gyroscopes as six-DOF motion sensing devices Figure 5.14 intended for low cost, low power consumption, and high performance, as well as a digital motion processor (DMP), which has the ability to do complicated computations. The device comprises a 16-bit analogue-to-digital converter. Due to this characteristic, it simultaneously captures motion in three dimensions. use a conventional data communication I2C bus.

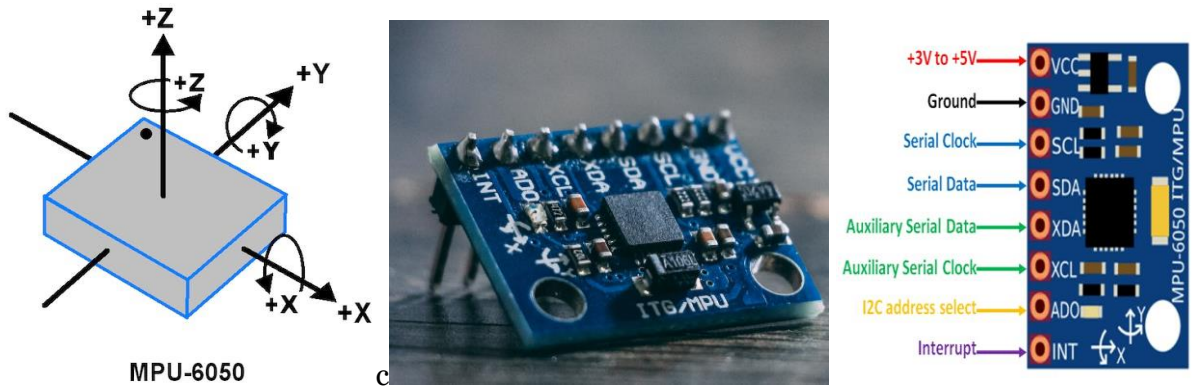


Figure 5.14: Inertial Measurement Unit MPU6050

The accelerometer in Figure 5.15 is a free-moving mass is suspended by springs and moves as the apparatus accelerates. The fixed electrode comb creates a capacitive interaction between itself and the suspended mass. Changes in capacitance are captured while the sensor moves and processed by an ADC to a numerical value ranging from 0 to 32,750. The gyroscope works in the same way as the accelerometer, but it relies on the Coriolis Effect rather than acceleration.

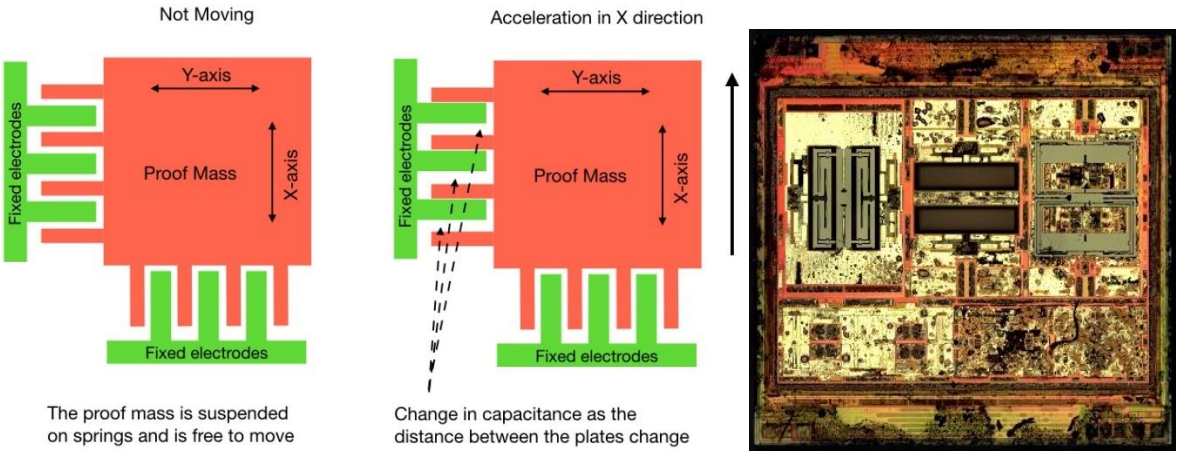


Figure 5.15: Streamlined internal configuration of the accelerometer

As previously stated, the capacitive sensors' analogue voltage readings are transformed into a digital signal ranging from 0 to 32750. These numbers represent the units of measurement for the gyroscope and the accelerometer. In order to convey useful information, the measuring units must be divided. As shown in Figure 5.16, the MPU6050 allocates measurement units by establishing four sensitive levels. The sensitivity level chosen depends on the IMU application. For instance, if the application requires high-speed rotations of more than 1500° per second, the gyroscope sensitivity must be configured to 2000°. since the gyro must cover a large

amount of rotation in a short amount of time. Because the gyroscope is used at medium speed in the majority of applications, the sensitivity level is set to 250° by default. This yields 131 measurement units per second per degree, representing an extremely precise level of accuracy. The standard accelerometer sensitivity is 2 g. This should be suitable for the majority of uses.

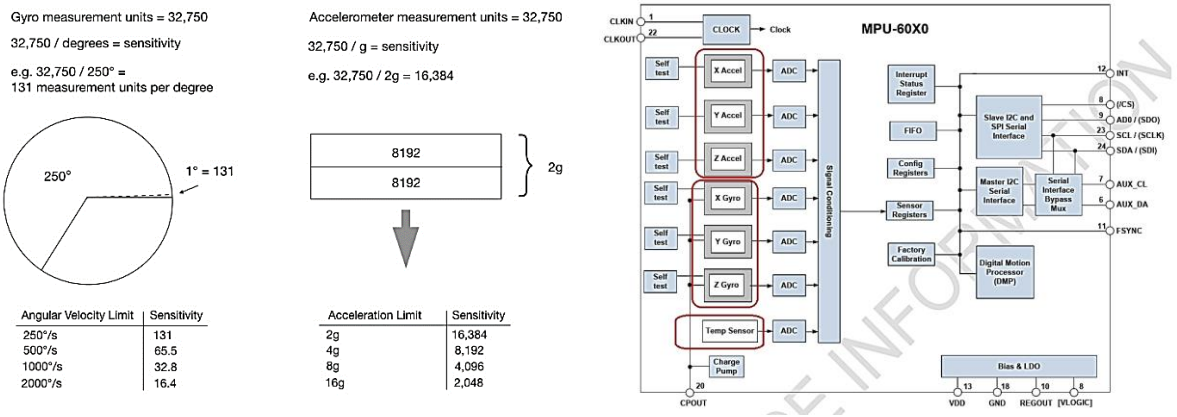


Figure 5.16: MPU6050 Sensitivity Levels for Measurement Units Allocation

Gyroscope Features: The MPU-6050 gyroscope includes a wide range of features, including:

- X, Y and Z axis digital output angular rate sensors (gyroscopes) with a user-programmable full-scale range of ±250, ±500, ±1000 and ±2000°/s
- By connecting an external sync signal to the FSYNC pin, we can sync images, videos, and GPS.
- Built-in 16-bit ADCs allow simultaneous sampling of gyroscopes.
- Temperature stability compensation is integrated in the sensor.
- Improved low-frequency noise performance and has a digitally programmable low pass filter
- Gyro operating current: 5.6 mA, and low standby power consumption: 5μA Factory calibrated sensitivity scale factor and has user self-test features.

Accelerometer Features: The MPU-6050 accelerometer includes a wide range of features, including:

- Digitally output three-axis accelerometer with a programmable full-scale range of ±2g, ±4g, ±8g and ±16g

- Built-in 16-bit ADCs allow simultaneous sampling of accelerometers without a multiplexer
- Accelerometer Normal Operating Current: 500μA and the current in the accelerometer mode is 10 A at 1.25 Hz, 20 A at 5 Hz, 60 A at 20 Hz, and 110 A at 40 Hz.
- User programmable interrupts, and support High-G Interrupt and user self-test.

Two methods exist for extracting meaningful data from MPU6050. Reading the raw sensor values and using them to calculate the orientation is the first method. The second method is to use firmware to extract data from an MPU's internal DMP, allowing users to add Motion Interface functionality directly into products with minimal development costs and effort. The DMP is also capable of doing the following computations inside the IMU chip: quaternion, Euler, flip/pickup detection, step detection and counting, and fundamental motion detection: driving, walking, running, biking, and tilting.

Because our project is for educational purposes, we used the direct reading and manipulation of IMU registers instead of the pre-prepared algorithm of DMP.

A-1) Reading Angles and Acceleration

Initially, we need to incorporate the Wire.h library, which facilitates I2C communication, and define relevant variables for data storage. Subsequently, we initialize the wire library and reset the sensor by assigning a value of 0 to register 6B which is power management register. Additionally, to configure the accelerometer's sensitivity and full-scale range to +/- 8g, we interact with the "*ACCEL_CONFIG*" register <hex 1C> and set the register bits as <*00010000*>. Similarly, to configure the gyro's sensitivity and full-scale range to 500 degrees/s, we communicate with the "*GYRO_CONFIG" register <hex 1B>* and set the register bits as <*00010000*>. Moving on to the loop section, we commence by reading the accelerometer data from register *"ACCEL_XOUT_H" (or "0x3B").* The data of each axis later is stored across two bytes. To retrieve all the data, we employ the "*requestFrom()"* function to start request the reading of 6 registers encompassing the X, Y, and Z axes "*ACC_X", "ACC_Y*"and "*ACC_Z*" by starting from reading the first register. Subsequently, we read and appropriately combine the data of registers sine their output is two-complement forma. To obtain output values ranging from -1g to +1g, which are suitable for angle calculations, the

outputs has been divide by the pre-chosen sensitivity. In addition, by utilizing the below equation, we compute the roll and pitch angles based on the accelerometer data.

```
ACC_Roll = ( atan (  ACC_Y / sqrt (pow(ACC_X, 2) + pow(ACC_Z, 2) ) ) * 180 / PI) - 0.58;
ACC_Pitch = ( atan ( -ACC_X / sqrt (pow(ACC_Y, 2) + pow(ACC_Z, 2) ) ) * 180 / PI) + 1.58;
```

Subsequently, employing the same steps, we acquire the gyroscope data. By reading, the six-registers of gyroscope starting by address <*0x43*> the first register address, where the value of axis is saved across two registers, and we suitably combine their data. To obtain the output in degrees/second, we divide the combined data by the pre-chosen sensitivity. Later, we integrate the gyroscope and accelerometer data and passes them through complementary-filter to get an accurate reading of angles. In this process, we assign a weightage of 98% to the gyroscope data since it is highly accurate and not affected by external forces. However, one drawback of the gyroscope is its tendency to drift over time. To mitigate this, we incorporate the accelerometer data, which accounts for 2% of the fusion. This small percentage is sufficient to counteract the gyroscope drift error.

B) Barometer

The MS5611 altimeter Figure 5.17 Micro-electromechanical systems with a high-resolution pressure sensor with a small design. This device measures pressure differences with an absolute precision of 1.5 mbar within the range of 450–1100 mbar. Overall, the sensor is capable of measuring pressures between 10 and 1200 mbar. In addition, the MS5611 sensor has an on-chip digital thermometer that measures temperature between -40 C and +85 C with an accuracy of 0.8 C. is designed for altitude precision of 10 cm. Given its low energy consumption, small size, cheap cost, and accuracy, the MS5611 barometer is an excellent option for altimeter readings in quadrotor applications. For data transmission to the STM32 microcontroller, SPI and I2C communication protocols are used. Where the protocol is selected using the “PS” pin and a pull-up resistor, with the high state activating the I2C protocol. The change in electrical resistance of the piezoresistive silicon diaphragm MS7101 Figure 5.17 is caused by a slight change in the membrane surface, and the linear relationship between

the applied pressure and the resistance changes the output and affects the pressure reading and then the altitude.

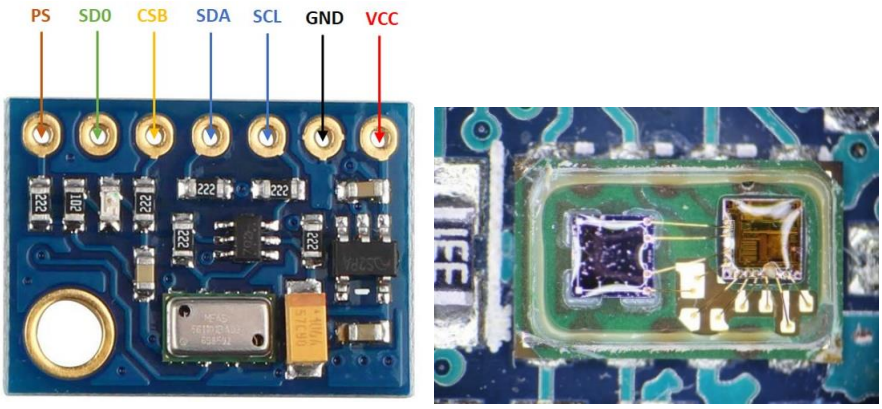


Figure 5.17: Altitude Measurement Sensor (MS5611)

C) Magnetometer

3-Axis Digital Compass IC HMC5883L Figure 5.18 is a multi-chip surface-mount device designed as a low-cost magnetic compass for low-field magnetic detecting applications. a 12-bit analogue-to digital converter that offers heading accuracy of 1° to 2°. the sensor support for the I2C communication protocol. Honeywell's HMC5883L is an asymmetric anisotropic magneto-resistance (AMR) Figure 5.18 with accurate linearity as well as in-axe sensitivity and very low cross-axis sensitivity. This device is intended to detect both the direction and intensity of the Earth's electromagnetic field, which ranges from 1000 to 8 gauss. The sensitivity of a 12-bit analogue-to-digital converter combined with low noise in 2 milli-gauss sensors. Magneto-resistive sensors are constructed of nickel-iron. Where, in the presence of a magnetic field, a change in the resistive components of the bridge induces a variance in volt across the output of the bridges. The HMC5883L is a small, low-cost sensor with high orientation accuracy that is ideal for quadrotor applications.

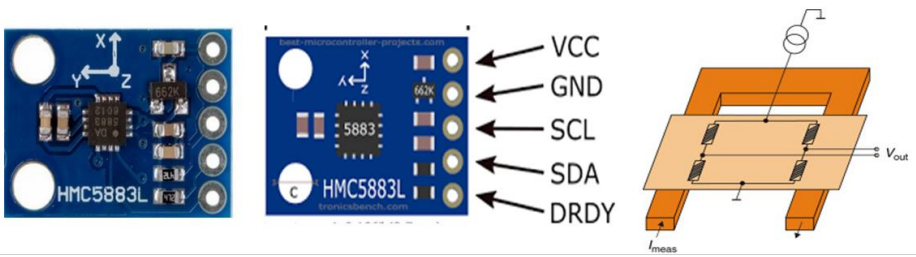


Figure 5.18: Magnetic Field Detection for Heading Measurements (HMC5883L)

D) GPS

Versatile u-box M8 GNSS GPS modules in Figure 5.19, utilize simultaneous receipt of up to 3 systems GPS and Galileo in addition to GLONASS or BeiDou, is tailored for cost-sensitive scenarios, and has an inbuilt flash that enables firmware upgrade in the future. The I2C interface enables communication with the CPU. Its operation is limited to 50 kilometres of altitude and 500 meters per second, for the velocity, with velocity accuracy of 0.05 meters per second and a heading precision of 0.3 degrees. Its sensitivities for track and navigating are -130 dBm for all satellites except the Galileo spacecraft, which is -127 dBm. It employs an SPI interface with a maximal transmission rate of 125 kB/s to communicate with the host CPU. A 32 kHz oscillator driven by an RTC crystal.

Figure 5.19: GPS for Positioning Measurement (U-Box GY-NEO-8M)

The GY-NEO-8M GPS Module has 4 pins:

- **VIN:** Module power supply – 5.3V
- **GND:** Ground
- **RX:** Receive data via serial protocol
- **TX:** Sending data via serial protocol

E) Battery Voltage Level Sensor

During quadrotor flight, it is necessary to know the battery voltage for more accurate flight time calculations and to compensate for the voltage drop's effect on control efficiency. For these kinds of calculations, a voltage sensor was used to measure

the actual battery voltage in each loop. Because we couldn't directly connect the 12.6V battery voltage (when it is full) to the 3.3V of STM32 analog pin. We used a simple voltage divider circuit Figure 5.20 in our quadrotor. Where the voltage divider provides two functions, First, divide the whole battery voltage (12.6V) by the maximum voltage permitted by STM32 analogue pin (3.3v), which is 3.3v. Second, respect the amperage limitation of the analogue pin. Taking into account the 12-bit resolution (4096 discrete steps or different voltage levels, it can sense about 0.8mv) of the analogue pin and the voltage divider's ratio, we can estimate the current battery voltage. The RPM of the quadrotor motor will be directly impacted by a decrease in battery voltage. By trial-and-error method we can determine the compensation factor that relates the battery level to the motor speed and ensures that the motors increase in speed when the battery is low. This compensates for the lost speed from the voltage drop.

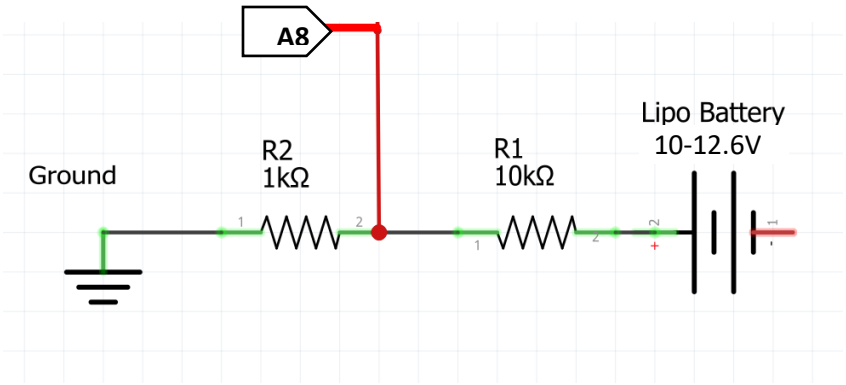


Figure 5.20: Battery Voltage Level Sensor (Voltage Divider) to Analog input

5.3.1.3 Quadrotor Actuators

A) ESC

Figure 5.21 depicts the utilized ESC. Four ESC SimonK Series support 40 A of continuous current, have dimensions of 68 mm by 25 mm by 8 mm, and weigh 41 grams. The BEC output supports until 3A/5V. the ESC is programmable using BLHeli firmware. The default control frequency is 8KHz, they are optimized for high performance, exhibiting superior linearity and a significantly quicker throttle response. Utilize authentic electronic components to guarantee high quality and improve the ESC's current endurance capacity. Protections include low-voltage cutoff (The default threshold is 5.0V/cell), overheat (When the ESC MOSFETS temperature exceeds 100 C°), and throttle signal loss (Power will slowly reduce to zero when signal

is lost). The throttle range is fully compatible with all receivers and can be configured to provide a linear, precise throttle response. All parameters, including the factory defaults, are programmable via a transmitter.

Figure 5.21: ESC for motor control (Simkon 40A 2-4S Lipo)

B) Brushless motors

Four EMAX XA2212 12V DC 980KV Brushless Motor in Figure 5.22 has been used to hold the quadrotor, it is low-cost and high power and efficiency motor, turn 980 time each minute at each voltage, support from 2-3S voltage (7.4-12.6 V) and 9.5 to 15.2 A of current, its Trust from 550 to 880 grams depend to the propeller size as Table 5.7. Its Weight is 68g. its performances suitable for low-speed training multi-rotor airplanes.

Figure 5.22: Brushless DC Motor (EMAX XA2212 980KV)

Table 5.7: Characteristics of the XA2212 (980KV) Brushless Motor with Different Propellers and Voltages

Motor Type	The Voltage (V)	Prop. Size	Current (A)	Thrust (G)	Power (W)	Efficiency (G/W)	RPM
XA2212 980KV	12	APC 10x4.7	15.2	880	181.2	4.9	6960
	8	APC 10x4.7	9.5	550	76	7.2	5470

	12	APC 9x6	12.3	730	147.6	4.9	8220
	8	APC 9x6	7.1	400	56.8	7.0	6090

C) Propellers

APC 1047 propeller with a central hub thickness of 8 mm, and 10g of weight. It made from plastic, and it is an excellent propeller for small multirotors, it has two sets of direction clockwise (CW) and counter clockwise (CCW), and is of high quality compared to its price, the center hole of the propeller is designed APC 10x4.7 is a propeller with a 10-inch diameter and a 4.7-inch pitch per revolution. From its datasheet in [207], Figure 5.23 illustrate the relationship between the drag and thrust coefficient respectively with the advance ratio of propeller at different RPM. Where all the propeller characteristics and experiments setup will find in [207-208].

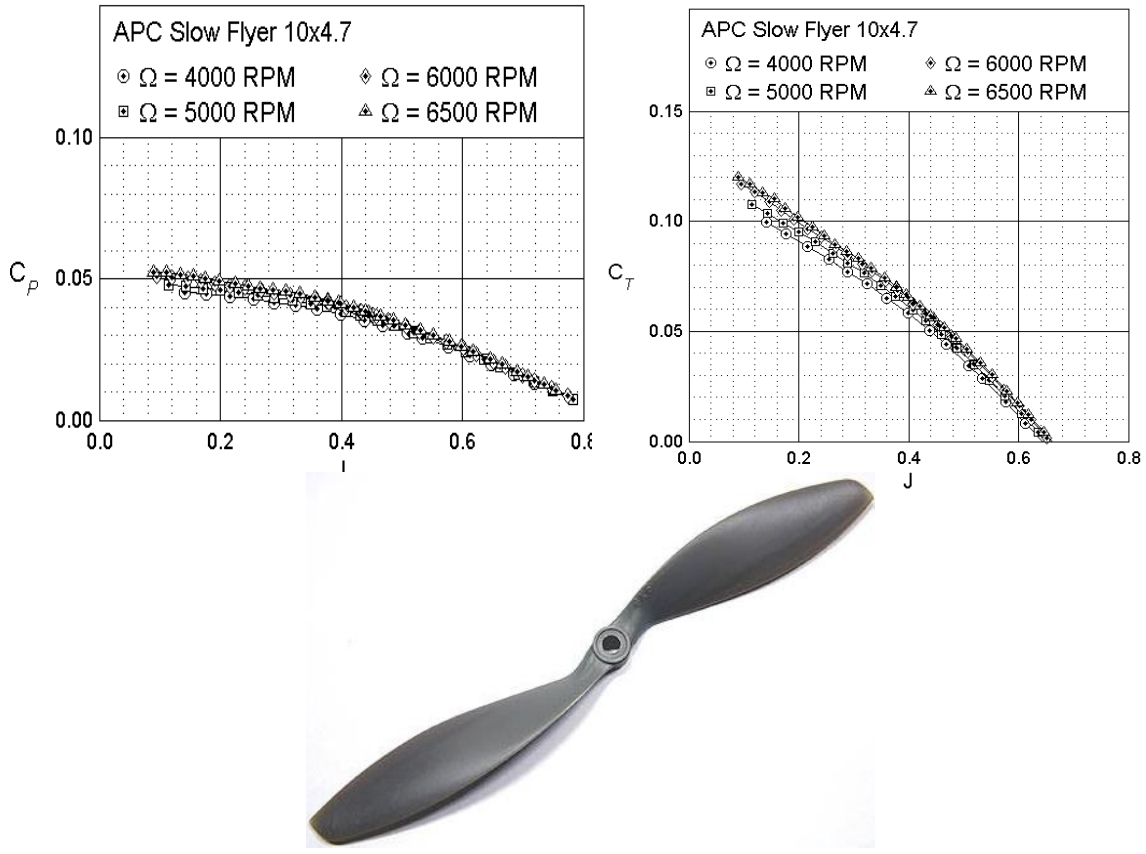


Figure 5.23 Propeller (APC 1047) and its main correspond characteristics [207]

5.3.1.4 Power Supply

In fact, many different types of Lipo batteries were used in our project; however, in this section, we will only focus on the battery that was used in the major experiments shown in Figure 5.24, a Lipo battery with a capacity of 3700 mAh and three cells (3S). Their total nominal voltage is 11.1V. Its weight is 327 grams. Sporting a continues discharge rate of 80CC and burst rates of 140C. The maximum continuous current draw is Battery capacity (Ah) multiplied by the discharge rate (C) (3700mAh x 80 = 296 Amperes), which can give each motor 74 amperes continuously, it gives more than what we need.

Figure 5.24: Battery Used in Major Experiments (Lipo 3700 mAh 11.1V 3S 80CC)

5.3.1.5 Quadrotor Communication devices

A) RC Transmitter / Receiver

Radio-link The AT9S in Figure 5.25 is a professional transmitter with 10 / 12 channels that operate at 2.4 GHz. Remote Control Radio Receiver and Transmitter (R9DS) Outstanding Anti-Interference DSSS and FHSS communication technologies provide the AT9S Pro transmitter with a steady broadcast signal and a long-range for aircraft flying up to 5.4 kilometres away. 3ms quick response and Real-Time Telemetry (operate with PRM-03), Throttle Cut, Dual-Rate, Programmable Mix Control, Fail Safe, Reverse Polarity Protection, Multiple Model Options Figure 5.27, and other features make it an excellent choice for both expert and novice users. It supports trainer mode and the majority of aviation simulator software, including Realflight, PhoenixRC, and FreeRider. Supports wired and wireless RC simulator trainer interfaces. It has Radio/receiver/model low voltage alerts. Furthermore, it is compatible with R9DS, R6DS, and R6DSM receivers. Data storage for 15 models and rapid model switching for helicopters, quadrotors, airplanes, gliders, vehicles, and

watercraft. It is 880 grams in weight. Support firmware updates to get new features. AT9S Pro also supports data cloning. The microcontroller unit's model number is STM32F103VET6 (32-bit ARM 72MHZ basic frequency, 512K flash capacity, and 12-bit ADC). Support many protocols like SBUS/PPM/PWM/CRSF. It features multimode presets for all kinds of UAVs.

The RadioLink R9DS receiver in Figure 5.26, which work at 2.4 GHz and support 10 channels via Sbus and 9 via PWM, can operate in the expanded DSSS and FHSS spectrums at the same time. It features an antenna with an iPEX connection that is removable. Operational voltage: 4, 8–6 V; current: 38–45 mA; (input voltage: 5 V). Unfortunately, PPM modulation is not supported by this receiver. This necessitates the use of a PPM encoder in Figure 5.25, which converts PWM to PPM modulation to reduce wiring. As described previously, PPM encoders convert multiple PWM channel signals into a PPM composite signal output.

Figure 5.25: RC Transmitter (Radio-link-AT9S Pro)

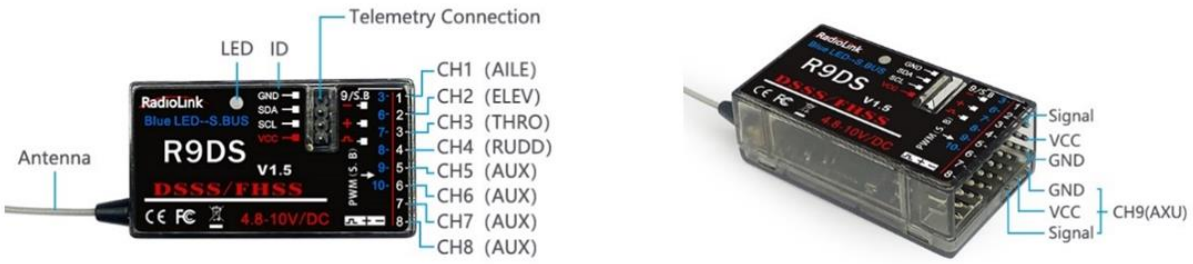


Figure 5.26: RC Receiver (RadioLink-R9DS)

Figure 5.27: Some Displays for Model Options & features of RC Transmitter

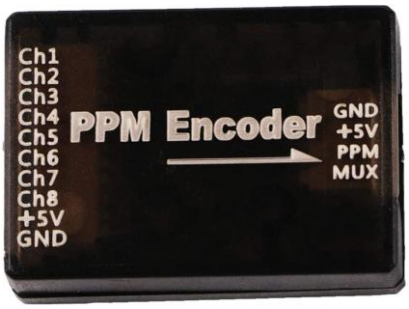


Figure 5.28: PPM encoder (convert 8 PWM channels to one PPM channel)

B) Telemetry System

The APC220 radio module in Figure 5.29 offers a straightforward and cost-effective solution for wireless data transmissions. Integrates a high-speed integrated microcontroller and IC to produce a clear UART/TTL interface, eliminating the need for packetizing and data encoding. is an inexpensive solution with superior range performance. Work at voltage 5V and maximum transmission distance of 1 KM at 9600 bps (line of sight). It has a data buffer of 256-bytes, can provide more than 100 channels, which makes it very useful for data telemetry. It has two parts, the first connected to the flight controller and the second connected to a PC via a RS232-TTL converter or to another microcontroller board like an Arduino via TX and RX pins. In our project, we used the serial communication protocol to send flight data to the

ground station (PC) at each loop, which then decoded the data to store and plot it. Figure 5.29 presents the wiring of modules to the PC and to the flight controller. The flight states that have been sent via telemetry system are, battery voltage, all quadrotor altitude and attitude states etc.

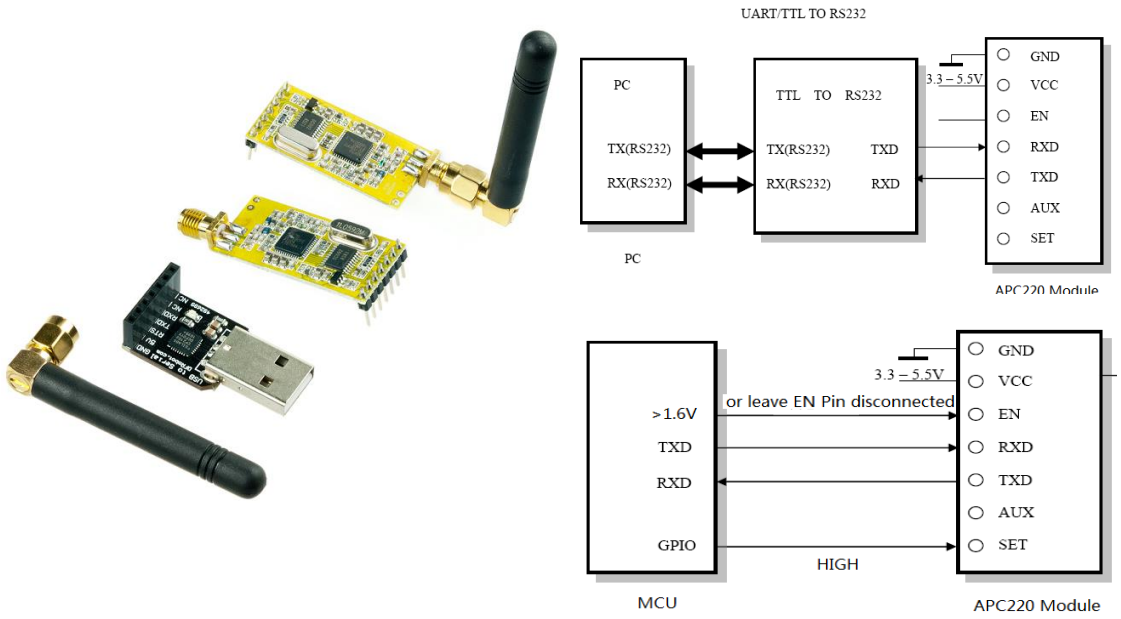


Figure 5.29: Telemetry APC220 radio module and Transmitter/Receiver wiring

5.3.1.6 Quadrotor global wiring and Algorithm

The overall wiring configuration depicted in Figure 5.30 illustrates the setup of our quadrotor, highlighting key components and connections. At the core, the quadrotor features an STM32 flight controller along with four propulsion chains. The ESC BLDM is powered by an 11.1-volt Lipo battery. To ensure stability for other devices, a 5-volt regulator is incorporated to provide a consistent 5-volt level. Alternatively, the ESC's BEC output, offering 3A/5V, can serve as an alternative 5-volt power source. Monitoring the Lipo battery level is crucial to compensate for the power drop effect in the control signal. This is achieved by utilizing a voltage divider and reading the level through the analogue A8 pin. Moving on to the sensory aspects, the quadrotor is equipped with various sensors, among which the MPU6050 holds significant importance. It measures acceleration and angular velocity, communicating these values to the STM32 flight controller via the I2C two-wire communication protocol. The communication takes place through the signal clock (SCL) and signal of data

(SDA), which are connected to STM32 digital pins B10 and B11. Other sensors, such as the MS5611 barometer and HMC5883L, also utilize the I2C protocol and are powered by 5 volts. Furthermore, the GPS module is linked to the STM32 flight controller using the serial TX and RX protocol, connected to pins A3 and A4, respectively. To streamline the operator control signals traveling from the transmitter to AT9S and subsequently to R9DS, a PWM to PPM encoder is employed. This enables the conversion of the signals to PPM modulation, significantly reducing wiring congestion. A single analogue pin A0, as opposed to nine pins required for PWM modulation then reads the PPM-encoded signals. This reduction in pin usage helps alleviate wiring complexities and frees up STM32 pins for other essential functions.

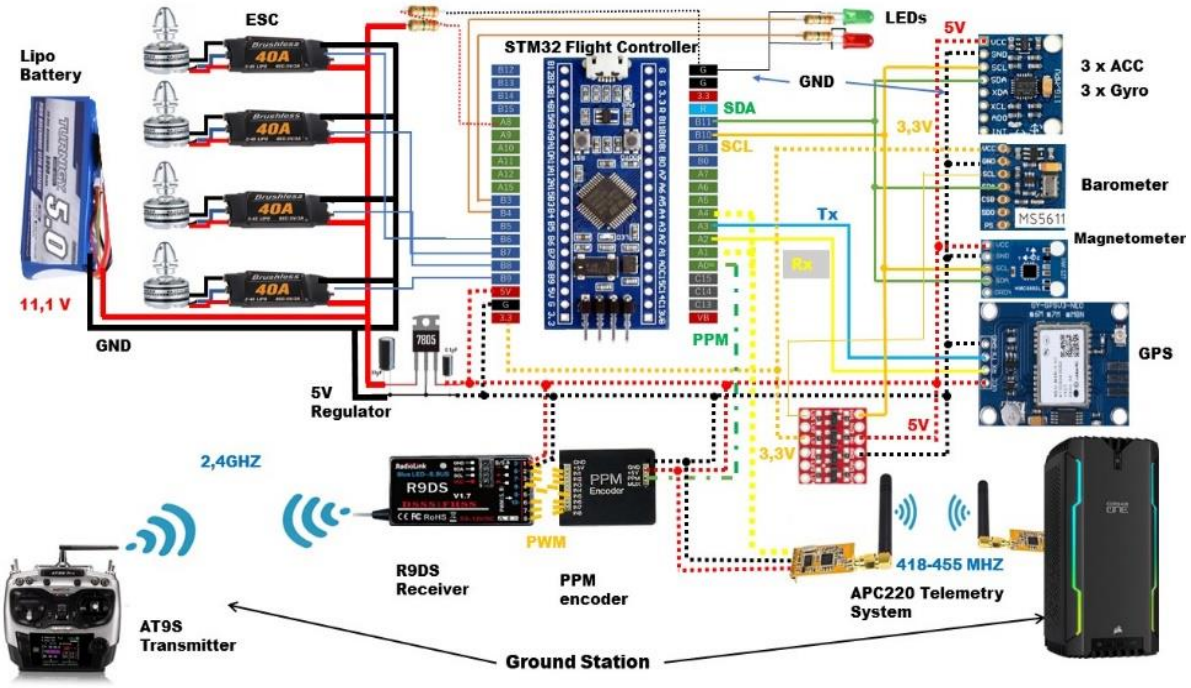


Figure 5.30: Quadrotor Global Wiring Configuration

The flowchart presented in Figure 5.31 outlines the algorithm employed in our quadrotor system. It initiates with the declaration of libraries, variable constants, and port assignments. Subsequently, it reads the preloaded values stored in EEPROM, which serve as calibration constants and define various limitations. Additionally, the algorithm sets up the necessary timer registers, reads the pulses from the RC receiver, and prepares the ESC output pulses. Next, the algorithm establishes communication with the sensors through I2C to verify their responsiveness. Specifically, it ensures that

the MS5611 barometer, HMC5883L compass and MPU-6050 IMU are functioning properly. Once the wiring and communication checks are successful, the algorithm proceeds with the initialization of sensor registers. This involves configuring the necessary settings based on requirements. For instance, for the IMU, the gyroscope configuration register is set to support a full scale of 500 DPs for the gyroscope, and the accelerometer configuration register is set to a full-scale range of +/- 8g for the accelerometer. Additionally, a low-pass digital filter is set to operate at "43 Hz" to filter the sensor data appropriately. The sensor is partially calibrated at this stage to establish an average initial value for quadrotor states. Before initiating flight, the algorithm checks the battery voltage to ensure it is fully charged and ready for operation. Similarly, the reception of RC signals is verified to confirm that the quadrotor is ready to receive control commands.

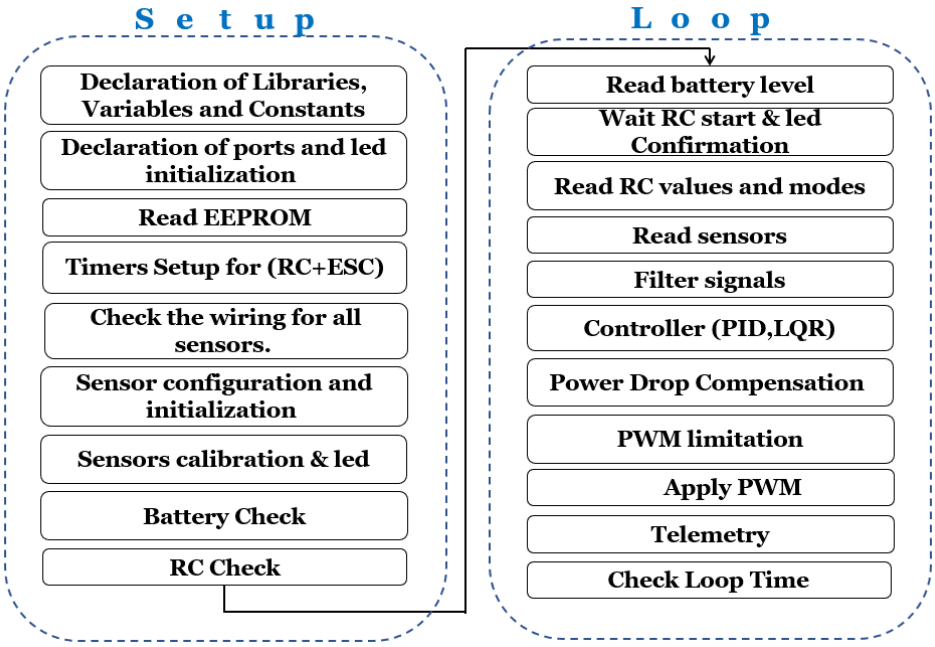


Figure 5.31: Quadrotor Algorithm Flowchart

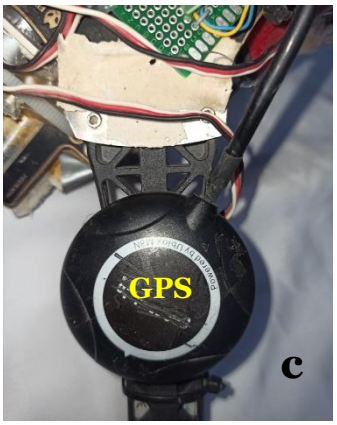
The main loop program begins by reading the battery voltage and continuously monitoring for incoming RC pulses. As soon as the first control command is received via RC, the algorithm starts reading and filtering the sensor signals. These signals are used as inputs for the PID controller, along with the corresponding RC control signals (references), to generate control signals for the actuators. These control signals are then compensated for any power drop. The output signals from the PID controllers

pass through a mixing function, which generates the PWM signals for the four motor-propellers. The flight states are sent to the ground station through a telemetry system at each iteration of the loop. To ensure the algorithm runs at the desired sampling time or "loop time," a loop time checker is implemented. This checker regulates the timing of the loop and ensures that all calculations, such as angle calculations and PID calculations, are executed accordingly.

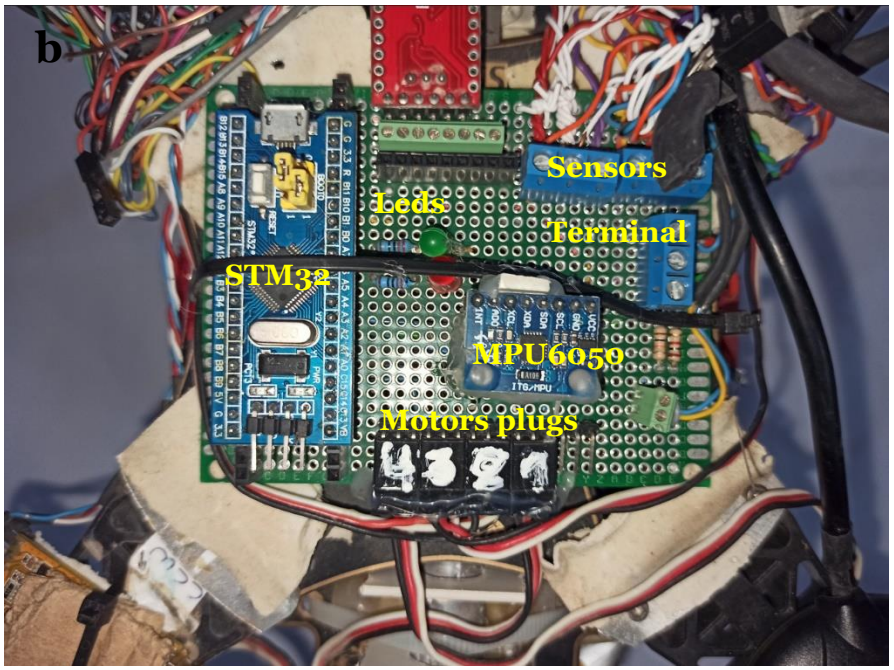
5.3.1.7 Building and Predicting Quadrotor Behaviors

To build a quadrotor Figure 5.32.a, we commence by acquiring the necessary components: a quadrotor frame, four motors, four propellers (two clockwise and two counterclockwise), ESCs Figure 5.32.e, a flight controller Figure 5.32.b, a power distribution board (PDB), a battery Figure 5.32.g, a radio transmitter and receiver Figure 5.32.h and 5.32.f, and any additional components such as a GPS module. Next, we proceed to assemble the frame according to the provided instructions. This entails attaching the arms, mounting plates, and landing gear in the prescribed manner. Subsequently, we securely install the motors to their designated positions on the frame. Each motor is affixed using screws or mounting brackets. Additionally, we connect the propellers, ensuring their correct orientation. The ESCs are then connected to their respective motors, facilitating control over motor speed and direction. We position the flight controller at the center of the frame and secure it using screws or adhesive pads. This vital component serves as the quadrotor's central processing unit, governing its stability, orientation, and flight modes. Following the wiring diagram provided, we establish connections between the ESCs and the flight controller. If desired, we also connect any supplementary components such as a GPS module or camera. We proceed by installing the power distribution board (PDB) and linking it to the flight controller.

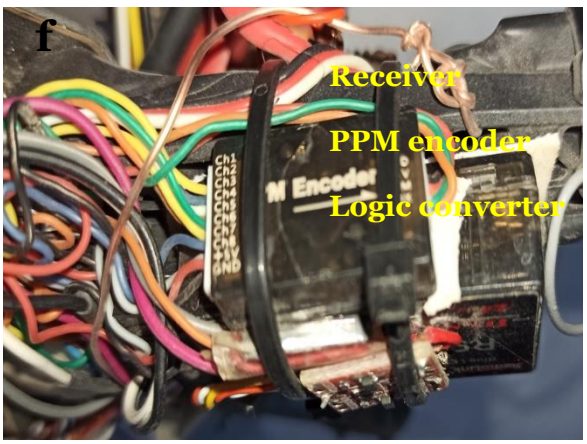

a
Motors
Propeller


GPS
c


d
APC220 Telemetry


b
Sensors
Leds
Terminal
STM32
MPU6050
Motors plugs


e
ESC


f
Receiver
PPM encoder
Logic converter

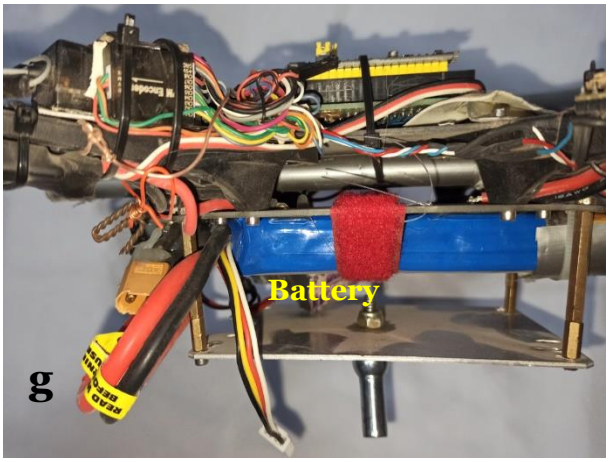

Battery
g

Figure 5.32: Quadrotor Assembly (a) quadrotor, (b) Mounting the Flight Controller, (c) GPS , (d) Telemetry, (e) ESC ,(f) RC received and the PPM encoder, (g) Battery, (h) Transmitter

The PDB functions to distribute power from the battery to the flight controller, ESCs, and other associated components. We ensure that the battery is properly connected to the PDB to supply power to the quadrotor. To set up the radio transmitter and receiver, we perform the binding process as outlined in the instructions. Furthermore, we establish the necessary connections between the receiver and the flight controller, ensuring appropriate channel assignments for throttle, yaw, pitch, and roll control. Before proceeding, we meticulously inspect all wiring connections to ensure their correctness and security. Our thorough examination involves checking for loose wires, potential shorts, or any other potential issues that may compromise the quadrotor's performance. To minimize vibrations during flight, we install and balance the propellers. We utilize a propeller balancer or adhere to the balancing instructions provided by the manufacturer to achieve proper balance for each propeller. This step is crucial and contributes to optimal flight performance. The flight controller requires configuration, which involves connecting it to a computer and utilizing the Arduino IDE. This enables us to establish flight modes, calibrate sensors such as the gyro,

accelerometer, and compass, adjust PID settings, and configure additional features or modes. Prior to the initial flight, we conduct a comprehensive pre-flight check. This entails a meticulous review of all components, connections, and settings. We verify the secure attachment of propellers, ensure proper battery connection, and confirm the accurate response of all controls. Finally, we embark on a test flight in a safe and open area. We gradually increase the throttle and evaluate the responsiveness and stability of the quadrotor. Our objective is to ascertain that the quadrotor hovers and manoeuvres as anticipated. Any necessary adjustments to the flight controller settings are made to optimize performance. The figure 5.32 presents our final quadrotor and its components.

After building the quadrotor based on the optimization tool, it was discovered that the actual weight of the quadrotor was less than the weight studied during the optimization process. This discrepancy necessitates the need to predict the quadrotor's behaviors with the new weight. The quadrotor used in the experimental tests is depicted in Figure 5.32, involved a quadrotor with a radius of 0.235 meters and a mass of 1.25 kg with the guard attached and 1.05 kg without the guard. The propellers had dimensions of 10 inches and a pitch of 4.7 inches. These values were input into the prediction tool [38], which is widely used by researchers in the field of UAV sizing.

The prediction tool provided different predictions for the quadrotor's inertia moments, motor thrust, and drag coefficients, as detailed in Table 5.8. These predictions were initially intended to be compared to the experimental results. However, since the actual weight of the quadrotor turned out to be less than the weight used in the optimization process, it is necessary to reassess and predict the behaviors of the quadrotor considering the revised weight. and was performed using

Table 5.8: Quadrotor prediction parameters using fight evaluation tools FlyEval [38].

Quadrotor Mass (kg)	m	1.05
Acceleration of Gravity (m/s²)	g	9.8
Quadrotor Inertia moments	J_{xx}	1.279e-2
J=diag (J_{xx}, J_{yy}, J_{zz}) (kg.m²)	J_{yy}	1.279e-2
	J_{zz}	2.315e-2
Distance of Motor to Center (m)	d	0.235
Thrust coefficient (N/(rad/s)²)	C_t	1.227e-5
Moment Drag coefficient (Nm./(rad/s) ²)	C_m	1.641e-7

Other prediction tools [27-28] give different predictions for the quadrotor's behaviours as in Figure 5.33.

Initially, the battery's consumed load is approximately 16.3 times the nominal current, resulting in a continuous load of 60.31 A. As long as this value falls within the acceptable range of the continuous discharge rate of 80/120 C-rating, this is an excellent value. Hovering flight time is predicted to be around 17.2minutes. At its optimum performance, the motor turns at 9973 RPM with an efficiency of 80.3%. The quadrotor is classified between FPV aerial explorers and FPV race multi-rotors due to its 46% throttle at 4,498 RPM and the 123 W/kg power-to-weight ratio is good. The thrust-to-weight ratio of 5.0:1, whereas the ideal ratio is greater than 2:1. This enhances the quadrotor's manoeuvrability and allows it to carry a payload of 1,656 grams, which is half of the maximum payload. All these values are inside the interval of +- 15 %. Flyeval Tools accurately predicts quadrotor geometry parameters [27].

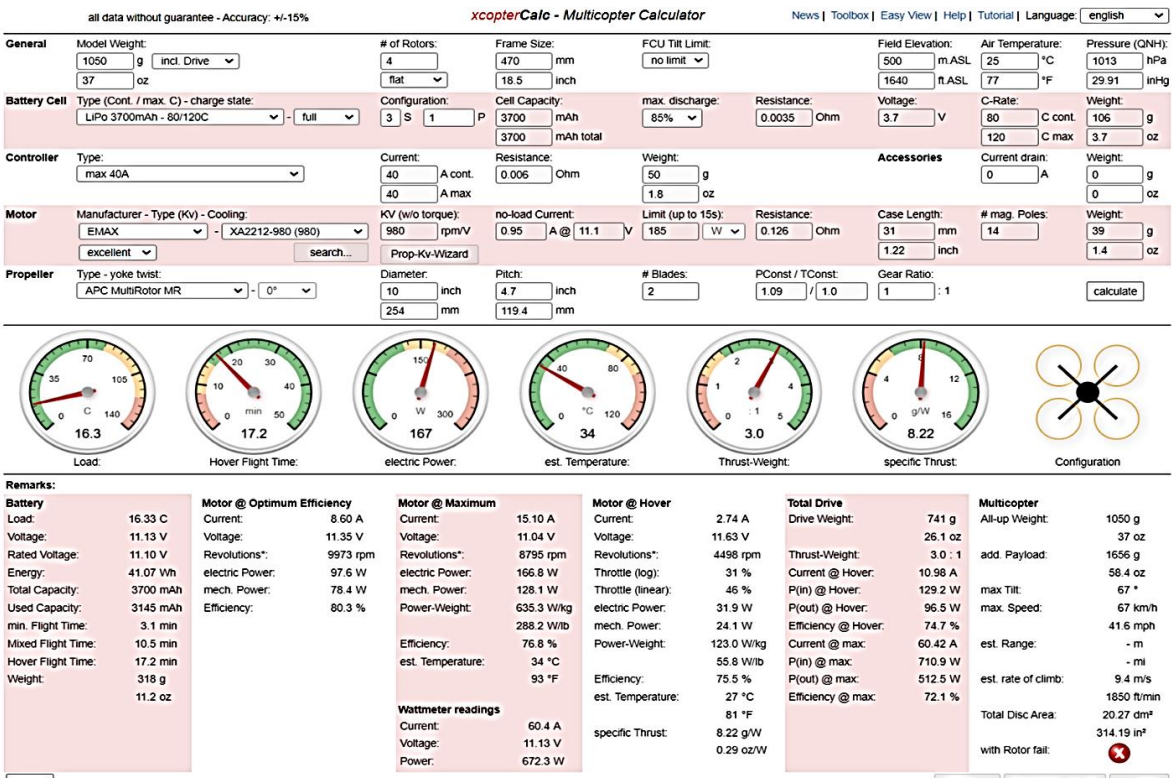


Figure 5.33: Predictions for the quadrotor's behaviours use multirotor calculator [27]

In conclusion, the use of optimization tools such as eCal and FlyEval flight evaluation tools and other online available tools has allowed us to make predictions regarding the behaviour of our quadrotor. These predictions include aspects such as the quadrotor's inertia moments, thrust coefficient, moment drag coefficient, hovering

flight time, thrust-to-weight ratio, manoeuvrability, and maximum payload. These results provide valuable insights into the potential performance of the quadrotor and inform the design and selection of its components to ensure optimal results. The next step in our quadrotor development process is to address the issue of vibration. This can significantly impact the performance of the quadrotor and result in reduced efficiency and stability. To mitigate these effects, it is crucial to balance the propellers and motors prior to the experimental testing phase. This will be addressed in detail in the following section.

5.4 Addressing Vibration Effects on ACC Readings

The influence of vibration on the quality of the ACC reading was investigated, and the range of frequencies that affect the system's stability was identified through the FFT tool. This noise was reduced by using vibration dampening between the brain and the airframe Figure 5.34, as well as balancing the propellers and motors.

Figure 5.34: Vibration dampening for flight controller

The experiment was performed as these steps:

- Balancing the propeller in static condition (using prop-balancer Figure 5.35).
- Balancing the motor in dynamic condition (using FFT tool).
- Balancing the ensemble motor-propeller in dynamic condition (using FFT tool).

5.4.1 Propellers Static Balancing

When a propeller spins at a given speed, it produces a downward force, which can cause vibration displacement in the vertical direction. The elasticity of the materials

used, as well as the unbalanced mass distribution on each blade, produce vibration displacement. Because of the centrifugal force created by propeller spinning, the force exerted by each blade must be identical. The imbalanced propeller can be corrected by removing material from the heavy blades or adding some layers of paint or adhesive to the lighter blades using the first propeller static balancer Figure 5.35 and later dynamic balancing explained in the next section.

Figure 5.35: Propeller static balancer

5.4.2 Motors Dynamic Balancing

By balancing each motor separately without the propeller and by tracking the ACC vibration strength values ACC_{xyz} at various speeds, motor vibration was minimized by putting an adhesive piece in various locations and sizes within the motor's rotating body until the lowest value of accelerometer vibration strength was reached. The imbalance in the motor rotation is therefore eliminated. In other words, mass well-distribution in the motor rotary body was successfully obtained. The experiment was repeated on each of the four motors. Where our indicator for vibrations strength. Averaging these values in time window of 5s and many attempts, Table 5.9. gives the vibration strength each motor before and after balancing and the total motor's vibration strength at same attempt.

Table 5.9: Vibration strength for imbalance/balance motors

Vibration Strength(g)	**$Motor_1$**	**$Motor_2$**	**$Motor_3$**	**$Motor_4$**	**Total**
Before Balancing	1.09	0.52	0.48	0.41	**0.62**
After Balancing	0.56	0.42	0.38	0.39	**0.43**

Table 5.9 shows the effectiveness of motor dynamic balancing reduced the strength of vibration as expected.

5.4.3 Propeller-motor dynamic balancing using FFT

Typically, there is very minor excess mass that is not detected during static balancing, the effect of this extra will manifest at high speeds by generating vibrations during propeller rotation, the solution to this is by spinning the motor-propeller at the appropriate speed and measuring vibration by the ACC strength ACC_{xyz} average , with preventing any movement, only the vibration propagating in the quadrotor structure. The balance is achieved by adding adhesive pieces to the propeller until the ACC strength average is as low as possible Figure 5.36 and Table 5.10 depicts the difference between balanced and unbalanced motor-propellers. The undistributed mass in the ensemble motor-propeller may not be detectable in static balancing, but in dynamic balancing, a minor excess mass impacts directly on the rotation stability. This is evident in Figure 5.36.a and Figure 5.36.b, There is a considerable decrease in vibration magnitude around all feasible motor dynamic frequencies when compared to the unbalanced situation, as shown by the red line in Figure 5.36.b, where the vibration strength is effectively minimized; however, in Figure 5.36.b the blue line indicates that the magnitude of vibration has been decreased by balancing, making the system suitable for data collection from the IMU with the least amount of noise.

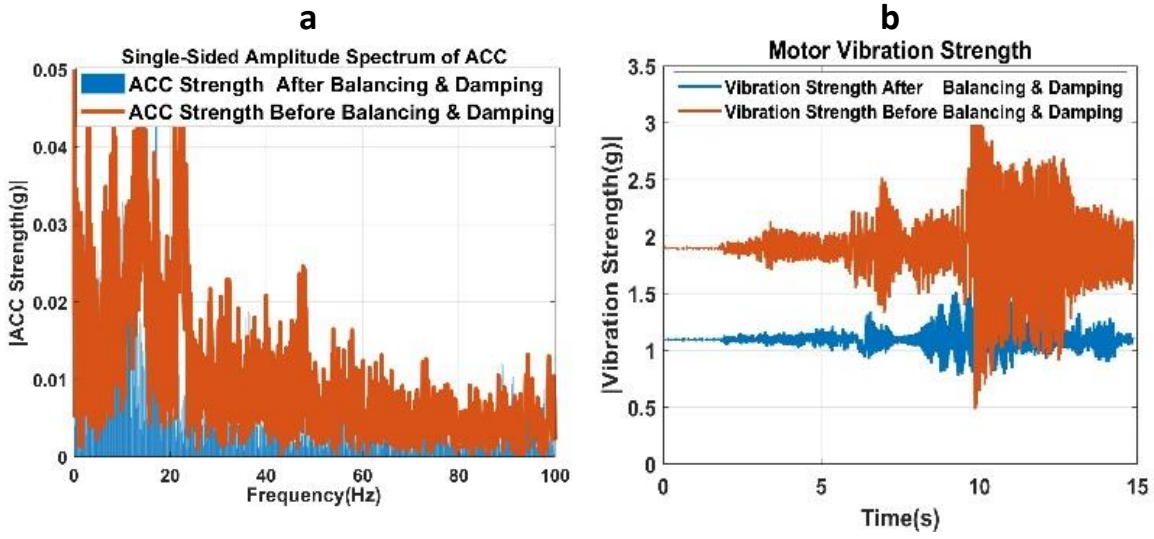


Figure 5.36: Motor 4 unbalanced/balanced vibration using ACC signal strength:

a) frequency Domain using FFT. **b)** Time Domain

Table 5.10: Vibration strength for imbalance/balance motors-propeller

Vibration Strength (g	$Motor_1$	$Motor_2$	$Motor_3$	$Motor_4$	Total
Before Balancing	1.99	1.89	1.78	1.92	**1.89**
After Balancing	1.09	0.89	1.28	1.12	**1.09**

Finally, the results of the vibration reduction efforts demonstrate the significance of propellers and motor balancing in ensuring the smooth operation of the quadrotor. The results from the static balancing of the propellers and dynamic balancing of the motors show a significant reduction in the vibration strength. Furthermore, the use of FFT analysis in propeller-motor dynamic balancing highlights the effectiveness of this approach in reducing the vibration levels. The combination of propeller and motor balancing, along with the implementation of dampening between the flight controller and the airframe, has resulted in a significant decrease in overall noise levels, ensuring a more efficient and reliable flight experience.

5.5 Sensor Calibration

5.5.1 IMU Calibration

The experiment was carried out by collecting the average acceleration for each orientation along all axes Figure 5.37, minimizing the criteria function (RMSE) in equation (3.4), and using a gradient-based method to find the best of the nine SEM parameters that reduced the RMSE. The results are shown below:

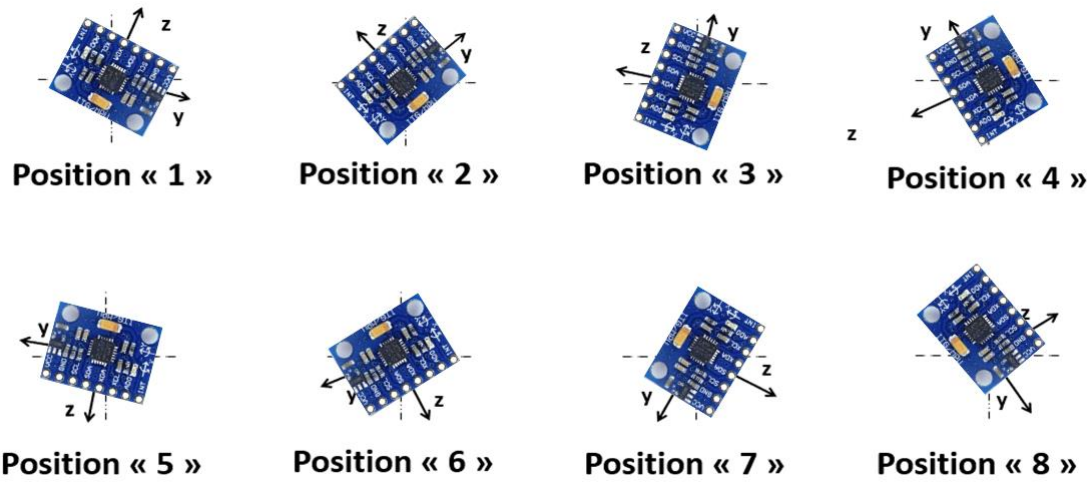


Figure 5.37: ACC calibration along “x” axis

$$T = \begin{pmatrix} 1.0000 & 0 & 0 \\ -0.0156 & 1.0000 & 0 \\ 0.0531 & 0.0034 & 1.0000 \end{pmatrix} S = \begin{pmatrix} 1.0064 & 0 & 0 \\ 0 & 0.9969 & 0 \\ 0 & 0 & 0.9937 \end{pmatrix} b = \begin{pmatrix} 0.0009 \\ -0.0052 \\ 0.0136 \end{pmatrix}$$

RMSE before calibration: 0.015470 g

RMSE after calibration: 0.000426 g

As an example:

The code of reading of IMU data has updated as following:

$$\begin{aligned} \mathrm{ACC_{fx}} &= 1.0064 * \mathrm{acc_{nx}} - 0.0009 \\ \mathrm{ACC_{fy}} &= -0.0157 * \mathrm{acc_{nx}} + 0.9969 * \mathrm{acc_{ny}} - 0.0052 \\ \mathrm{ACC_{fz}} &= 0.0534 * \mathrm{acc_{nx}} \; 0.0034 * \mathrm{acc_{ny}} + 0.9937 * \mathrm{acc_{nz}} - 0.0136 \end{aligned}$$

Before calibration it was:

$$\begin{aligned} \mathrm{ACC_{fx}} &= 1.0 * \mathrm{acc_{nx}} \\ \mathrm{ACC_{fy}} &= 1.0 * \mathrm{acc_{ny}} \\ \mathrm{ACC_{fz}} &= 1.0 * \mathrm{acc_{nz}} \end{aligned}$$

Figure 5.38 demonstrates a significant reduction in the deviation (RMSE) before and after correction using SEM along 24 orientations.

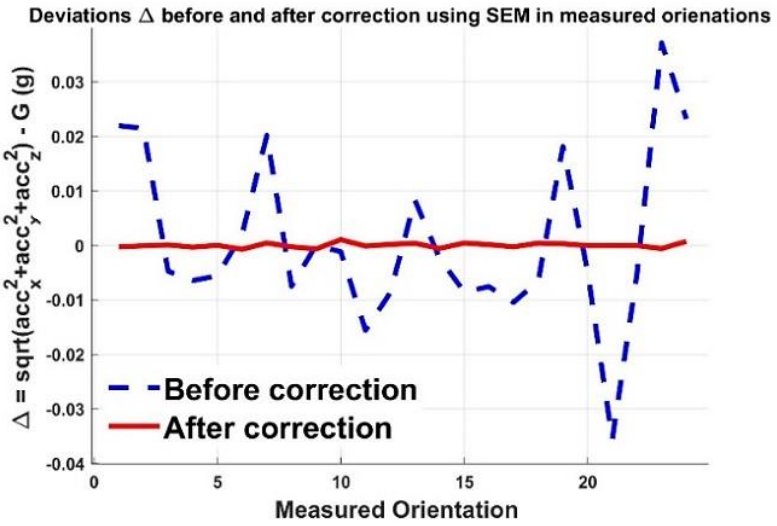


Figure 5.38: RMSE Improvement Before and After SEM Correction Across 24 Orientations

The IMU sensor calibration experiment showed a significant improvement in accuracy. Which indicated a reduction in deviation (RMSE) across 24 orientations before and after the correction with the application of SEM nine parameters. This

highlights the effectiveness of the SEM correction method in calibrating the IMU sensor and enhancing its accuracy.

5.5.2 Compass Calibration

The results of calibration in Figure 5.39 after and before calibration, demonstrate a considerable improvement in orientation accuracy. exhibits the readings of the calibrated HMC5883l magnetometer. As expected after calibration, the combined bias "A" and scale factors "b", soft-iron, and misalignments have been accurately predicted on the sphere manifold whose radius matches the local Earth's magnetic field norm. As expected, the combined bias is a bit high, and this is because the magnetometer used in this experiment is located close to some metal screws and bolts and some rods used to fix the quadrotor body, which affected the magnetometer reading by introducing "larger variations in the magnetic field and due to metal magnetization currents, causing a hard iron effect in the local magnetic field" [72].

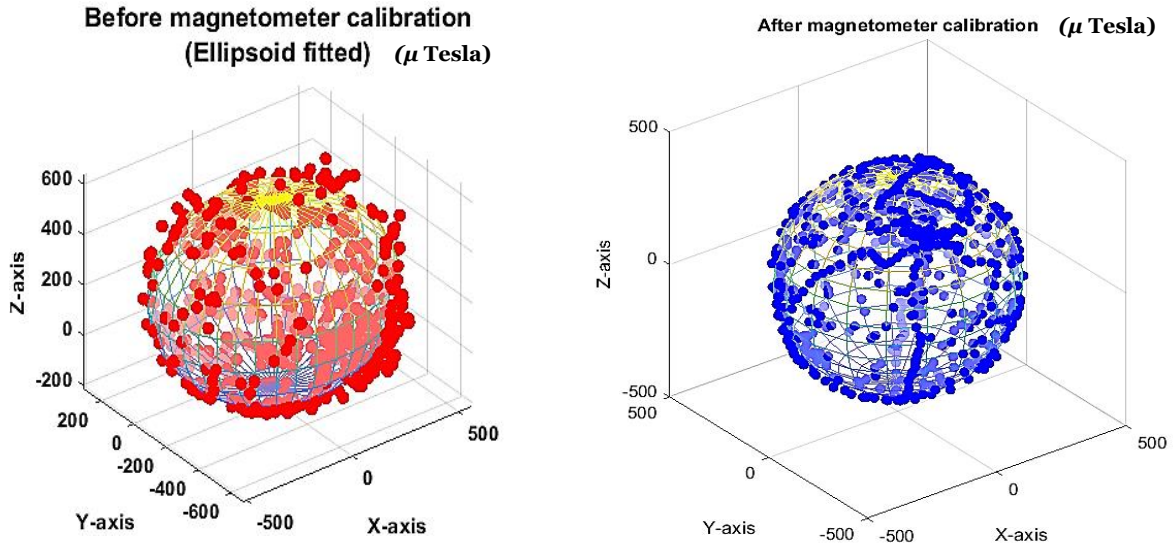


Figure 5.39: HMC5883l Magnetometer-calibrated measurements were plotted on the sphere manifold with normalized radius after calibration.

$$A = \begin{pmatrix} 0.0020 & 0 & 0 \\ 0 & 0.0020 & 0.0003 \\ 0 & 0.0003 & 0.0020 \end{pmatrix} \text{(n.u.)}. \quad \text{Combined bias} \quad b = \begin{pmatrix} -84.4699 \\ -37.8363 \\ -7.3886 \end{pmatrix} (\mu \text{ Tesla})$$

The results of the compass calibration indicate a marked improvement in

orientation accuracy after determining the SEM parameters, including combined bias and scale factors, for the HMC5883l magnetometer.

5.6 Filtring

In our work we focused in complementary filtering, this kind of filter is widely used in quadrotor projects and proves robust and easy to tune and implement in no powerful calculators. The traditional use of this filter is for a single input integration. In order to properly prepare the sensors for use, the IMU drift and biases, misalignments, as well as the magnetometer biases, scale factor, etc. were experimentally extracted and compensated for use. Later, the IMU ACC and GYRO readings are filtered using the LPF and HPF (low pass and high pass). A power spectral density (PSD) Figure 5.40 of the IMU accelerometer and gyroscope data obtained during an experiment demonstrates that there is no resonance peak (This is a quick explanation without discussing bias instability, velocity random walk, or acceleration random walk). Which has been removed owing to the well-balanced rotor propellers and well-selected filter cut-off frequencies of 5Hz and 45Hz, respectively, for accelerometer and gyroscope measurements, this comes from:

$$f_{cut\_off} > 5.f_{max} \text{ and } f_{sampling} > f_{cut\_off} \quad (5.2)$$

Where:

f_{cut_off} : Is the cut-off frequency.

$f_{sampling}$: Sampling frequency.

f_{max} : Desirable maximum band-width.

There was an interest in modelling and controlling the quadrotor at about frequency of 7 Hz (This is only an expectation of quadrotor dynamic bandwidth based on prior – knowledge about the system), which leads to choosing the complementary filter (β = 0,973) based on the equation (2.49), so the frequency of the filters was set at 45 Hz and the sample-rate set at 250 Hz equation (5.2). than the factor 25 or more of the rule of thumb between cut-off frequency and the sampling frequency is satisfied. Which resulted in a substantial reduction in noise as shown in Figure 5.41.

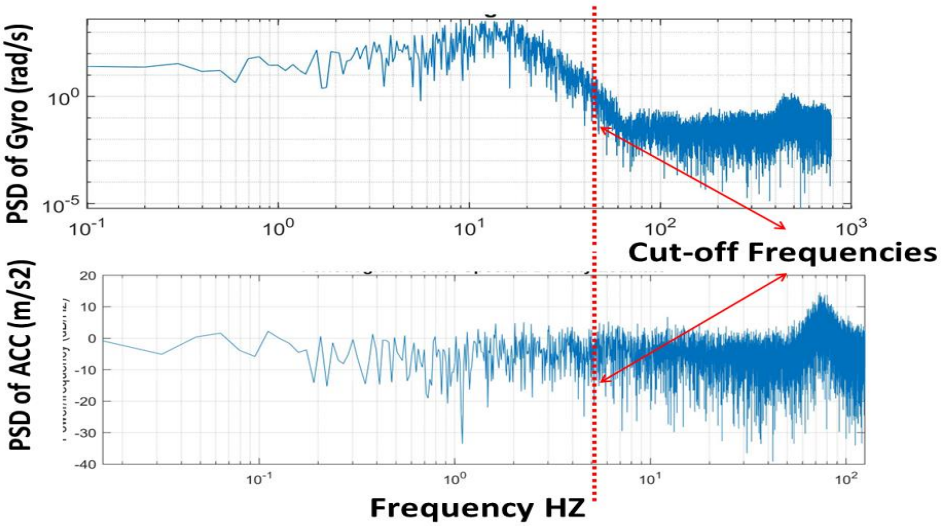


Figure 5.40: Power Spectral Density for ACC and Cut-off frequencies

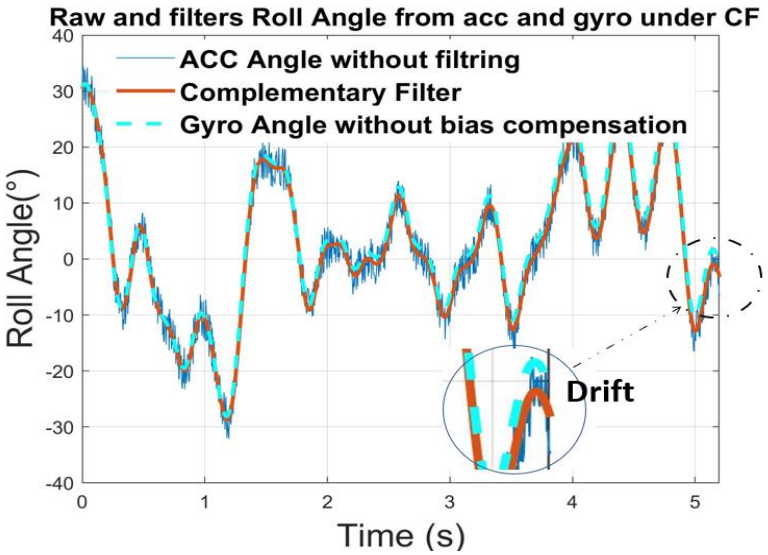


Figure 5.41: Comparison between raw and filters measurements from accelerometer (Acc) and gyroscope under CF

The complementary filter was implemented using the following equation:

$$Angle = \beta * (Angle + GyroRate * Ts) + (1 - \beta) * AccAngle. \quad (5.2)$$

Where angle is the estimated orientation and *GyroRate* is the gyroscope measurement, *Ts* is the sampling period, and *AccAngle* is the orientation estimate from the accelerometer.

5.7 Identification

In this work we identified the quadrotor dynamic using two approaches, white box Grey and black-box. In the white box we estimate the model parameters based on the first principal model, in which the physical parameters of quadrotor dynamic models are extracted from direct measurements. like mass. The extraction of parameters like moments of inertia and aerodynamic coefficients or motor coefficients from experiments, falls under the category of grey box identification. While in black-box identification we used direct approach where the collection of data was from the input and the output of the target dynamic

5.7.1 Grey-Box Identification

In the previous section, we deduced the features and characteristics of the quadrotor that can be obtained by defining the quadrotor components, and we expected the values of the quadrotor geometric characteristics, e.g., moment of inertia, motor coefficients, etc. In this section, we will discuss the experimental setup undertaken to estimate the principal system parameters, including moments of inertia, propulsion coefficients, Examining the dynamic equations of the system reveals that they are composed of sensor-observable variables such as roll, pitch, and yaw. As for the constants in the equation, such as mass, moments of inertia, and motor coefficients, some are deduced directly, such as mass, while others are deduced from experiments, such as moments of inertia. Next, examine how to extract the constants of the quadrotor, which will be difficult to measure directly and can only be determined through experiments. The quadrotor moment of inertia I_x, I_y and I_z and aerodynamic coefficients k_T and K_τ will be extracted from experiments in next section.

5.7.1.1 Motor-propeller thrust and drag coefficients

It is vital to identify the propulsion physical characteristics for an effective quadrotor design. Experiments are required since most components have insufficient or inaccurate data. The thrust coefficient was measured using the testbed depicted in Figure 5.42. A tachometer was used to measure the speed of the propeller at different

PWM output scales ranging from 1 to 2 ms. Utilizing Arduino to generate and transmit various PWM levels between 1 and 2 ms to the motor controller ESC, which is powered by a Li-Po battery and, in response to the PWM signal received from the Arduino board, adjusts the motor speed based on the signal received, which in turn sends the signal that simulates the motor-propeller speed to the Arduino board. This signal is then analysed and stored for later comparison with the control signal. Throughout this whole process, all of the forces caused by the engine turning at different speeds have been measured. In both experiments, these steps were performed identically.

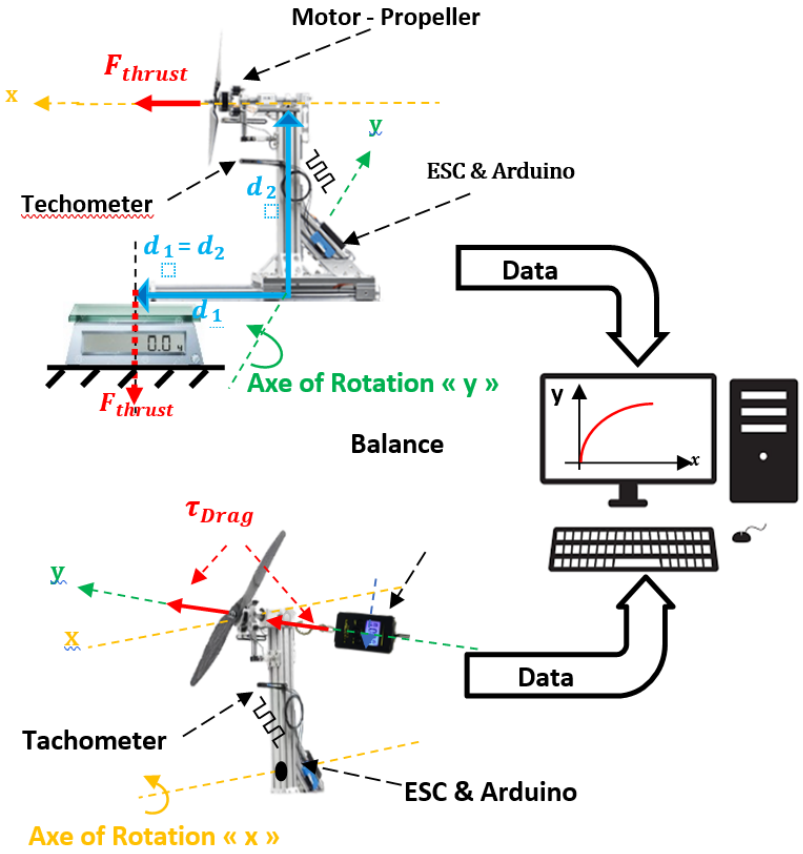


Figure 5.42: Testbed for estimation of motor-propeller thrust and drag coefficients

In the first attempt Figure 5.43.a the k_T thrust coefficient, which relates the motor-propeller angular velocity to the resultant perpendicular force, was extracted. While on the second try Figure 5.43.b the drag coefficient k_τ was found, which shows the relationship between the angular speed of motor-propeller and the lateral force caused by its rotation. Theoretical aerodynamics has already shown that the thrust force changes in a proportional way to the speed square. Where k_T = 1.075e-5 and k_τ=9.7e-6. The result experimental values of k_T, k_τ and the predicted values from the

optimization approach in a Table 5.6 are in the same degree. Because the brushless motors and the propellers of our quadrotor are similar, we assumed that their aerodynamic coefficients are also approximately equivalent.

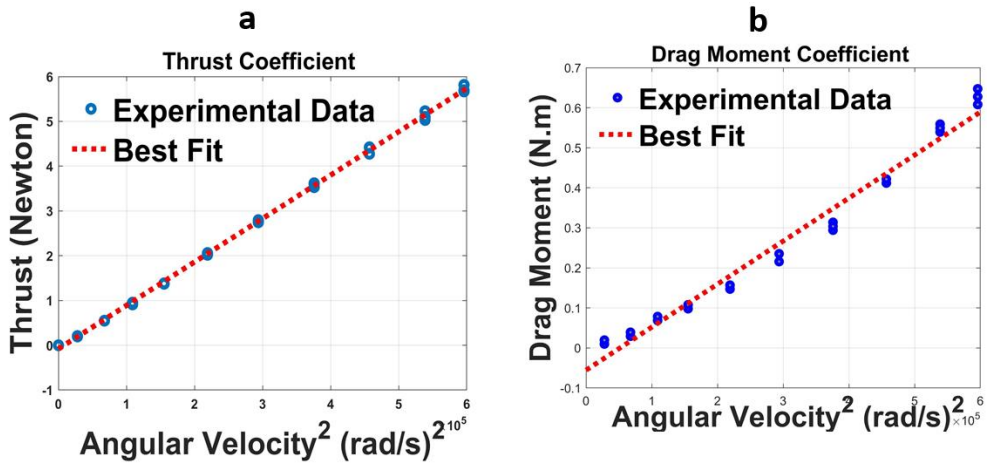


Figure 5.43: The relationship between propeller-speed and the forces (drag and thrust) generated by the motor.

Table 5.11: The drag and thrust coefficient extracted from experimental and prediction tool

Extracted From	k_T **N/(rad/s)** ²	k_τ **N.m/(rad/s)**^{2}
Experimental	1.075e-5	9.7e-6
Optimization	1.227e-5	1.641e-7

5.7.1.2 Quadrotor Moment of Inertia Estimation

One of the numerous mass factors that characterize the stability of an object and the forces required to change its motion is the moment of inertia (MOI). It is defined as the sum of the products of each particle's mass and the distance squared from the axis of rotation. It quantifies the body's resistance to changes in rotational speed. The quadrotor dynamic equations clearly reveal a direct and tight link between the change in angular velocities and the values of MOI, for this purpose, a bifilar pendulum was used to estimate the MOI. The experiment was done as Figure 5.44, where two thin wires of the same length are used to suspend the quadrotor bodying and twist the quadrotor by a small initial angle. The quadrotor will oscillate around a vertical axis, then the frequency of oscillation is measured and recorded.

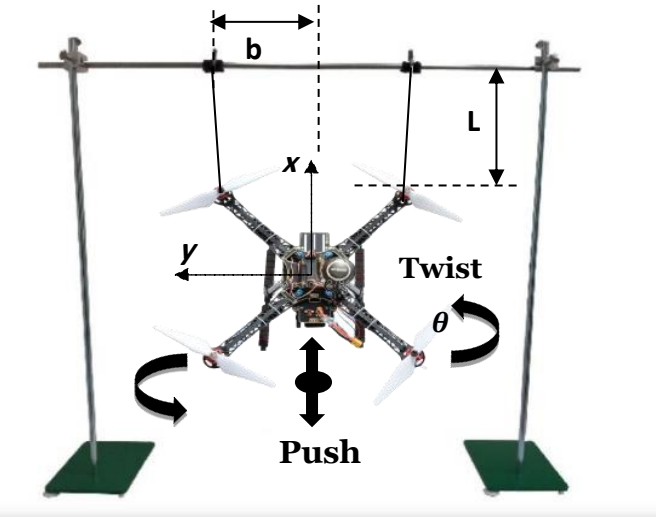


Figure 5.44: Testbed for estimate the quadrotor MOI (Bifilar-Pendulum)

By assuming that the angular movement is modest and ignoring damping, the reduced equation of oscillating motion is given as follows:

$$\ddot{\theta} + \frac{mgb^2}{I_x L}\theta = 0 \tag{5.2}$$

Where:

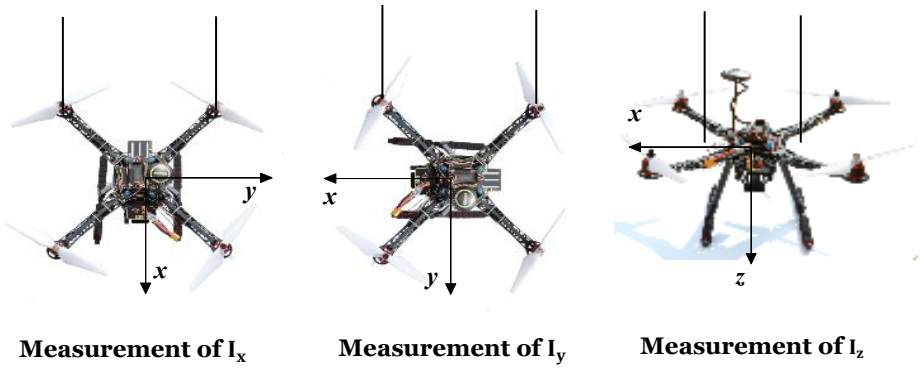
m (kg) is quadrotor mass, g (m/s^2) is the earth gravity. b (m) is distance from the wire to the body's center of gravity. L (m) is the wire length. I_x is the body moment of inertia with respect to axes x. θ (°) is the oscillation angle.

The period of oscillation T is expressed as follows:

$$T = 2\pi\sqrt{I_x L / mb^2} \tag{5.3}$$

$$I_x = \frac{mgb^2T^2}{4\pi^2 L} \tag{5.4}$$

Repeat this procedure for the quadrotor's other axes y, and z to obtain I_y and I_z respectively. This experiment was done at different starting angles and wire lengths as showed in Figure 5.45.



Measurement of I_x **Measurement of I_y** **Measurement of I_z**

Figure 5.45: Quadrotor orientation while estimating MOI with respect of x, y, z, axis

To ensure that the experiment would succeed and provide a result that is nearly accurate, the test bench itself was examined to ensure that it would yield the correct answer and predict the Earth's gravity value correctly. This test-band examination was done as follows:

As before, the quadrotor was given a slight push forward, but this time rather than being twisted to make it act like a simple pendulum and recording the period time T at various runs. The straightforward pendulum moves back and forth. According to this formula:

$$\omega = \sqrt{g/L} \implies g = \omega^2 L = \frac{4\pi^2}{T^2} L \tag{5.5}$$

Where $\omega = 2\pi f$ and $T = 1/f$

In several experiments to extrapolate the quality of the test-band, Table 5.12 shows the experiment test-band predictions for the value of Earth's gravity. These results clearly show that the experimental table is trustworthy as long as it accurately predicts the value of the Earth's gravity. This allows us to adopt the inertia moments obtained through many runs Figure 5.46 and is mentioned in Table 5.13, after being averaged.

Table 5.12: Accuracy of testbed in Prediction of Earth's gravity value (g=9.8 m/ s²)

g_1	g_2	g_3	g_4
9.7	9.8	9.6	9.9

The quadrotor inertia moments about I_x and I_y are almost identical, proves the quadrotor asymmetric.

Table 5.13: Quadrotor inertia moments from experimental and prediction approach

Inertia Moments (kg.m²)	I_x	I_y	I_z
Experimental values	0.0124	0.0128	0.0281
Predicted values	0.0127	0.0127	0.0231

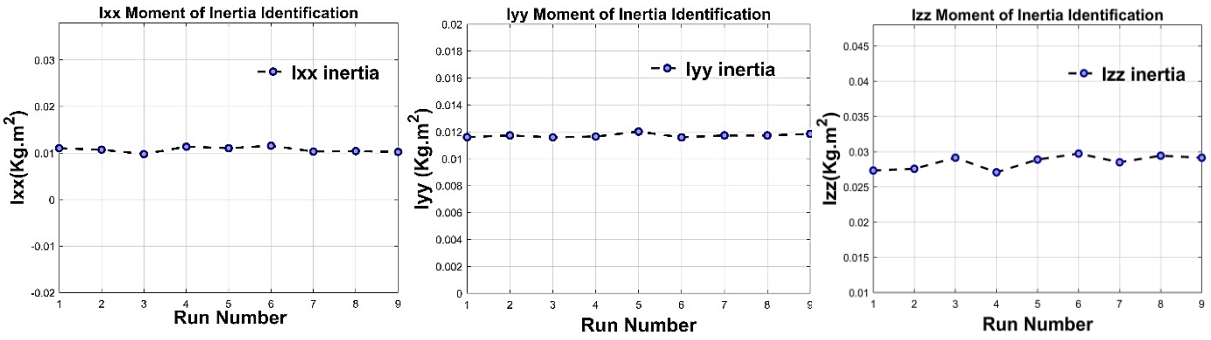


Figure 5.46: Quadrotor inertia-moments values I_x, I_y, I_z over many running

5.7.2 Black-Box Identification

Since the dynamics of a quadrotor is unstable, two feedback controllers are implemented to stabilize the attitude dynamics. The quadrotor angles by C_1 for each angle, and angular velocities each one by C_2 as showed in Figure 5.47, where y_1, y_2 are the dynamic response of angular velocity and the angle for each channel, G_1 is purely integrator r is the reference and d_s is the control disturbances while n_s is the outputs noise.

The first controller C_1, is used to maintain the system closer to the operating point, where the linearization condition is met and the attitude angles do not exceed 20°. Both controllers' gains are set by trial and error and are explained below:

C_1 is only proportional controller with: K_p=5.0; that maintains the minimal stabilization for angle.

C_2: is PID controller where: K_p= 1.3; K_i= 0.01; K_d= 0.025. Which maintains the minimal stability for attitude.

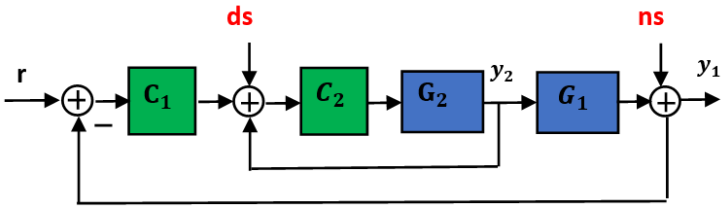


Figure 5.47: Quadrotor Roll & Pitch under cascade controllers (Maintains minimal stabilization for angular velocity)

5.7.2.1 PRBS identification Input

As mentioned before about a prior knowledge gained from analyses of quadrotor in open-loop responses, the dominant dynamics for roll and pitch and yaw attitude are located at a range of 0.1 to 20 rad/s as in Table 5.14. Then the good excitation on the spectrum between (0.1 rad/s) and the highest cutoff frequency (20 rad/s), which translates to τ_{max} τ_{min} used for the solve the equations (2.51) that gives N and Δt as: Δt =0.03 and N =424. Furthermore, the amplitude *a* is selected to obtaining a satisfactory compromise between the competing demands of angle limitation of linearity and signal-to-noise ratio in the measurement output, $a = 20°$ is the best choose. Our excitation input is persistently exciting of order 50, this means that we can estimating models of order lower than 50.

Table 5.14: The expected quadrotor attitude bandwidths and PRBS signal features

Roll Rate Bandwidth τ_{min} - τ_{max} **(rad/s)**	**Pitch Rate Bandwidth** τ_{min} - τ_{max} **(rad/s)**	**Yaw Rate Bandwidth** τ_{min} - τ_{max} **(rad/s)**	**PRBS Sequence Length N**	**Clock Period Δt (s)**	**PRBS Magnitude (a) (°)**
0.1 - 20	0.1 - 20	0.1 - 10	424	0.03	20

The experiments were repeated to obtain two distinct data sets, one for identification and the other for validation.

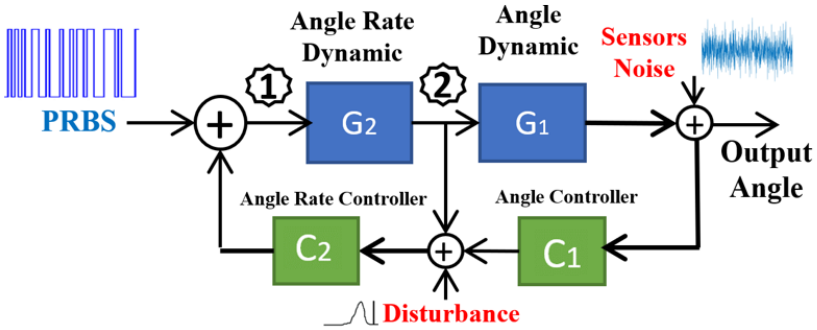


Figure 5.48: Quadrotor under PRBS excitation input and the data collected points

The target identification system G2 is subject to the sum of the well-selected PRBS signal and the output of angle rate controller C2, denoted as the signal presented at point ”1” in Figure 5.48. The response of the target system G2 to this combined signal is represented by point ”2”. The data collected from points 1 and 2 will subsequently be utilized for the identification of the dynamic behaviour of attitude G2.

5.7.2.2 Setting Up Testbench and Data Source for Quadrotor Dynamics Identification

To prevent the effect of interconnection between the (roll pitch yaw) dynamics from appearing in measured signals, we performed the test on a fixed axis where only the desired dynamic appears in measured signals. The nonlinearities in quadrotor dynamics have been treated by limiting the maximum angle of attitude to 20 degrees. An experimental dataset for quadrotor attitude identification is collected with the help of a serial communication protocol. Where the angles and their rates and accelerations are measured respectively by the both gyroscope and accelerometer. A magnetometer is used to gather data from the YAW dynamic. To overcome the close-loop identification effect, sensor parameters have been well-prepared as mentioned in filtering results section.

NB: *All outputs of the quadrotor model include integrators. The integrators are unstable numerically and cannot be identified, for this reason the modelling done*

through (U_1, U_2, U_3, U_4) control signals to the attitude derivatives outputs (p, q, r, z) was measured directly from inertial sensors, where: $p \approx \dot{\phi}$, $q \approx \dot{\theta}, r \approx \dot{\psi}$

5.7.2.3 Order Selection and Input Delay

As prior knowledge about the model and about order and its causality, and based on Akaike's information criterion. The model is almost of order two or more for each dynamic. and it can be said that the model parameter n_a, or ($\hat{G}(q, \theta)$ poles are not less than 2 and the input- parameter dependency n_b of ($\hat{G}(q, \theta)$ zeros is 1 or more. Finally, the delay-input n_k is predicted to be non-zero but less than nine.

5.7.2.4 Analysis of Experimental Results in Black-Box Identification

The identification is performed by using the power of MATLAB software, where different models are identified and validated for various orders as shown in a Table 5.15. The best model is selected through two main selection steps.

a- The first step is based on the accuracy, or the fit to identification and validation signals, where two kinds of signals have been used; square and PRBS signals. Residual analysis is used as another validation tool, where the error autocorrelation approach to the white-noise autocorrelation, which means, it must approach to the Dirac-pulse. The system causality is confirmed if we get zero for the negative lag in cross correlation.

b- In a second step, MATLAB Simulink was used to identify models placed in close-loop with pre-defined controllers (used by a real quadrotor), and comparison was performed between the quadrotor's response and the simulation model's response to the same input signal that was used in the identification and validation phases.

Table 5.15: Roll & Pitch Rate identified models ARX, ARMAX, BJ and their fits to training and validations signals.

Selected Model For (roll, pitch) rates	Validation-1 PRBS Signal Fitting (%)	Validation-2 Square Signal Fitting (%)	Training Fit (%)	BEST	Passes to 2^{ed} Eliminator Step
ARMAX4342	57.38	79.18	98.27	3	
ARMAX3331	**62.09**	**75.2**	**98.23**	**1**	**Pass**
ARMAX3231	59.39	61.61	98.15	2	
ARMAX3221	65.06	75.3	98.17	4	
ARMAX3211	57.08	70.77	97.92	5	
ARX10105	**75.84**	**80.15**	**98.3**	**1**	**Pass**
ARX221	51.59	45.52	96.92	3	
ARX220	45.9	39.04	96.94	3	
IV445	77.93	65.85	97.85	3	
IV554	**77.34**	**70.23**	**98.12**	**1**	**Pass**
IV445 refine	77.62	75.76	96.71	2	
BJ23221	**58.33**	**65.25**	**98.09**	**1**	**Pass**
BJ13321	61.58	66.91	98.16	2	

Note: In this context, (IV) refers to an ARX model, where parameter identification is achieved through the Instrumental Variable (IV) method, while ARX typically utilizes the gradient descent algorithm for the same purpose.

From Table 5.15, the selected models from the first validation step and based on identification and validation fit, the best models for each structure are ARMAX3331, ARX10105, IV554, BJ23221, and based on residual analyses that well interpret the mismatching between the identified model the one-step-predicted-output and the validation measured dataset output. The remaining models that are inside the 99% confidence interval are:

ARX10105: Discrete-time ARX model, $A(z)y(t) = \mathcal{B}(z)u(t) + e(t)$ Where Polynomial orders is na=10 nb=10 nk=5. Fit to estimation data: 98.3%, and 75.84% fit to PRBS and 80.15% fit to Square Signal

$$A(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_1 z^{-10}$$
$$\mathcal{B}(z) = b_0 + b_5 z^{-5} + b_2 z^{-2} + \cdots + b_{14} z^{-14}$$

Where: $b_1 = b_2 = b_3 = b_4 = 0$

Table 5.16: ARX10105 polynomial A(z), B(z) coefficients

a_0	1	b_0	0
a_1	-0.579	b_5	-0.005
a_2	-0.115	b_6	-0.004
a_3	-0.571	b_7	-0.004
a_4	-0.010	b_8	-0.003
a_5	0.0728	b_9	-0.002
a_6	0.035	b_{10}	-0.002
a_7	-0.012	b_{11}	-0.0017
a_8	0.075	b_{12}	-0.0008
a_9	0.052	b_{13}	-0.0007
a_{10}	0.055	b_{14}	-0.0007

IV554: Is ARX discrete time model, $A(z)y(t) = \mathcal{B}(z)u(t) + e(t)$. Polynomial orders na=5 nb=5 nk=4. Fit to estimation data: 98.12%, and 77.34% fit to PRBS and 70.23% fit to Square Signal.

The residual of the two models ARX10105 and IV554 are shown in Figure 5.49 to Figure 5.50.

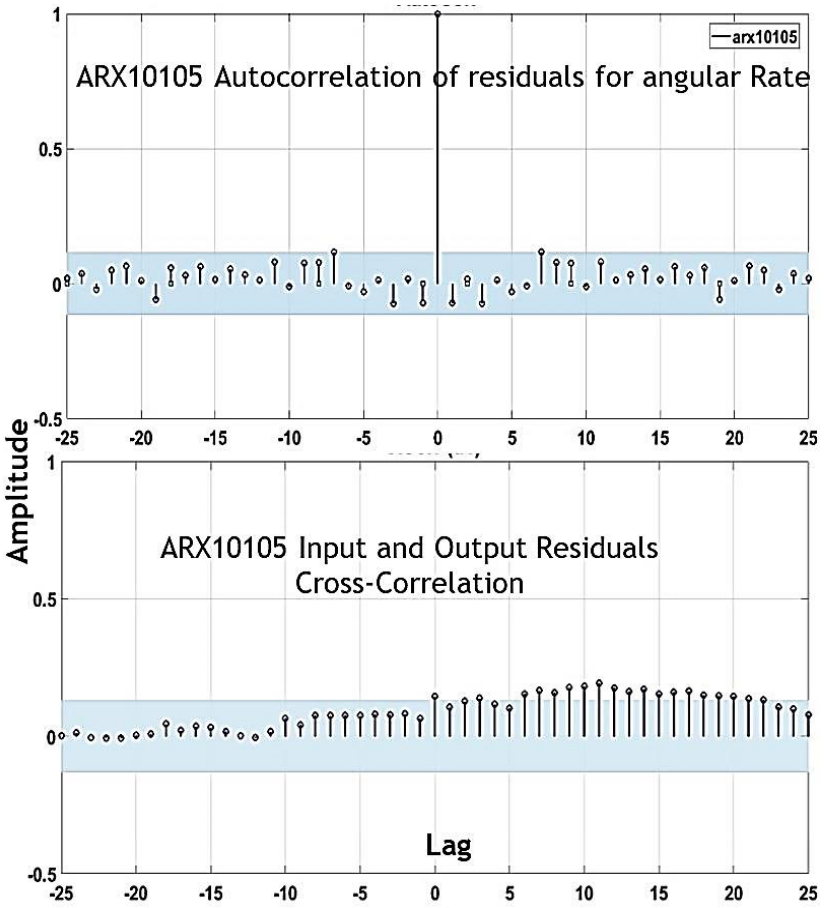


Figure 5.49: ARX10105 Residuals-Autocorrelation for angular Rate & ARX10105 Input and Output Residuals Cross-correlation

Where it is concluded that the identified models for output roll and pitch angular velocity are validated and the autocorrelation function is a pulse of Dirac. Since it is nearly within the 99% bounds for negative lag, causality is established. However, these two models can be considered useful. ARX10105 Residuals-Autocorrelation for angular Rate ARX10105 Input and Output Residuals Cross-correlation.

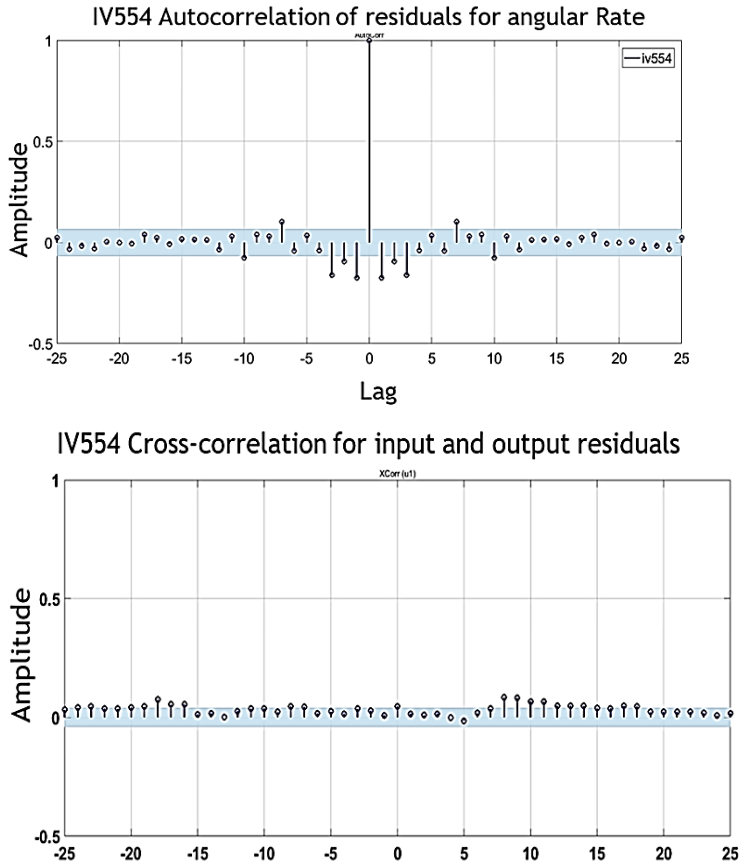


Figure 5.50: IV554 Residuals-Autocorrelation for angular Rate & IV554 Input and Output Residuals Cross-correlation

Figure 5.51, illustrates the extent of tracking the models for a PRBS input, which shows that both models almost have the same behaviour for signal tracking. The first stage has not until now helped in choosing of the best system between the two remaining models.

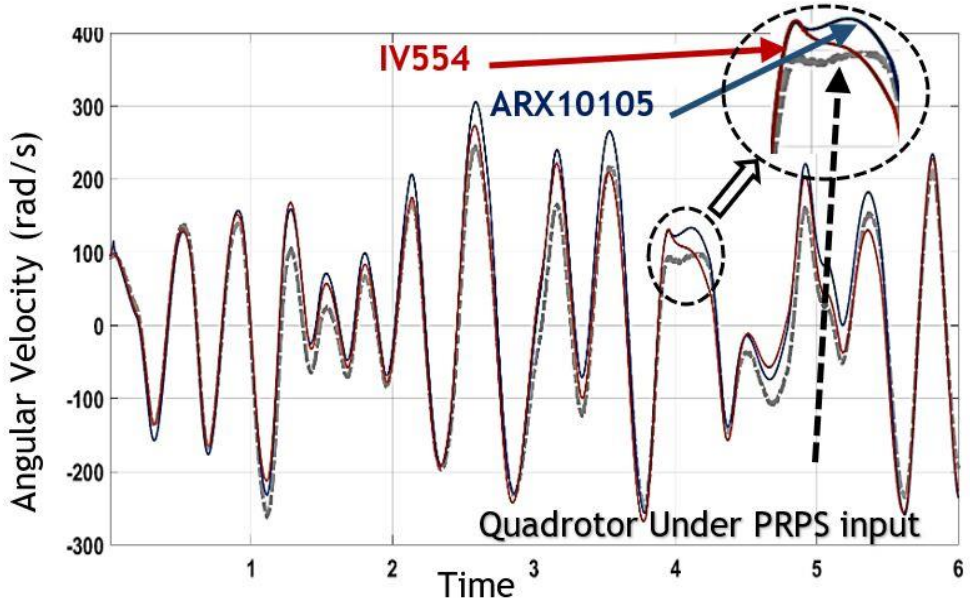


Figure 5.51: ARX10105 and IV554 models under PRBS signal input

The two remaining models from the first exclusion test have been examined under another critical exclusion test as already mentioned, where both models ARX10105 and IV554 were simulated under a real input-output signal gathered from a quadrotor in a close-loop using two cascade controllers for stabilizing the angle and angular velocity of the quadrotor Figure 5.52. Where the achieved response and the quadrotor collected response were later compared. The comparison step has done so through subjecting the quadrotor angles under PRBS input not fast, which later showed, through its response in Figure 5.53.a and 5.53b, for angle and its velocity. There was a significant similarity between both models and quadrotor response.

It was further noted that natural frequency and damping are almost the same for both models' responses. It was also noted that ARX10105 is a little more accurate than IV5504. This was confirmed by performing another experiment in which the PIDs gains are changed, and by using the root-mean-square error between the responses we found that the RMSE of ARX10105 & Quadrotor (RMSE =38.67) is less the RMSE of IV554 & Quadrotor (RMSE =52.53), this way confirmed that ARX10105 is an approximate model to the quadrotor dynamics.

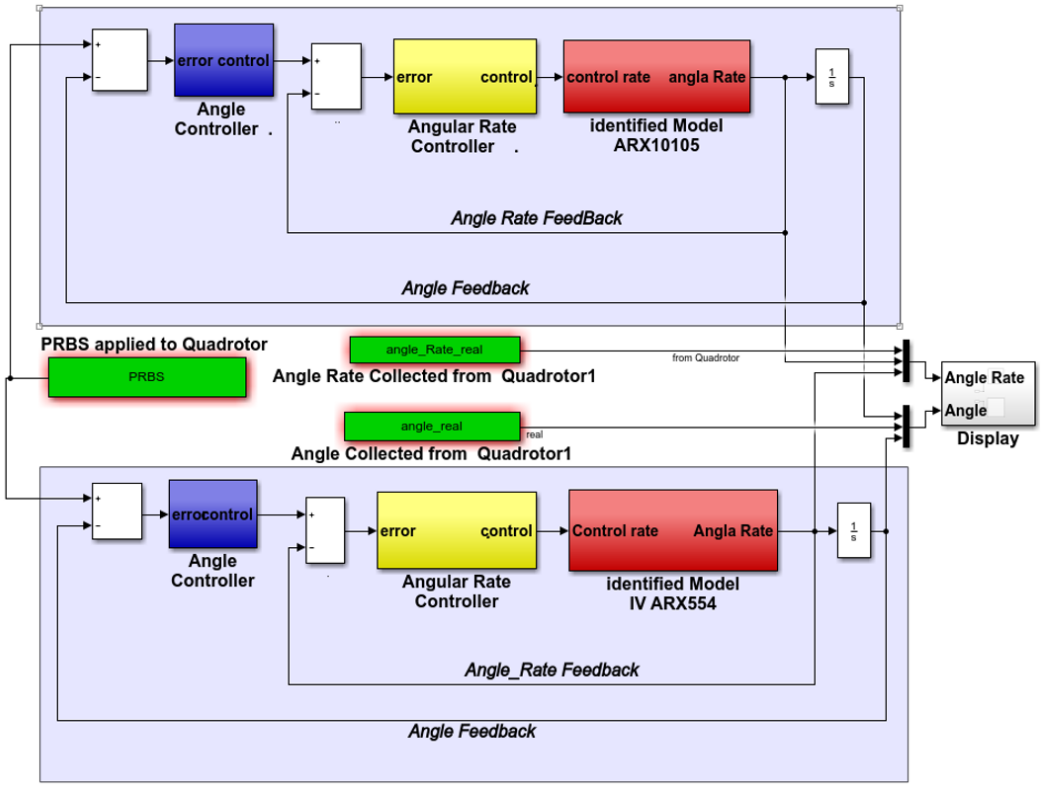


Figure 5.52: Comparative Simulation of ARX10105 and IV554 Model Responses with Real Input-Output

The quadrotor is symmetric about its two axes, x and y, in other words, roll and pitch dynamics can be treated the same way. In these experiments, the symmetry between roll and pitch response has been almost captured through the location of the pole and zeros, where tiny differences in the numeric values of the poles and zeros are caused by the asymmetries of inertial nonidentical properties resulting from improper mounting of the battery on the quadrotor frame or from some devices, e.g., sensors. In this work, we represented roll and pitch rate with the same model ARX10105.

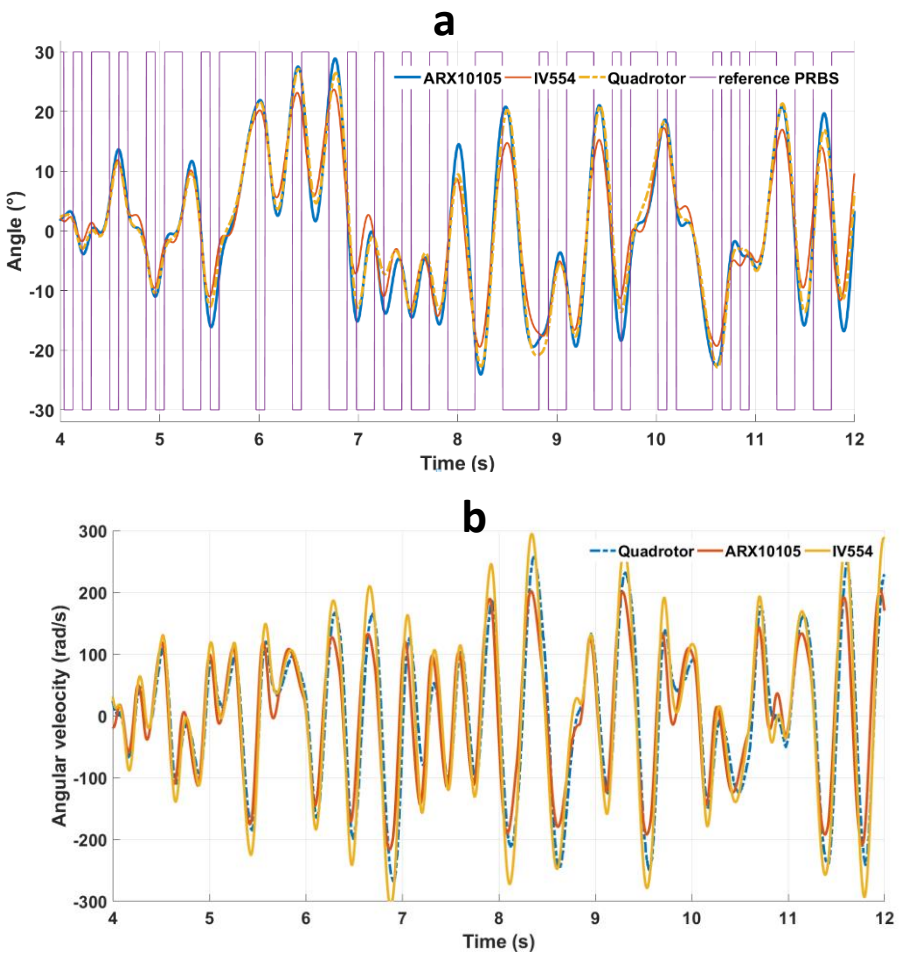


Figure 5.53: Real quadrotor under different PID gains and PRBS input (references) with the identified models (a) angle response**.** (b) angle rate response

Figure 5.54 shows the accuracy of pitch and roll rate responses, as the uncertainty range has a very small amplitude in both phase and magnitude across the entire bandwidth (approx. 0.1–20 rad/s). Since the objective of this research is modeling for the purpose of controlling, it is reasonable to predict some uncertainty at both higher and lower frequencies. Therefore, the accuracy of the model near the crossover frequency of 10.2 rad/s is the most important factor. This degree of model precision can be considered sufficient. For unstable models generated from closed-loop data, uncertainty analysis is particularly informative. Due to the low-frequency action of the feedback controller, where the true dynamics of the open-loop system will be hidden,

it is harder to get accurate information about the model at low frequency than it would be in an open-loop identification.

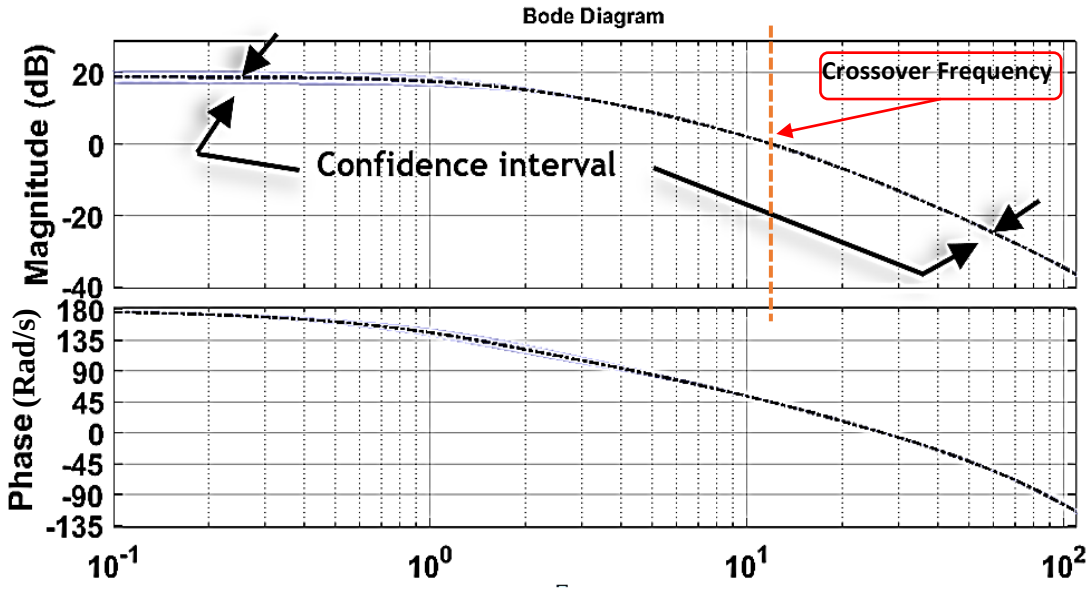


Figure 5.54: Pitch and Roll rates frequency response and the accuracy

The yaw rate dynamic has been identified and validated by passing through the same previous steps. However, unlike roll and pitch, its accuracy in open-loop is shown in Figure 5.55. using Matlab tool, we identified yaw rate dynamic in continues domain by the transfer function as below:

$$TF(s) = \frac{0.8009\ s - 3.141}{s\text{^}2 + 3.449\ s + 1.414}\ \text{(Yaw Rate Dynamic)}$$

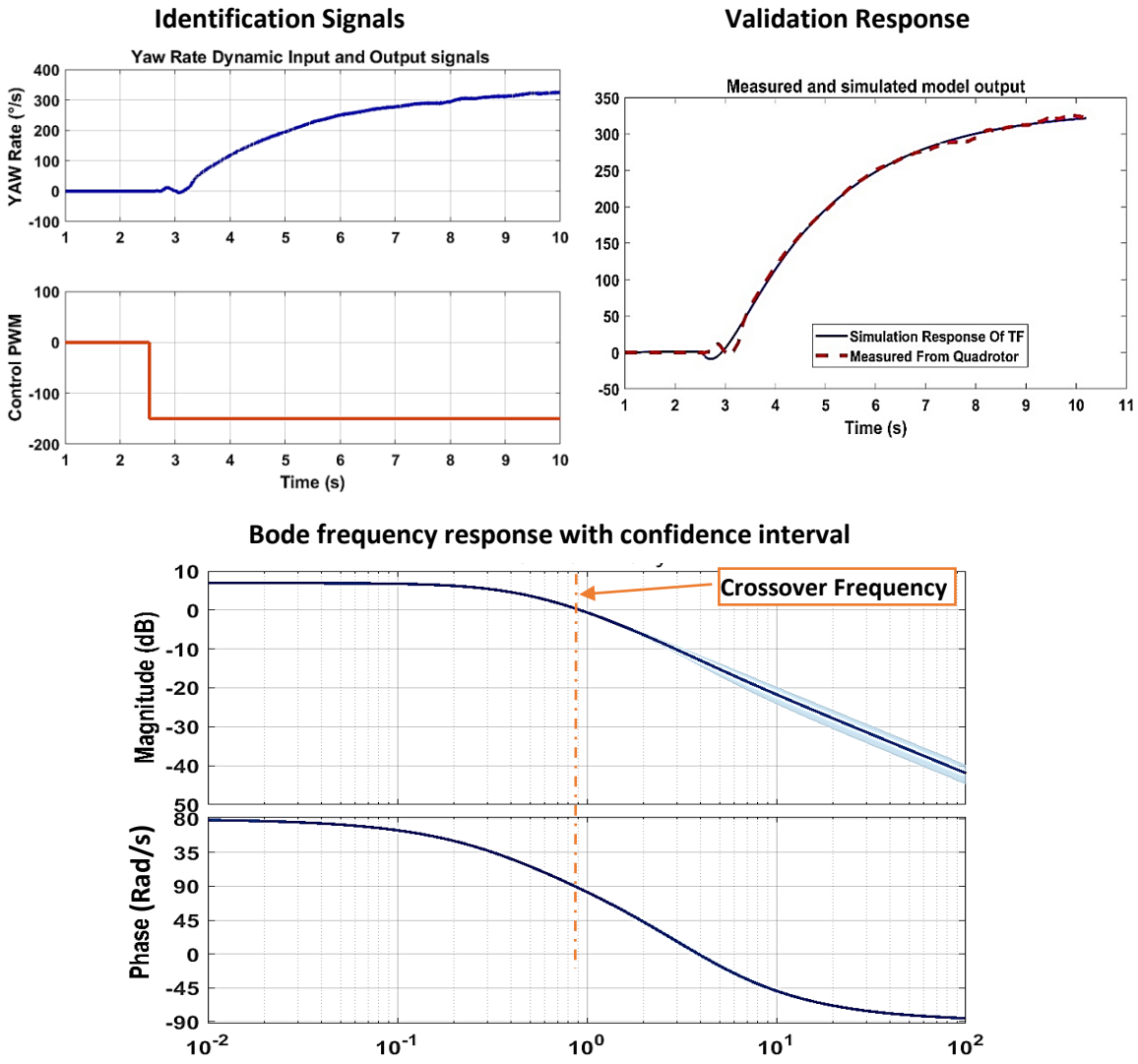


Figure 5.55: Yaw-Rate identified mode in open-loop responses

This level of identification accuracy is first confirmed in the identification and validation fit steps, and besides, confirmed by residual-output-autocorrelation and input-output-residual cross-correlation, which are totally inside the confidence interval. The yaw rate model can be considered adequate since the feedback controller low-frequency action is not present in open-loop identification. In addition, the frequency response reveals that the yaw rate dynamic is dominated at about 0.90 rad/s. Table 5.17 illustrates the quadrotor identified dynamics crossover frequencies.

Table 5.17: Quadrotor identified dynamics crossover frequencies

Quadrotor Dynamics	**Roll Rate**	**Pitch Rate**	**Yaw Rate**
Crossover Frequency (rad/s) About	10.2	10.2	0.91

5.7.3 Modelling by Neuro-fuzzy based

Achieving higher levels of performance in modeling requires the use of nonlinear techniques. In the present study, we employed the ANFIS neuro-fuzzy modeling method to enhance the performance level via fuzzy inference. Specifically, a neuro-adaptive learning-based ANFIS system modelling was utilized to determine the quadrotor attitude. Initially, we evaluated the input variables for the ANFIS model. Here, feature selection techniques sequential forward search algorithms have been employed to find the global optimal solution by systematically evaluating all possible input value combinations. These methods are guaranteed to identify the optimal solution, but it is computationally expensive and time-consuming, which is not an issue for us because our search spaces are relatively small. Because we intend to employ the ANFIS model as a controller by inversion, our regressor will have few variables.

First, we selecting ten ANFIS input candidates for our regressor, including the system's four inputs representing the attitude controller output, denoted as (u), and six outputs representing the attitude rate response, denoted as (y). Where here the input candidates are chosen to be [*u(k-1), u(k-2), u(k-3), u(k-4), u(k-5), u(k-6)],* while the outputs candidates chosen [*y(k-1), y(k-2), y(k-3), y(k-4)*]. While the ANFIS model output is *y(k)*. To minimize the total squared error, we employed the sequential forward search heuristic technique for input selection, progressively selecting each input. Figure 5.56 illustrates the sequential forward search process depicted in the search for optimal input combinations in the ANFIS model. Initially, the algorithm starts by using a single input and trains and validates the ANFIS model, calculating the respective train and validation RMSE (Root Mean Square Error). The process continues by iteratively adding inputs from the remaining possible combinations and evaluating their impact on minimizing the RMSE. The algorithm identifies the input combination that yields the minimal RMSE and retains it as the best combination. Subsequently, the procedure repeats with the addition of a third input, following the same methodology.

The result of feature selection algorithm SFS after iteratively searches for the best subset of features are three ANFIS inputs [*y(k-1), y(k-2), u(k-1)*] Where these inputs are the final subset that produces the best “RMSE”, Considering that the model constructed using these inputs demonstrates the lowest training RMSE and validation

RMSE, it can be inferred that this particular combination of inputs yields the most accurate and reliable results.

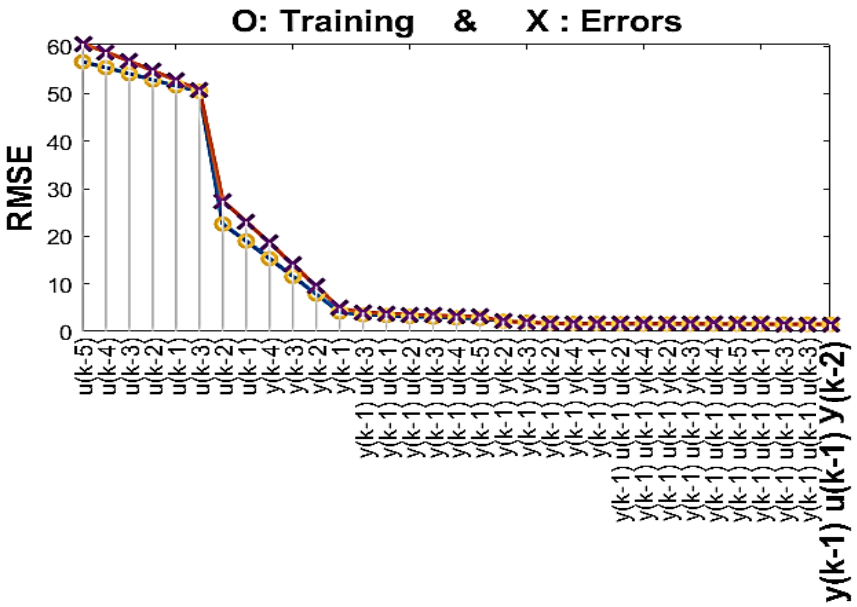


Figure 5.56: SFS in the search for optimal input combinations for ANFIS model

The model with the minimum validation error (*RMSE_checking=1.5074*) and the corresponding training error (*RMSE_training=1.4977*) was evaluated. The results of the evaluation are illustrated in Figure 5.57

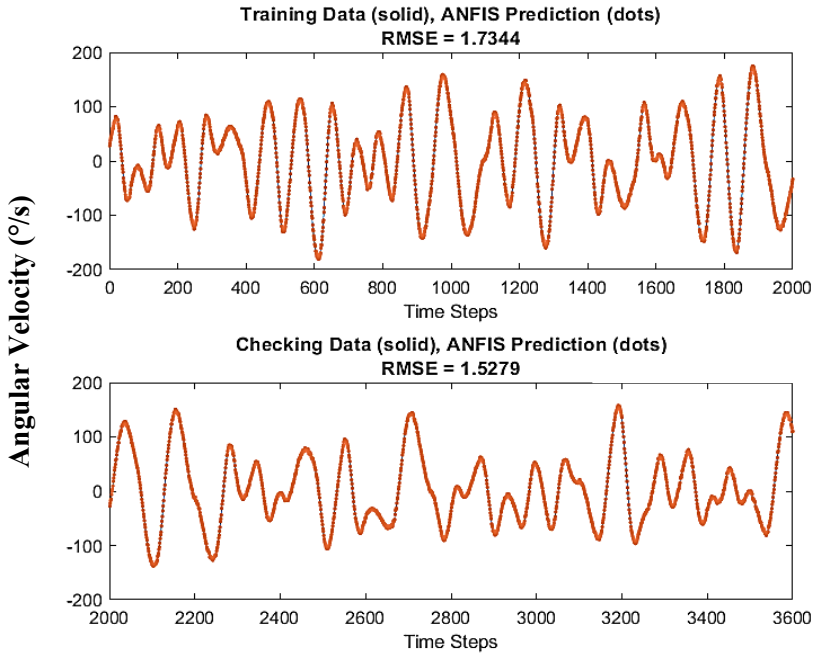


Figure 5.57: ANFIS Model response for training and validation signals

Where the identified model utilizes only six gaussian membership functions (MF), two for each input, including the past control inputs *u(k-1),* and the previous system output y(k-1) and the one before that, y(k-2), with only eight rules. and the Partitioning-based clustering technique grid-like structure is used for the best selection of input MF and reduces the number of rules. Figure 5.58 depicts the roll-rate dynamic's control input MF, with the output set to linear.

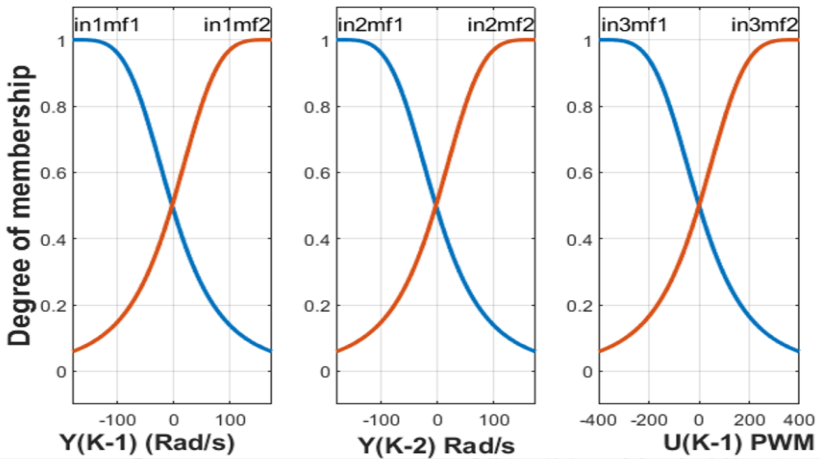


Figure 5.58: Quadrotor control inputs MFs of Roll & Pitch attitude rate dynamics shaped based on grid clustering

in Figure 5.59 illustrate Neuro-fuzzy network structure for quadrotor attitude rate dynamic use three inputs and one output. The output ranged between *[-180.46 to 175.05]* , For the linear outputs MFi are :

$f = a_1 x_1 + b_1 x_2 + c_1 x_3 + d_1$ where *MFi =[a b c d]*

MF1 = [1.2186 -0.23507 0.0082186 2.9959]

MF2 = [1.041 -0.010909 0.010717 -5.291]

MF3 = [2.3171 -1.5471 -0.013526 58.772]

MF4 = [1.6996 -0.45478 -0.050651 26.633]

MF5 = [2.0341 -0.78981 -0.04664 -64.238]

MF6 = [1.5449 -0.90082 0.0087429 -19.672]

MF7 = [1.0603 -0.02046 0.029156 4.8101]

MF8 = [1.2745 -0.2927 0.015373 -4.3333]

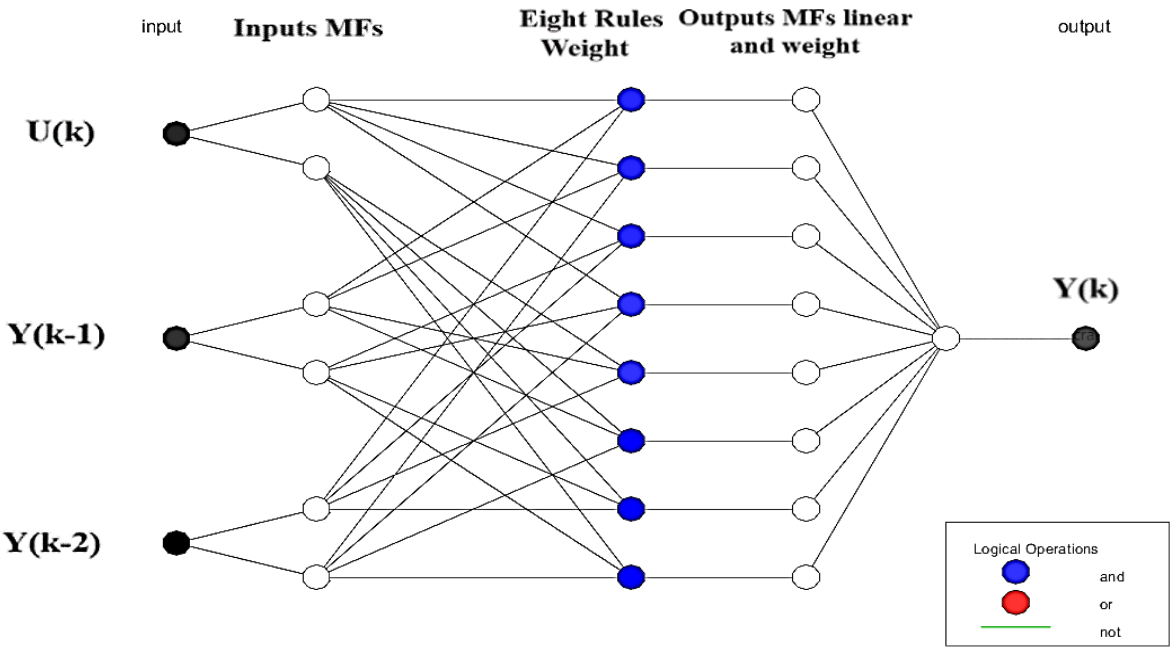


Figure 5.59: ANFIS network structure for attitude rate dynamic

And the eight rules are as follows:

Rule : **if** $y(k-1)$ **is** $A_{1,i}$ **and** $y(k-2)$ **is** $A_{2,i}$ **and** $u(k-1)$ **is** $A_{3,i}$ **Then** $y(k) = MF_i$

Where i=*1,..,8*. And

$A_{1,i} = [\ In1Mf1 \quad In1Mf1 \quad In1Mf1 \quad In1Mf1 \quad In1Mf2 \quad In1Mf2 \quad In1Mf2 \quad In1Mf2\ ]i$

$A_{2,i} = [\ In2Mf1 \quad In2Mf1 \quad In2Mf2 \quad In2Mf2 \quad In2Mf1 \quad In2Mf1 \quad In2Mf2 \quad In2Mf2\ ]i$

$A_{3,i} = [In3Mf1 \quad In3Mf2 \quad In3Mf1 \quad In3Mf2 \quad In3Mf1 \quad In3Mf2 \quad In3Mf1 \quad In3Mf2]i$

The ANFIS model behaviours compared to the quadrotor attitude rate *G2* were simulated under a real input-output signal gathered from a quadrotor in in Figure 5.60.

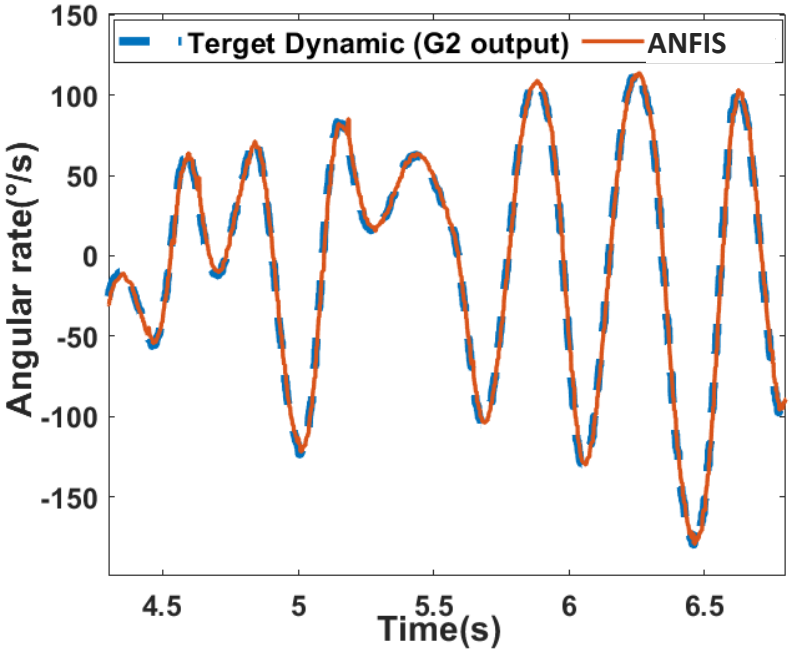


Figure 5.60: ANFIS model output tracking attitude rate dynamic (*G2*) response

5.8 Quadrotor Control

In this section, we will present the experimental results of quadrotor control using different controllers, such as PID, LQR, and control structures. Additionally, we will discuss how to utilize the gathered models from the identification step to fine-tune the controller gains before applying them to the actual quadrotor. This approach not only saves time but also reduces the risk of damaging the quadrotor during testing. Moreover, we will analyse the performance of each controller. By leveraging the identification model, we aim to optimize the controller's performance and enhance the overall quadrotor control experience.

5.8.1 PID controller

In the realm of quadrotor control, achieving stable and precise performance is of paramount importance. To enhance the control system's efficiency, we have adopted a validated model at hand (ARX10105), to redesign the controller so as enhance the performance of the closed loop while retaining the same control structure. To acquire

the suitable controller parameters, a tuning using MATLAB Simulink was performed as following:

1- Using MATLAB Simulink for simulate the quadrotor inner and outer loop, where in the inner loop we replace the angle rate dynamic G2 by the one validated in the identification phase ARX10105. In the outer loop which represent the angle dynamic G1 we replace it by purely integrator (1/s). Two controller C1, C2 that stabilised the quadrotor inner and outer loop, were simulated using their previously used gains from the identification phase (they were not optimal gains, but maintain the minimal stability for hole quadrotor dynamics).

2- The tuning of the two controller C1, C2 gains was performed in Simulink environment where we used many techniques first trail-and-error and later optimisation ones. To fine-tune the gains of controller C1 and C2, optimization was performed in a SIMULINK environment using two options:

5.8.1.1 First option of tuning: involves tuning the controllers C1 and C2 separately.

In the **first step**, as shown in Figure 5.61.a we focus on tuning only the inner loop controller (angle rate) C2, which is a PID controller. We utilize the PID Tuning Matlab tool, with the angle rate as the feedback signal, to automatically adjust the gains of the PID controller for a SISO plant ARX10105. The goal is to achieve a balance between performance and robustness, ensuring a response time of less than 2 seconds and minimal displacement. After tuning the gains, they are updated in the discrete PID block in Simulink, and a simulation is performed to observe the system response with the new gains.

In the **second step**, we introduce the outer loop to the inner loop (using the newly tuned gains) and repeat the same process to tune the outer loop controller C1, which is a proportional controller. The results of this option are presented in Table 5.18. Figure 5.61.b illustrating the outcome of separately tuning the controllers C1 and C2 using the PID Tuner tool.

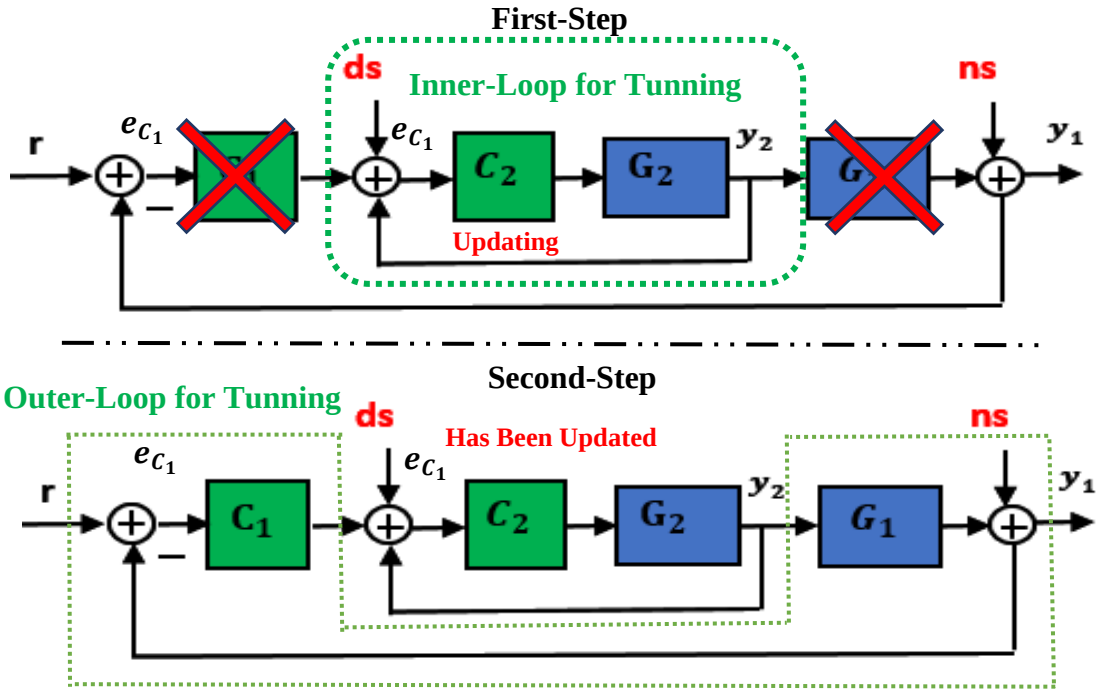


Figure 5.61: Separately tuning the controllers C1 and C2 using the PID Tuner tool

5.8.1.2 Second option of tuning: A heuristic tool called a Genetic Algorithm (GA) is utilized to tune the gains of both controllers C1 and C2 simultaneously, aiming to find the best combination between them. The choice of a GA is motivated by its ability to converge to the global optimal solution. GAs tends to be better suited for finding good solutions within a reasonable time frame, especially when the global optimum is difficult to find using deterministic methods. The optimization process involves minimizing a performance metric known as the Integral of Time Multiply Absolute Error (ITAE). To begin, the initial GA parameters for the PIDs controllers are determined using a heuristic trial-and-error method. These initial gains are selected based on the identification process, ensuring the quadrotor's stability is maintained at a minimum.

The GA employed in this option consists of 20 generations and a population size of 25. The fitness function used in the GA evaluation, is the sum of the errors $\boldsymbol{F_{ITAE\,Total}}$. The fitness function is defined as follows:

$$\boldsymbol{F_{ITAE\ of\ C_1}} = \int_0^t \boldsymbol{t|e_{C_1}(t)|dt} \quad \text{and} \quad \boldsymbol{F_{ITAE\ of\ C_2}} = \int_0^t \boldsymbol{t|e_{C_2}(t)|dt}$$

$$F_{ITAE\ Total} = F_{ITAE\ of\ C_1} + F_{ITAE\ of\ C_2}$$

Here, t represents time, and e_{C_1}, e_{C_2} represent the errors between the setpoints (target angle and target angle rate) and the feedback states.

The four gains required for controllers C1, C2 are encoded as chromosomes (k) within the GA optimization process.

K (Chromosome structure) =

$C_{1\,Kp}$	$C_{2\,Kp}$	$C_{2\,Ki}$	$C_{2\,Kd}$

Table 5.18: Comparison of C1 and C2 Tuning Between PID Tuner Tool and GA

	Controller C1			**Controller C2**
	Kp	Ki	Kd	Kp
PID Tuner Tool	0.346	0.535	0.0264	1.57
Genetic Algorithm	0.523	0.0289	0.0254	2.82

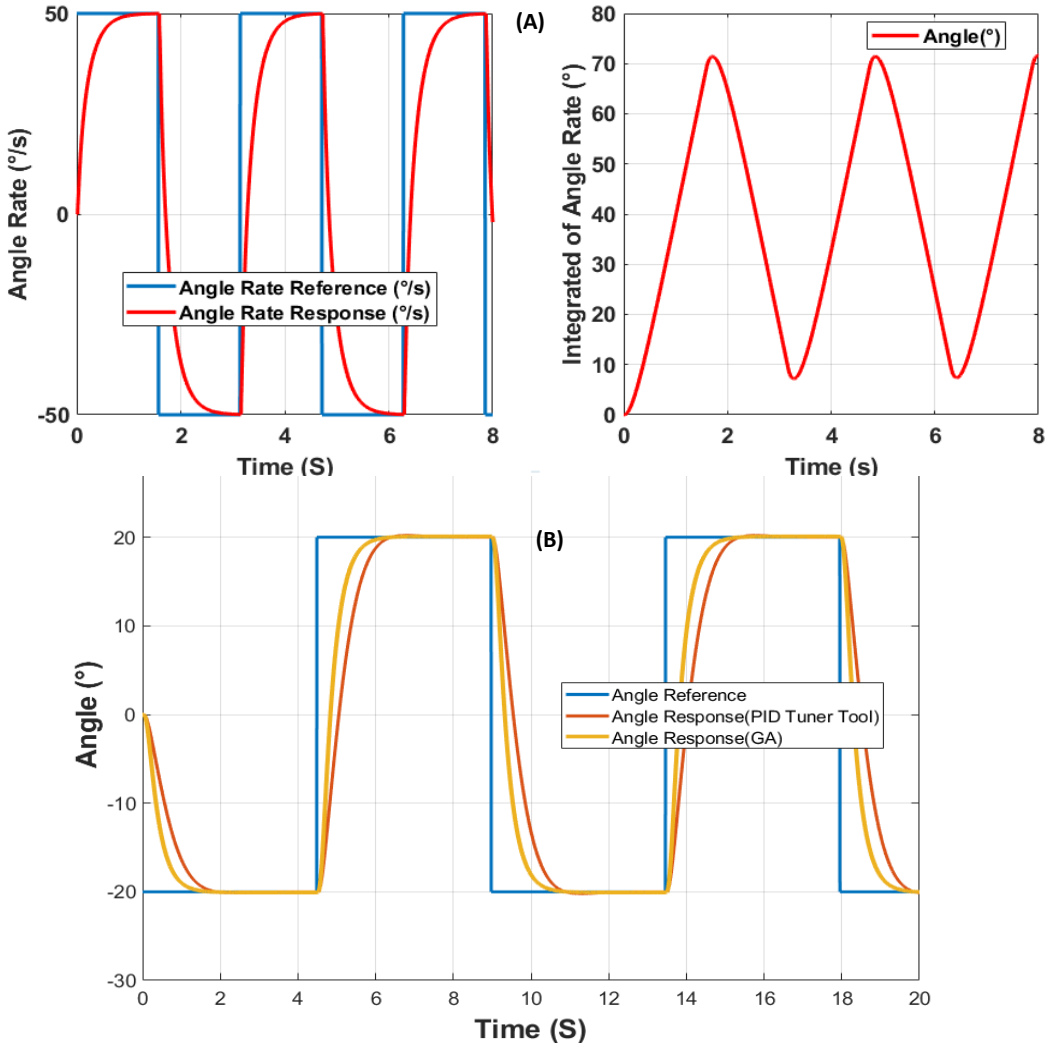


Figure 5.62: Comparative Quadrotor Responses for Inner and Outer Loops - PID Tuned by MATLAB Tune-Tool vs. PID Tuned by GA (A) inner-loop (B) Outer-Loop

Based on the response from the quadrotor ARX10105 Figure 5.62, we can observe that the system designed based on GA performs slightly better than the system designed using PID tuning tools. This indicates that GA has discovered the a highly effective solution for the entire system. In fact, the obtained gains from both options are very close to each other. However, since we intend to apply these gains to the actual quadrotor, our objective is to narrow down the research range for each gain. For example, the proportional gain of C1, which is suitable for the real quadrotor, lies approximately between 1 and 5. Nonetheless, there may be minor adjustments required when applying these gains to the actual quadrotor to achieve the optimal response. Subsequently, we discovered that the optimal gains for C1 and C2, which ensure stability and accurate trajectory tracking of the real quadrotor, are:

C_1 P controller, where: K_p=3.0.

C_2: PID controller where: K_p= 0.55; K_i= 0.045; K_d=0.01.

The application of the new gains leads to a substantial enhancement in real quadrotor performance compared to the previous parameters, as demonstrated in Figure 5.63, for roll and pitch angles and Figure 5.64 shows roll angles tracking a variable square reference and the response of its derivative (roll rate), demonstrate a considerable increase in quadrotor performance relative to the previous settings. The yaw angle response rate under square input is depicted in Figure 5.65.

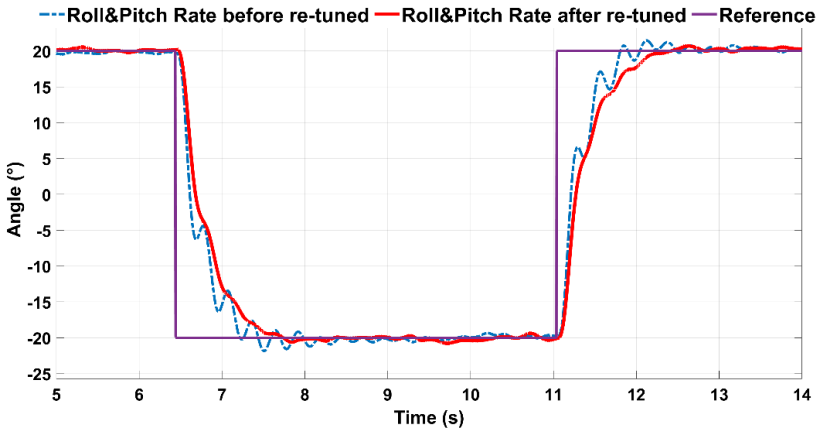


Figure 5.63: Roll and Pitch Angle Responses of the Real Quadrotor Before and After PID Gain Retuning

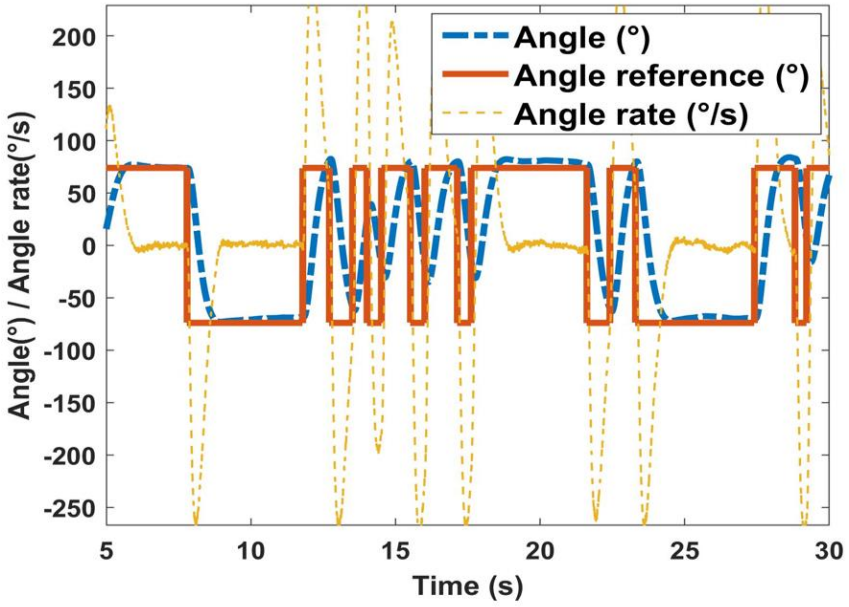


Figure 5.64: Quadrotor angle roll and its rate tracking variable square reference after tuned the PIDs gains

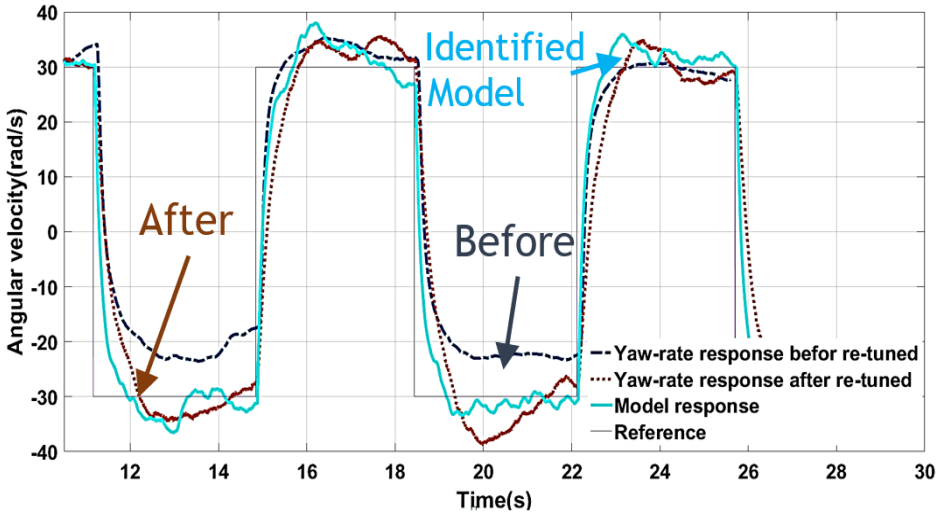


Figure 5.65: Real quadrotor Yaw rate response before and after re-tuned the PID gains and the response of identified model under the same input

As long as we possess an accurate quadrotor dynamics model that effectively represents the real quadrotor's behaviours to various inputs, while that assumption is true, a linear controller can easily be designed by relying only on the identified model. But unfortunately, this argument is not always correct. Forcing reality to fit the model is not possible to some degree because a model is never true or correct. There are always attempts to develop more effective identification techniques that could explain reality more closely and more precisely. As long as the identified models that mimic the quadrotor behaviours are available, any controller can now be designed only by using MATLAB tools without the need for experimental attempts on a real quadrotor, which would save a lot of time and effort and material damage.

5.8.2 State Feedback Control for Quadrotor Attitude

The State Feedback Control, as mentioned in the theoretical part, involves the use of the state vector to compute control actions for specified system dynamics using feedback gains. The state feedback is designed to guarantee closed-loop system stability and certain performance criteria. In this work, we have used two methods to determine state feedback gains. In the first, the gains are adjusted using genetic algorithms assuming that the system is controllable. In the second, we used the LQR (Linear Quadratic Regulator) approach.

In the first method, we used a genetic algorithm optimization tool within the MATLAB environment (discrete) to achieve this. Additionally, we incorporated the identified model of the quadrotor's attitude, G2, specifically, the more accurate model ARX10105, into a closed-loop system with G1, the discrete integrator.

For LQR, we based on tuning of weighting matrices Q and R tuned based on desired performance criteria such as rise time and settling time or by employing Bryson's rule. Moreover, we employed state-space equations to represent the linearized model of the system, incorporating our quadrotor parameters obtained from white and grey-box identification, such as the quadrotor moment of inertia. Through the implementation of these techniques, our aim was to stabilize the attitude of the quadrotor using the LQR regulator. The work provides a comprehensive overview of the various techniques and approaches that can be used to effectively stabilize such systems and overcome the challenges posed by their instability.

5.8.2.1 States feedback optimization by genetic algorithm

The feedback gains (*K*) were adjusted by utilizing the identified model of quadrotor attitude, denoted as G2 which is discrete and represented by "ARX10105." This model was integrated into a closed-loop system with *G1 (*Discrete-Time Integrator*)* and the state feedback, specifically the angle and its rate as depicted in Figure 5.66.

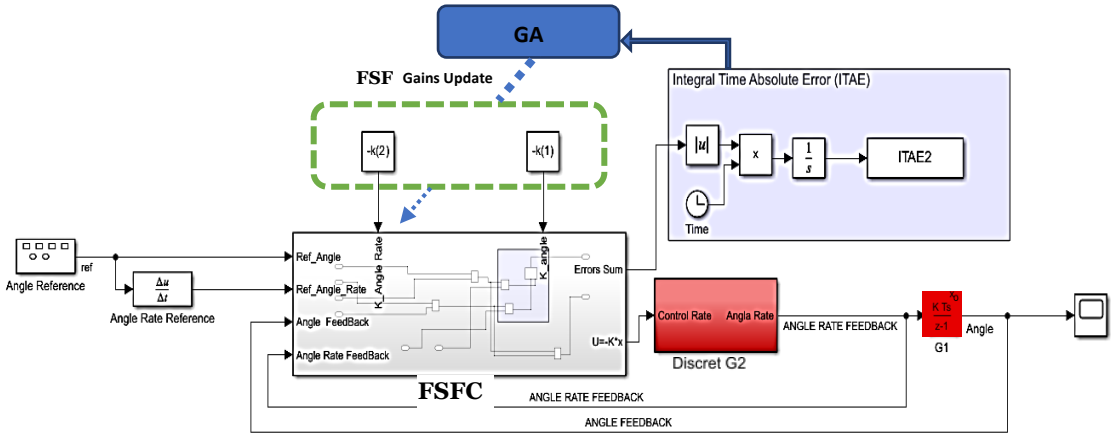


Figure 5.66: Tuning the states feedback gain *K* using GA

For each channel of quadrotor attitude, two gains, namely *k1* and *k2* were controlling the channels angle and it derive. A genetic algorithm was employed to

optimize these gains. The fitness function was defined as the Integral Time Absolute Error (ITAE) between the states and their desired signals. Through the execution of 25 generations with a population size of 30, the genetic algorithm successfully identified the optimal gains that minimize ITAE, shown in Table 5.19, thereby stabilizing the quadrotor during its trajectory tracking with minimal possible error as shown in Figure 5.67. To date, our efforts have yielded successful tuning of the state feedback gain, leading to the stabilization of the "ARX10105" model.

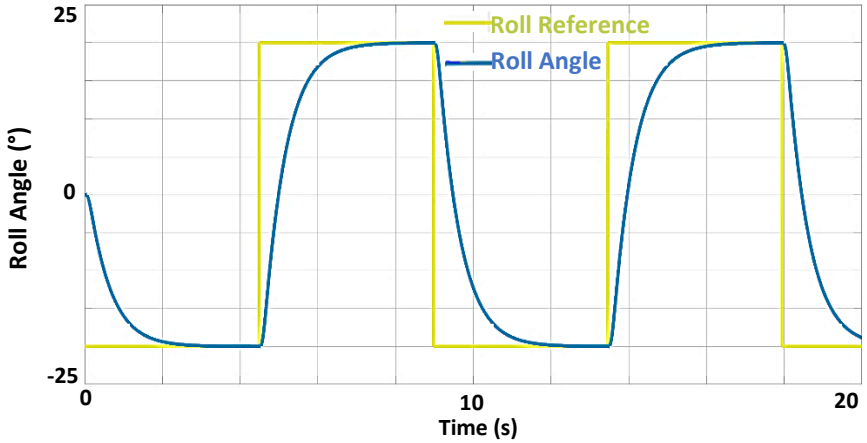


Figure 5.67: Simulation of quadrotor roll under FSFC tracking square signal

Table 5.19: FSFC gains tunning by help of GA

	Roll Rate	**Pitch Rate**
k1	6.72	6.72
k2	1.44	1.44

The gains obtained from the optimization phase were subsequently implemented in the quadrotor, resulting in a remarked improvement in its performance, as demonstrated by the results depicted in the accompanying Figure. 5.68.

5.8.2.2 LQR approach

A) Weight Matrix Q, R

The weighting matrices Q and R, are used to help in finding the optimal gain *K* of LQR, Q and R, are used to determine the relative importance of the states and control

inputs in the cost function. For example, if we want to prioritize a faster rise time, we can assign a higher weight to the states that contribute to the rise time in matrix Q, and a lower weight to the control inputs in matrix R. When considering the LQR design, it involves following two primary steps:

1) **The first step** involves defining the representation of the attitude dynamics to be used in the LQR design process. Given that ARX10105 is a 10th-order model, for simplicity, we have opted to work with a 2nd-order transfer function obtained from the identification process. Additionally, we considered the linearization model as a secondary option.

1-a) ***Using Second-order transfer function*** **(TF)** obtained from the identification step, which can be transformed into either continuous or discrete state space representation (the only matter here is the controller should be discrete to be implemented in the embedded system). These representations are then utilized in conjunction with the Q and R matrices to tune the LQR feedback gains.

TF (s) = $\left(\frac{7.685\ s - 189.6}{s\text{^}2 + 19.91\ s + 75}\right)$, which is obtained from the identification step (MATLAB identification toolbox), that minimally mimics the attitude of the quadrotor.

We defined the state-space representation of the LTI model, including the state matrix A, input matrix B, output matrix C, and direct feedthrough matrix D. from TF

Continuous-time state-space: A = $\begin{pmatrix} -19.91 & -9.376 \\ 8 & 0 \end{pmatrix}$, B= $\begin{pmatrix} 4 \\ 0 \end{pmatrix}$, C= $(1.921 \quad -5.926)$,

D=(0) Where x_1 is the roll or pitch and x_2 is the corresponded rate.

And its controllable canonical form: A = $\begin{pmatrix} 0 & -75 \\ 1 & -19.91 \end{pmatrix}$, B= $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$, C= $(7.685 \quad -342.7)$

The system is controllable since the rank of the controllability matrix (A, B) is 2, which is equal to the system's dimension. Controllability guarantees that there exist control inputs that can steer the system's states according to the desired objectives, regardless of the ability to observe the system's internal states accurately.

1-b) **Using Linearized dynamics**, as explained, linearized model can also be employed for this purpose, in this approach, the identified values of I_x, I_y, which

are provided in Table 5.13, are substituted into the linearized model to facilitate the design of the Q and R matrices as following:

$$\begin{cases} \ddot{\phi} = \frac{1}{I_x} U_2 \\ \ddot{\theta} = \frac{1}{I_y} U_3 \\ \ddot{\psi} = \frac{1}{I_z} U_4 \end{cases} \Rightarrow \frac{\mathrm{d}}{\mathrm{dt}} \begin{pmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 1/I_x & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1/I_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1/I_z \end{pmatrix} \begin{pmatrix} U_2 \\ U_3 \\ U_4 \end{pmatrix}$$

Let us decompose the state space into sub-state spaces, specifically considering each channel of roll, pitch, and yaw separately. This decomposition aims to enhance clarity and facilitate individual tuning of each channel, as there is no interaction between the channels in the linearized representation. The resulting representation is as follows:

- $\ddot{\phi} = \frac{1}{I_x} U_2 \Rightarrow \begin{pmatrix} \phi \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} \begin{cases} x_1 \\ \frac{1}{I_x} U_2 \end{cases} \Rightarrow \begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{I_x} \end{pmatrix} U_2$ (Pitch)
- $\ddot{\theta} = \frac{1}{I_y} U_3 \Rightarrow \begin{pmatrix} \theta \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x_3} \\ \dot{x_4} \end{pmatrix} \begin{cases} x_4 \\ \frac{1}{I_y} U_3 \end{cases} \Rightarrow \begin{pmatrix} \dot{x_3} \\ \dot{x_4} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{I_y} \end{pmatrix} U_3$ (Roll)
- $\ddot{\psi} = \frac{1}{I_y} U_3 \Rightarrow \begin{pmatrix} \psi \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} \Rightarrow \begin{pmatrix} \dot{x_5} \\ \dot{x_6} \end{pmatrix} \begin{cases} x_5 \\ \frac{1}{I_y} U_4 \end{cases} \Rightarrow \begin{pmatrix} \dot{x_5} \\ \dot{x_6} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_5 \\ x_6 \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{1}{I_z} \end{pmatrix} U_4$ (Yaw)

Because our sensor directly measures the angular rate, and considering that the integration of the angular rate is part of the output system, then:

$y_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, the same thing for other outputs y_2, y_3.

Where I_x = 0.0124 I_y=0.00128 I_z=0.0281

2) **The Second Step** is the tunning of the weighting matrices Q and R, we defined the desired rise time, settling time, and steady-state response characteristics. In our case each sub-state spaces for each channel have a simple two states (angle and its rate), and because we want to design a control system that stabilizes the quadrotor attitude with achieving a desired rise time and settling time as following:

- Desired Rise-Time < 1 seconds
- Desired Settling-Time < 2 seconds

First option in choose Q and R that reflect the desired performance, common

approach is to set the elements of Q proportional to the desired state error weights, and set the elements of R proportional to the desired control effort weights. To reflect these performance requirements in the weighting matrices Q and R, we do the following:

Assign a high weight to the state variable x_1, which contributes to the rise time, by setting the first element of Q to a large value and assign a lower weight to the control input, which affects the settling time, by setting the elements of R to a smaller value

Our choice was: $Q_1 = \begin{pmatrix} 100 & 0 \\ 0 & 30 \end{pmatrix}$ and $R_1 = (1)$ (**First option)**

NB: *Note that the choice of values for Q and R is often an iterative process and depends on desired performance. Our choice of Q and R indicating that we want the control system to prioritize a fast rise time over a small control effort, and to penalize large state errors more than high control efforts.*

Second option in choose Q and R, we based on Bryson's Rule which provides a simple formula for setting the values of the Q and R matrices based on the states and control maximum permissible values. In our linearization condition we assumed the maximum acceptable angle is 20° which equal in radian to 0.35 in these moments its rate is about 89 °/s which equal to 1.55 rad/s.

Bryson's Rule states that the Q and R matrix should be chosen as:

$$Q_2 = \begin{pmatrix} \frac{1}{\textbf{maximum acceptable value of } (x_1^2 = \phi \textit{ or } \theta)} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\textbf{maximum acceptable value of } (x_2^2)} \end{pmatrix} = \begin{pmatrix} \frac{1}{0.35^2} & 0 \\ 0 & \frac{1}{1.55^2} \end{pmatrix} = \begin{pmatrix} 8.16 & 0 \\ 0 & 0.41 \end{pmatrix}$$

Because our controller saturation set at 400, we conclude

$$R_2 = \frac{1}{\textbf{maximum acceptable value of } (u_1^2)} = 1/400 = 0{,}0025$$

Means $Q_2 = \begin{pmatrix} 8.16 & 0 \\ 0 & 0.41 \end{pmatrix}$ and $R_2 = (0.0025)$ (**Second option**)

The two calculated values of Q and R, corresponding to the two available choices,

are employed in tuning the discrete-time LQR to obtain the feedback gain matrix *K*. This process is facilitated by utilizing the two available representations of the attitude dynamics.

The Table 5.20 below presents the results of this process, showcasing several approximate solutions for the feedback gain:

Table 5.20: LQR feedback gains results via several options

		Attitude-Dynamics G2 Model from	
		Transfer-function	**Linearized-dynamics**
Weigh-Matrices Q and R by	**First-Option**	K1=6.2724 3.3090	K2 = 4.9891 2.7554
	Second-Option	K3=35.2289 6.6981	K4=15.7379 3.5831

After incorporating the feedback gains *Ki* into the closed-loop system, which exhibits enhanced response characteristics. This phenomenon is illustrated in Figure 5.68, the figure also showcases the response of the ARX10105 model and the actual quadrotor system for identical gains. These responses are plotted while the system traces a square signal with amplitude of 20°.

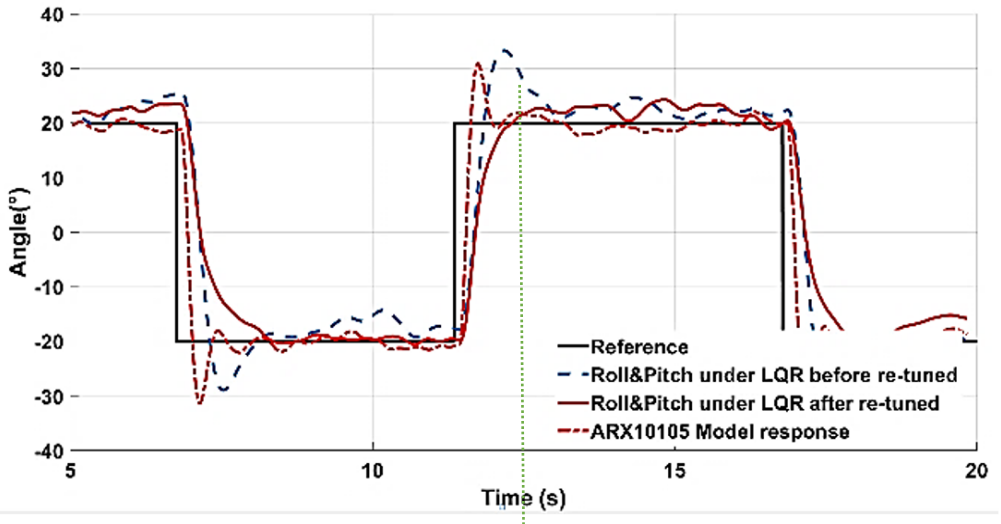


Figure 5.68: Real quadrotor roll and pitch angles response before and after re-tuned LQR

As noted previously, the identified model is an attempt to fit reality to some degree. This is visible in the ARX model response, which came close to simulating roll and pitch quadrotor response with a little skewed slightly. In its best case, it is valid and possibly credible. We must always be ready to modify and develop a model to include new

observations and new facts, and we cannot disregard the phenomena that conflict with the model.

5.8.3 Quadrotor at Hovering Position

All prior control designs for the quadrotor and corresponding responses were conducted by isolating the attitude axes, thereby treating each quadrotor's dynamics independently. This approach involved allowing the dynamics of the quadrotor under consideration to evolve while constraining the influence exerted by the other dynamics of the quadrotor. Presently, having completed the calibration of all attitude dynamic controllers, it becomes necessary to evaluate their interactions within a three-dimensional context. To this end, a three-dimensional testing apparatus, referred to as the "3D test bench," was devised. A visual representation of this test bench is provided in Figure 5.69.

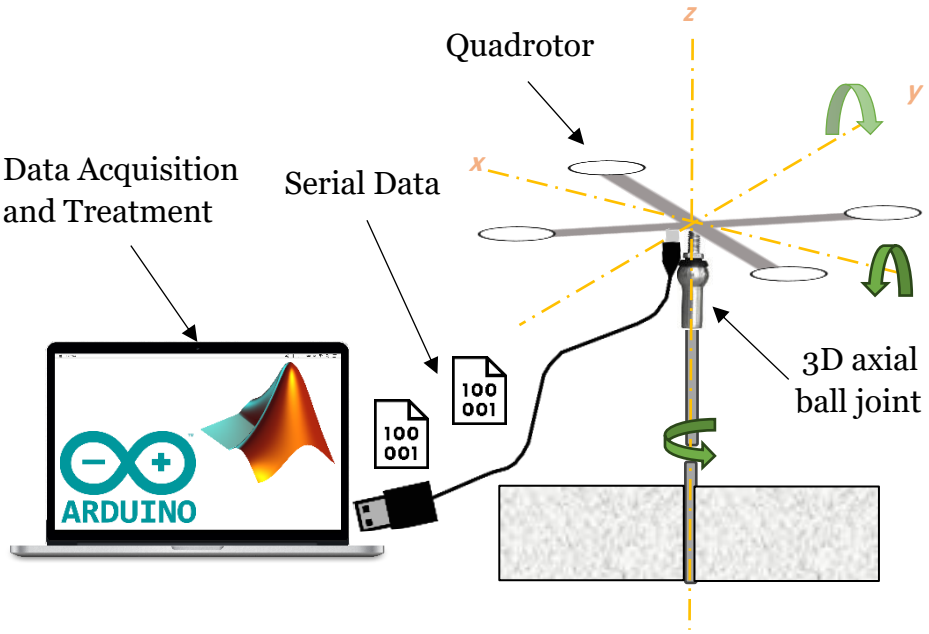


Figure 5.69: Experimental Assessment of Quadrotor Attitude Using a 3D-test-bench

The implementation of this test bench involved the acquisition of a specialized three-dimensional axial ball and socket joint. This joint enabled angular movements around the x, y, and z axes, effectively inhibiting any unintended displacements. Significantly, this test bench facilitated the interaction among the various attitude channels. After power on the quadrotor, a command was issued to maintain a hovering

configuration, characterized by all angles being set to 0°, while the quadrotor was under PID controllers tuned from the previous step. The response data emanating from the quadrotor during this hovering state was captured utilizing a serial communication protocol. Two distinct communication modalities were employed for this purpose, a wired connection and a wireless transmission. In the first option, a USB cable was employed, while in the subsequent alternative or the second option, a telemetry system was employed. The response of the quadrotor in a hovering state, where all angles and angular rates were set at 0° and 0°/s respectively, has been depicted in Figure 5.70.

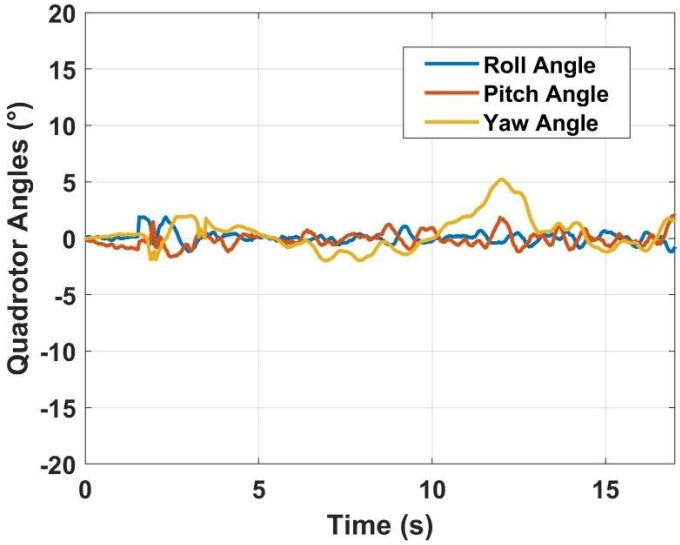


Figure 5.70: Real quadrotor roll, pitch and yaw angles response at hovering position

It is readily apparent from these responses that our design exhibits a noteworthy degree of accuracy. This assertion is substantiated by the observation that the quadrotor's attitude angle responses during hovering closely approximate the 0° reference value.

5.9 Conclusion

The experimental chapter in this study aimed to present the design, testing and results of a quadrotor system. To begin with, the quadrotor sizing was performed through stress and deformation analysis, and prediction parameters were determined using flight evaluation tools and a multi-copter calculator. This was followed by a comprehensive hardware and software selection, which included the design of the overall quadrotor system, hardware and wiring selection, flight control and sensors, actuators, power supply, communication devices, telemetry system and the global

wiring of all the devices. The hardware design was illustrated using images, and the quadrotor algorithm was explained in detail. In order to reduce the vibrations, the FFT technique was used. The experiment steps included balancing the propeller in static condition, balancing the motors in dynamic condition, and balancing the ensemble motor-propeller in dynamic condition by using the FFT tool. The results showed a significant reduction in the strength of vibrations for both balanced and imbalanced motors and propellers. The sensor calibration results showed improvements in orientation accuracy after calibrating the IMU and Compass. The IMU calibration results showed a reduction in the RMSE after calibrating the nine SEM parameters or the SEM matrix. The Compass calibration results showed a considerable improvement in orientation accuracy after calibration. The filtering results demonstrated the selection of filter cut-off frequencies for accelerometer and gyroscope, and the selection of the complementary filter coefficient. A comparison was made between raw and filtered measurements from the accelerometer and gyroscope under the complementary filter. The identification results demonstrated two techniques, the first being the Grey-Box Identification Results where the quadrotor moment of inertia and aerodynamic coefficients were extracted from experiments using test benches. The results were then compared with the Quadrotor inertia moments and drag and thrust coefficients from the experimental and optimization approach. The second approach was the Black-Box Identification Results, where a test bench and quadrotor controller structure were used to maintain the system stable during the identification process. The Excitation Input PRBS was chosen based on prior knowledge of quadrotor bandwidths, and the Order Selection and Input Delay were selected. The best model was determined through two main eliminator selection steps and later compared to extract the best model that well mimicked the quadrotor behaviour, which was later used in the controller design step. The identification results also demonstrated the use of Neuro-fuzzy, where the MF was shaped based on grid clustering. Finally, the control system results showed the response of the re-tuned PID controller and Full State Feedback controllers after tracking a variable square reference. The re-tuned PID controller demonstrated successful tracking, and the Full State Feedback controller also showed a successful response. The validation of the quadrotor attitude control design was carried out by maintaining the quadrotor in a hovering state. These results provide valuable insights into the design and operation of quadrotor systems and lay a strong foundation for future work in this field.

Obstacles in My PhD Research and Solutions

During the course of my PhD work, I encountered a multitude of obstacles that presented significant challenges. However, through diligent effort and perseverance, I was able to overcome these obstacles and achieve my research objectives. It is important to acknowledge and discuss these obstacles in my PhD thesis in a scientifically rigorous manner, in order to provide a comprehensive understanding of the difficulties encountered and the methods employed to overcome them. The use of precise language and a systematic approach will help to demonstrate the rigor of my research and the validity of my conclusions.

- One of the major hindrances encountered during my PhD work was the procurement of components required for the construction of a quadrotor system. Delays in delivery, which often lasted for several months, had a substantial impact on the progress of my research, as the availability of essential components was a crucial factor in the development of my experiments and the attainment of my objectives. Despite these challenges, I employed various strategies, such as seeking alternative sources and implementing contingency plans, to mitigate the adverse effects of the delayed deliveries and ensure that my research progress was not significantly impacted. These experiences highlight the importance of timely procurement and effective supply chain management in the successful execution of scientific research.
- The second substantial impediment encountered during the course of my PhD work was that, as usual in scientific experiment attempts, inherent risks and errors were inherent to the construction and experimentation of the quadrotor. Despite meticulous planning and execution, instances of loss, damage, or destruction of critical components, such as electronic parts, propellers, motors, ESCs, sensors, motherboards, and even the quadrotor itself, were observed during experimental activities. This resulted in the need for repair, reconstruction, or redevelopment of the aircraft, necessitating significant amounts of time, patience, and resources, including additional costs and expenses. These setbacks had a significant impact on the progress of the experimental activities and resulted in a delay in the completion of my PhD work within the desired timeframe. These experiences emphasize the crucial importance of considering and addressing the inherent risks in scientific experimentation, and the need for perseverance and adaptability in the face of obstacles.
- As is widely acknowledged, the quadrotor represents an inherently unstable system, much like other unstable systems, but its challenges lie in the complex

environment and conditions in which it operates. For instance, the two-wheeled inverted pendulum vehicle, a subject of my previous Master's degree project, also embodies an unstable system, albeit one that operates in a terrestrial environment. Any errors in programming or malfunction can result in the vehicle coming to a halt and falling to the ground, albeit with limited consequences, given the relatively small potential energy associated with its height. However, the situation is starkly different when it comes to quadrotors operating in an aerial environment. A small mistake can have catastrophic consequences, resulting in the complete loss of the quadrotor, which may have taken months to develop, as well as incurring substantial costs in terms of time, effort, and material. Moreover, the moral cost of such incidents cannot be overstated. Hence, it is imperative for developers in this field to consider these challenges and test their creations in a controlled environment that is specifically designed for the purpose. This is precisely the rationale behind the filing of our patent for a dedicated experiment platform “Universal Testbench for Multirotors”.

- The previously discussed controlled Test bench (previous point,), dedicated to the study and development of the quadrotor, serves as a valuable tool for assessing the functionality of the quadrotor. However, it is important to note that a more accurate measurement system is necessary to fully evaluate the accuracy and robustness of a proposed quadrotor controller. Specifically, in order to test the performance of a controller designed to control the quadrotor's angles while following a specific path, the use of an external and more precise measurement system, such as the Vicon or a similar motion tracking system, is recommended. This is because the use of these systems allows for a more accurate evaluation of the quality of the quadrotor controllers, rather than relying on the sensors within the aircraft.
- It is unfortunate that during the course of our study, access to the highly-accurate Test bench systems mentioned previously was not feasible. As a result, our evaluation of the proposed quadrotor controllers was limited to the measurements provided by the quadrotor's internal sensors. Nevertheless, it is imperative to understand that in order to invest in advanced controllers, such as those based on reinforcement learning, the availability of these Test bench platforms is essential. Without access to these platforms, the accuracy and robustness of the controllers cannot be fully evaluated, hindering progress in this field.

Conclusion

In conclusion, our research has comprehensively explored various facets of UAVs, focusing on quadrotors. This study was driven by the rapid evolution of UAV technology, which has ushered in a new era of possibilities across numerous applications, including surveillance, reconnaissance, mapping, and delivery services. This study outlines the key steps to produce a comprehensive and well-conceived project. It highlights the fundamental physical characteristics of the quadrotor frame and its various forms, the impact of size on velocity and power, the correlations between weight and wing size properties, and the optimal selection of motors, blades, and other quadrotor components. Additionally, this study critically analyses the most frequently used techniques for sizing UAVs, with a focus on the data-driven clustering approach for selecting quadrotor components based on design trends. The result of the sizing process involves selecting quadrotor components, including the frame, flight controller, sensors, and actuators. The hardware and software used in the quadrotor are subsequently described. The significance of sensor calibration and filtering in ensuring the accuracy of sensor data for effective aircraft control was underscored, which are crucial for effective aircraft control. The most important sensors, such as the IMU and the magnetometer, were calibrated, and a comprehensive discussion of calibration techniques and filtering methods was provided. This discussion also considered complementary filters and the frequency band selection based on the quadrotor's attitude bandwidth for high-pass and low-pass filters. Vibration, a significant threat to aircraft stability, was studied to determine scientifically sound and effective methods for its reduction. This was achieved by balancing each propeller motor, and the resulting balance was analysed using the FFT tool. Following a comprehensive review of existing techniques and technologies, the identification of quadrotor dynamics was extensively analysed. The coefficients of propulsion and moment of inertia of the drone were determined through estimation techniques, incorporating both input-output data and prior knowledge. The attitude model was then identified through the utilization of a black-box identification approach, employing models such as ARX, ARMAX, and ANFIS. The focus of these models was to facilitate the design of control systems for the quadrotor. The dynamic modelling and control techniques of the quadrotor were thoroughly investigated, resulting in the stabilization of its attitude through the use of cascade PID controllers, followed by a single PID and LQR controller, where their gains have been optimized using many

strategies, which were later experimentally applied in a quadrotor, resulting in improved behaviour. Subsequently, we conducted simulations and implemented an adaptive sliding mode controller augmented with a second-order sliding mode super-twist observer using MATLAB software. This approach yielded satisfactory results in the preparatory phase. The next step involves applying the SMC to a real quadrotor. This implementation is intended for future work. Finally, the validity of these findings was substantiated through simulations and experiments with a real quadrotor system. Following a comprehensive review of existing techniques and technologies, the identification of quadrotor dynamics was extensively analysed. The coefficients of propulsion and moment of inertia of the drone were determined through estimation techniques, incorporating input-output data and prior knowledge. The attitude model was then identified using a black-box identification approach, employing ARX, ARMAX, and ANFIS models. The focus of these models was to facilitate the design of control systems for the quadrotor. The dynamic modelling and control techniques of the quadrotor were thoroughly investigated, resulting in the stabilization of its attitude through the use of cascade PID controllers, followed by a single PID and Full states feedback controllers, where their gains have been optimized using many strategies, which were later experimentally applied in a quadrotor, resulting in improved behaviour. Subsequently, we conducted simulations and implemented an adaptive sliding mode controller augmented with a second-order sliding mode super-twist observer using MATLAB software. This approach yielded satisfactory results in the preparatory phase. The next step involves applying the SMC to a real quadrotor. This implementation is intended for future work. Finally, the validity of these findings was substantiated through simulations and experiments with a quadrotor system.

Perspective

The perspective of this PhD thesis highlights potential future directions for research and development related to the design and control of UAVs. Our research has contributed showing of a comprehensive engineering pathway for UAV design and control, construction of a reliable experimental Open-Source platform (quadrotor) for academic and other purposes, and development of a practical adaptive control strategy for real-world applications, taking into consideration various effects. However, the research still has some limitations, including the absence of test bench systems to assist

in evaluating proposed quadrotor controllers and the absence of a VICON system to aid in new controller suggestions or other developments. These limitations offer opportunities for further research and development in this area. One potential area for future research is the exploration of alternative test bench systems that could be used to evaluate the performance of the proposed quadrotor controllers. Additionally, the development of a VICON system that could aid in new controller suggestions and other developments would be an exciting area for further investigation. and other devices such as a camera, a hand, and so on. Following the thorough examination, numerous intriguing directions for further investigation and development are revealed. The initial focus should be on the application of SMC on the real quadrotors or any other robust controller including the heavy-computation one. This task requires dealing with practical issues caused by hardware constraints, environmental factors, and other considerations. A successful implementation will help to clearly show how useful and effective the proposed approaches are. A possible approach for improving the accuracy and depth of sensor data is the integration of advanced sensor technologies, such as vision sensors. To improve state observation, researchers should look into integrating cutting-edge sensors and fusing data from various sources. The quadrotor's vision and navigational abilities will be greatly enhanced by these developments, opening up new possibilities for autonomous flight. It is essential to broaden the study's focus to include algorithms for autonomous path planning and navigation. This extension ought to result in the creation of algorithms that enable quadrotors to independently explore complex and dynamic surroundings, skillfully avoid obstacles, and build the best flying trajectories. Applications in dynamic and uncertain environments require such capabilities. The implementation AI techniques offers a chance to improve the adaptability and decision-making abilities of the quadrotor. To improve control procedures, researchers should investigate the use of reinforcement learning and other AI techniques. Quadrotors can gain knowledge from their surroundings thanks to this integration, which also increases their adaptability and capacity for problem-solving. The robustness and fault tolerance of quadrotors should also be a priority. In order to ensure the safety and reliability of UAV operations, particularly in mission-critical environments, robust strategies must be devised to gracefully manage sensor failures, motor defects, or unforeseen events. This study's practical relevance goes beyond its local setting. The numerous applications of optimal control strategies and sensor calibration techniques across multiple domains should be explored and discussed by researchers. It is important to think about how these methods might be modified and

applied in the sectors of agriculture, search and rescue, surveillance, environmental monitoring, and other pertinent ones. Researchers should investigate how this research can result in more environmentally friendly UAV designs given the increasing emphasis on environmental factors. Examine the viability and advantages of electric or hybrid propulsion systems, taking into account how they would affect the total environmental impact of UAV operations as well as their effects on endurance and sustainability. Finally, it is crucial to promote cooperation and multidisciplinary research. To expand the research horizon, create creative solutions, and handle difficult challenges that cut across particular disciplines, promote collaboration with professionals from fields like computer science, materials engineering.

BIBLIOGRAPHY

[1] Kendoul, F. (2012). Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378.

[2] Lykou, G. Moustakas, D. Gritzalis, D. (2020). Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies. *Sensors*, 20(12),3537.

[3] Telli, K., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., & Mansoor, W. (2023). A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems*, 11(8), 400.

[4] Quan, Q. (2017). Introduction to multicopter design and control (1). *Singapore Springer* (pp. 150-160). https://doi.org/10.1007/978-981-10-3382-7

[5] Bhandari, A. Ahmad, F. Kumar, P. Patil, PP. (2019) Design and Vibration Characteristics Analysis of Quadcopter Body Frame. *International Journal of Applied Engineering Research,* 8(14),66–70.

[6] Hernandez, A., Copot, C., De Keyser, R., Vlas, T., & Nascu, I. (2013, Oct). Identification and path following control of an AR. Drone quadrotor. *In 2013 17th* IEEE *international conference on system theory, control and computing (ICSTCC),* 583-588.

[7] Stanculeanu, I., & Borangiu, T. (2011). Quadrotor black-box system identification. *World Academy of Science, Engineering and Technology*, 5, 276-279.

[8] Oliveira, E. C. L., Araujo, J. P. L., Silva Silveira, A., Silva, O. F., Vidal, J. F., França Silva, A. (2019 March). Quadrotor Black-Box System Identification Using Metaheuristics. *DINAME 2019, The XV International Symposium on Dynamic Problems of Mechanics, Buzios, RJ, Brazil.*

[9] Liu, Y. X., Li, X. S., Zhang, X. J., & Feng, Y. B. (2014). Novel calibration algorithm for a three-axis strapdown magnetometer. *Sensors*, 14(5), 8485-8504.

[10] Tedaldi, D., Pretto, A., & Menegatti, E. (2014, May). A robust and easy to implement method for IMU calibration without external equipments. *In 2014 IEEE International Conference on Robotics and Automation (ICRA)* 3042-3049.

[11] Rohac, J., Sipos, M., & Simanek, J. (2015). Calibration of low-cost triaxial inertial sensors. *IEEE Instrumentation & Measurement Magazine*, 18(6), 32-38.

[12] Han, B., Zhou, Y., Deveerasetty, K. K., Hu, C. (2018, August). A review of control algorithms for quadrotor. *In 2018 IEEE international conference on information and automation (ICIA)* 951-956.

[13] Chelihi, A., Perozzi, G., & Sentouh, C. (2021, June). Optimal direct adaptive model-free controller for twin rotor MIMO system using Legendre polynomials and PSO algorithm. *In Advances in Nonlinear Dynamics: Proceedings of the Second International Nonlinear Dynamics Conference (NODYCON 2021)*, (2),83-93.

[14] Guettal, L., Chelihi, A., Ajgou, R., & Touba, M. M. (2022). Robust tracking control for quadrotor with unknown nonlinear dynamics using adaptive neural network based fractional-order backstepping control. *Journal of the Franklin Institute*, 359(14), 7337-7364.

[15] Mokhtari, M. R., & Cherki, B. (2015). A new robust control for minirotorcraft unmanned aerial vehicles. *ISA transactions*, 56, 86-101.

[16] Nadour, M., Boumehraz, M., Cherroun, L., & Puig Cayuela, V. (2019). Hybrid type-2 fuzzy logic obstacle avoidance system based on horn-schunck method. *Electrotehnica, Electronica, Automatica,* 67(3), 45-51.

[17] Guettal, L., Chelihi, A., Touba, M. M., Glida, H. E., & Ajgou, R. (2020, May). Adaptive Fuzzy-Chebyshev Network-based Continuous Sliding Mode Controller for Quadrotor Unmanned Aerial Vehicle. *In 2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP)*, 382-387.

[18] Bartolini, G., Pisano, A., Punta, E., & Usai, E. (2003). A survey of applications of second-order sliding mode control to mechanical systems. International Journal of control, 76(9-10), 875-892.

[19] Chamola, V., Kotesh, P., Agarwal, A., Gupta, N., & Guizani, M. (2021). A comprehensive review of unmanned aerial vehicle attacks and neutralization techniques. *Ad hoc networks*, 111, 102-324.

[20] Clothier, R., Palmer, J., Walker, R., & Fulton, N. (2010). Definition of airworthiness categories for civil Unmanned Aircraft Systems (UAS). *The 27th Congress of the International Council of the Aeronautical Sciences (ICAS), 1-12.*

[21] Hassanalian, M., & Abdelkefi, A. (2017). Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91, 99-131.

[22] Kim, M., Kim, J., Byun, Y., Kim, J., & Kang, B. (2016). Study on Analysis of Vibration Characteristics and Modal Test for a Quad-Rotor Drone. *Journal of the Korean Society for Precision Engineering*, 33(9), 707-714.

[23] Phang, S. K., Li, K., Yu, K. H., Chen, B. M., Lee, T. H. (2014). Systematic design and implementation of a micro unmanned quadrotor system. *Unmanned Systems*, 2(02), 121-141.

[24] Kuantama, E., Craciun, D., Tarca, R. (2016). Quadcopter body frame model and analysis. *Annals Of the University of Oradea*, 71-74.

[25] Niemiec, R., Gandhi, F. (2016). A comparison between quadrotor flight configurations. *42nd European Rotorcraft Forum, Lille, France, Lateral,* 1000, 2

[26] Bohorquez, F. (2007). Rotor Hover Performance and System Design of An Efficient Coaxial Rotary Wing Micro Air Vehicle, *PhD thesis. University of Maryland, College Park, Department of Aerospace Engineering*.

[27] Markus, M. (March2020). eCalc-Reliable Electric Drive Simulations. https://www.ecalc.ch/

[28] Christian, P. (March2020). Drive calculator (v3.4). http://www.drivecalc.de, March 2020

[29] Bershadsky, D., Haviland, S., Johnson, E. N. (2016). Electric multirotor UAV propulsion system sizing for performance prediction and design optimization. *In 57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 0581.

[30] Ampatis, C., Papadopoulos, E. (2014). Parametric design and optimization of multi-rotor aerial vehicles. *Applications of Mathematics and Informatics in Science and Engineering*, 1-25.

[31] Gur, O., Rosen, A. (2009). Optimizing electric propulsion systems for unmanned aerial vehicles. *Journal of aircraft,* 46(4), 1340-1353.

[32] Rand, O., Khromov, V. (2021). Mission Oriented Multi-Prop UAV Analysis Using Statistical Design Trends. *Aerospace*, 8(11), 321.

[33] Winslow, J., Hrishikeshavan, V., & Chopra, I. (2018). Design methodology for small-scale unmanned quadrotors. *Journal of Aircraft*, 55(3), 1062-1070.

[34] Gatti, Mauro, and Fabrizio Giulietti. "Preliminary design analysis methodology for electric multirotor." IFAC Proceedings 46(30), 58-63.

[35] Albéri, M., Baldoncini, M., Bottardi, C., Chiarelli, E., Fiorentini, G., Raptis, K. G. C., Mantovani, F. (2017). Accuracy of flight altitude measured with low-cost GNSS, radar and barometer sensors: Implications for airborne radiometric surveys. *Sensors,* 17(8), 1889.

[36] Kim, M., Joo, H., Jang, B. (2017, July). Conceptual multicopter sizing and performance analysis via component database. *In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN),* 105-109.

[37] Dai, X., Quan, Q., Ren, J., Cai, K. Y. (2019). An analytical design-optimization method for electric propulsion systems of multicopter UAVs with desired hovering endurance. *IEEE/ASME Transactions on Mechatronics*, 24(1), 228-239.

[38] Quan, B. Dai, X. (March2020). flight evaluation calculator. https://www.flyeval.com/recalc.html.

[39] Biczyski, M., Sehab, R., Whidborne, J. F., Krebs, G., & Luk, P. (2020). Multirotor sizing methodology with flight time estimation. *Journal of Advanced Transportation, 1(2), 3.*

[40] Nicoud, J. D., Zufferey, J. C. (2002, September). Toward indoor flying robots. *In IEEE/RSJ International Conference on Intelligent Robots and Systems* (1), 787-792.

[41] Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE robotics & automation magazine*, 19(3), 20-32.

[42] Kumar, S., & Mishra, P. C. (2016). Finite element modeling for structural strength of quadcoptor type multi mode vehicle. *Aerospace Science and Technology* (53), 252-266.

[43] Bhatia, V., Karthikeyan, R., Ganesh Ram, R. K., & Cooper, Y. N. (2014). Design optimisation and analysis of a quadrotor arm using finite element method. *Applied Mechanics and Materials,* (664), 371-375.

[44] Anudeep, M., Diwakar, G., & Katukam, R. (2014). Design of a quad copter and fabrication. *International Journal of Innovations in Engineering and Technology*, 4(1), 59-65.

[45] Zhang, Q., Chen, J., Yang, L., Dong, W., Sheng, X., & Zhu, X. (2015). Structure optimization and implementation of a lightweight sandwiched quadcopter. *In Intelligent Robotics and Applications: 9th International Conference, ICIRA* (3), 220-229.

[46] Derrouaoui, S. H., Bouzid, Y., & Guiatni, M. (2022). Adaptive integral backstepping control of a reconfigurable quadrotor with variable parameters' estimation. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* 236(7), 1294-1309.

[47] Kekec, T., Ustundag, B. C., Guney, M. A., Yildirim, A., & Unel, M. (2013, November). A modular software architecture for UAVs. *In IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society* 4037-4042.

[48] Goswami, M., Kumar, A., & Chavan, P. (2021). Development of an Autonomous Quadrotor with a Robust Real-time Control, *A Review on Modeling, Estimation, Control, and Hardware Aspects*.

[49] Spica, R., Giordano, P. R., Ryll, M., Bülthoff, H. H., & Franchi, A. (2013). An open-source hardware/software architecture for quadrotor UAVs. *IFAC Proceedings Volumes,* 46(30), 198-205.

[50] Lorenz, M., Daniel, A., Beat, K., Mathieu, B., Paul, R., (2022 June). PX4 Drone Autopilot User Guide (v1.14.0). docs.px4.io/main/en/concept/px4_systems_architecture.html.

[51] Meier, L., Tanskanen, P., Heng, L., Lee, G. H., Fraundorfer, F., & Pollefeys, M. (2012). PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots,* 33, 21-39.

[52] López, J., Royo, P., Pastor, E., Barrado, C., & Santamaria, E. (2007, November). A middleware architecture for unmanned aircraft avionics. *In Proceedings of the 2007 ACM/IFIP/USENIX international conference on Middleware companion,* 1-6.

[53] Honvault, C., Le Roy, M., Gula, P., Fabre, J. C., Le Lann, G., & Bornschlegl, E. (2005). Novel generic middleware building blocks for dependable modular avionics systems. *In Dependable Computing-EDCC 5: 5th European Dependable Computing Conference* (5), 140-153.

[54] Maza, I., Kondak, K., Bernard, M., & Ollero, A. (2010). Multi-UAV cooperation and control for load transportation and deployment. *In Selected papers from the 2nd International Symposium on UAVs,* 417-449.

[55] Yedamale, P. (2003). Brushless DC (BLDC) motor fundamentals. *Microchip Technology Inc*, 20(1), 3-15.

[56] Infineon XC800 Family (2009), AP08091 Sensorless Control of Brushless DC Motor using Infineon XC864 Microcontroller", *Infineon Technologies AG*.

[57] Rambabu, S. (2007). Modeling and control of a brushless DC motor. *Master of Thesis In Power Control and Drives Technology, National Institute of Technology Rourkela*.

[58] Wiberg, J. (2003 December). Controlling a brushless DC motor in a shift-by-wire system, Institutionen för systemteknik, (3517), 47.

[59] Gill, R., & D'Andrea, R. (2019). Computationally efficient force and moment models for propellers in UAV forward flight applications. *Drones*, 3(4), 77.

[60] Mogensen, K. N. (2016). Motor-control considerations for electronic speed control in drones. *Analog Applications Journal Texas Instruments*.

[61] Alexanderson, E. F. W., Edwards, M. A., & Willis, C. H. (1938). Electronic speed control of motors. *IEEE Electrical Engineering*, 57(6), 343-354.

[62] Boualem, S., Kraa, O., Benmeddour, M., Kermadi, M., Maamir, M., & Cherif, H. (2022). Power management strategy based on Elman neural network for grid-connected photovoltaic-wind-battery hybrid system. *Computers and Electrical Engineering*, (99), 107823.

[63] Dell, R., & Rand, D. A. J. (2001). Understanding batteries. *Cambridge : Royal Society of Chemistry, c2001* (1st). ISBN : 1-84755-222-6.

[64] Fuller, N. W., Diane, T., Zhaodan K., Understanding bat flight as a model for bio-inspired aircraft designs. Integrative and comparative biology, (55).

[65] Ebeid, E., Skriver, M., & Jin, J. (2017, August). A survey on open-source flight control platforms of unmanned aerial vehicle. *In 2017 Euromicro Conference on digital system design (dsd),* 396-402.

[66] Lim, H., Park, J., Lee, D., & Kim, H. J. (2012). Build your own quadrotor: Open-source projects on unmanned aerial vehicles. *IEEE Robotics & Automation Magazine*, 19(3), 33-45.

[67] Spica, R., Giordano, P. R., Ryll, M., Bülthoff, H. H., & Franchi, A. (2013). An open-source hardware/software architecture for quadrotor UAVs. *IFAC Proceedings Volumes*, 46(30), 198-205.

[68] Kangunde, V., Jamisola, R. S., & Theophilus, E. K. (2021). A review on drones controlled in real-time. *International journal of dynamics and control (9)*, 1-15.

[69] Poddar, S., Kumar, V., & Kumar, A. (2017). A comprehensive overview of inertial sensor calibration techniques. *Journal of Dynamic Systems, Measurement, and Control*, 139(1), 011006.

[70] Farrell, J. A., Silva, F. O., Rahman, F., & Wendel, J. (2021). *Inertial Measurement Unit Error Modeling Tutorial: Inertial Navigation System State Estimation with Real-Time Sensor Calibration.* IEEE Control Systems Magazine 6(42), 40-66.

[71] Ru, X., Gu, N., Shang, H., & Zhang, H. (2022). MEMS inertial sensor calibration technology: Current status and future trends. *Micromachines*, 13(6), 879.

[72] Shin, E. H., & El-Sheimy, N. (2002). A new calibration method for strapdown inertial navigation systems Z. *Vermess*, 127(1), 1-10.

[73] Aydemir, G. A., & Saranlı, A. (2012). Characterization and calibration of MEMS inertial sensors for state and parameter estimation applications. *Measurement*, 45(5), 1210-1225.

[74] Li, J., Jiao, F., Fang, J., & Ma, Y. (2013). Integrated calibration method for dithered RLG POS using a hybrid analytic/Kalman filter approach. *IEEE Transactions on Instrumentation and Measurement*, 62(12), 3333-3342.

[75] Yadav, N., & Bleakley, C. (2016). Fast calibration of a 9-DOF IMU using a 3 DOF position tracker and a semi-random motion sequence. *Measurement*, 90, 192-198.

[76] Zhao, G., Yang, Q., & Song, L. I. (2014). Ten-position Systematic Calibration Method for FOG Strap-down Inertial Navigation System. *Journal of Projectiles Rockets Missiles & Guidance,* (6), 8-12.

[77] Bonin-Font, F., Massot-Campos, M., Negre-Carrasco, P. L., Oliver-Codina, G., & Beltran, J. P. (2015). Inertial sensor self-calibration in a visually-aided navigation approach for a micro-AUV. *Sensors*, 15(1), 1825-1860.

[78] Kim, A., & Golnaraghi, M. (2004). Initial calibration of an inertial measurement unit using an optical position tracking system. *In PLANS 2004. Position Location and Navigation Symposium,* (IEEE Cat. No. 04CH37556) ,96-101.

[79] Frosio, I., Pedersini, F., & Borghese, N. A. (2008). Autocalibration of MEMS accelerometers. *IEEE Transactions on Instrumentation and Measurement*, 58(6), 2034-2041.

[80] Bonnet, S., Bassompierre, C., Godin, C., Lesecq, S., & Barraud, A. (2009). Calibration methods for inertial and magnetic sensors. *Sensors and Actuators A: Physical*, 156(2), 302-311.

[81] Šipoš, M., Roháč, J., & Nováček, P. (2012). Improvement of electronic compass accuracy based on magnetometer and accelerometer calibration. *Acta Physica Polonica* A, 121(4), 945-949.

[82] Sipos, M., Paces, P., Rohac, J., & Novacek, P. (2011). Analyses of triaxial accelerometer calibration algorithms. *IEEE Sensors Journal*, 12(5), 1157-1165.

[83] Caruso, M. J. (1997). Application of magnetoresistive sensors in navigation systems. *In Sensors and Actuators,* SAE SP-1220.

[84] Gebre-Egziabher, D., Elkaim, G. H., Powell, J. D., & Parkinson, B. W. (2006). Calibration of strapdown magnetometers in the magnetic field domain. *Journal of aerospace engineering*, 19(2), 87-102.

[85] Vasconcelos, J., Elkaim, G., Silvestre, C., Oliveira, P., & Cardeira, B. (2008). A geometric approach to strapdown magnetometer calibration in the sensor frame. *IFAC Proceedings Volumes*, 41(1), 172-177.

[86] Bowditch, N. (2018). The American Practical Navigator: An Epitome of Navigation, *California State University Maritime Academy (Navigation - United States)* (2).

[87] Renaudin, V., Afzal, M. H., & Lachapelle, G. (2010). Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors*, (Vol 2010).

[88] Alken, P., et al. (2021). International Geomagnetic Reference Field: The Thirteenth Generation. *Earth, Planets and Space*, 73(1), 1-25.

[89] Markovsky, I., Kukush, A., & Van Huffel, S. (2004). Consistent least squares fitting of ellipsoids. *Numerische Mathematik*, 98(1), 177-194.

[90] Balamurugan, G., Valarmathi, J., & Naidu, V. P. S. (2016). Survey on UAV navigation in GPS denied environments. *In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, 198-204.

[91] Gowda, M., Manweiler, J., Dhekne, A., Choudhury, R. R., & Weisz, J. D. (2016, October). Tracking drone orientation with multiple GPS receivers. *In Proceedings of the 22nd annual international conference on mobile computing and networking*, 280-293.

[92] Crassidis, J. L., Markley, F. L., & Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of guidance, control, and dynamics*, 30(1), 12-28.

[93] Lu, Y., Xue, Z., Xia, G. S., & Zhang, L. (2018). A survey on vision-based UAV navigation. *Geo-spatial information science*, 21(1), 21-32.

[94] Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). Estimation with applications to tracking and navigation: theory algorithms and software. *John Wiley & Sons*.

[95] Arasaratnam, I., & Haykin, S. (2009). Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6), 1254-1269.

[96] Ristic, B., Arulampalam, S., & Gordon, N. (2003). Beyond the Kalman filter: Particle filters for tracking applications. *Artech house*.

[97] Julier, S. J., & Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. Proceedings of the IEEE, 92(3), 401-422.

[98] Hosseinyalamdary, S. (2018). Deep Kalman filter: Simultaneous multi-sensor integration and modelling; A GNSS/IMU case study. *Sensors,* 18(5), 1316.

[99] Wang, S., & Yang, Y. (2012, July). Quadrotor aircraft attitude estimation and control based on kalman filter. *In Proceedings of the 31st Chinese Control Conference*, 5634-5639.

[100] Xiong, J. J., & Zheng, E. H. (2015). Optimal kalman filter for state estimation of a quadrotor UAV. *Optik,* 126(21), 2862-2868.

[101] You, W., Li, F., Liao, L., & Huang, M. (2020). Data fusion of UWB and IMU based on unscented Kalman filter for indoor localization of quadrotor UAV. *IEEE Access*, 8, 64971-64981.

[102] Benzerrouk, H., Nebylov, A., & Salhi, H. (2016). Quadrotor UAV state estimation based on High-Degree Cubature Kalman filter. *IFAC-PapersOnLine*, 49(17), 349-354.

[103] Kaba, A., & Ermeydan, A. (2022). Improved particle filter-based estimation of a quadrotor subjected to uncertainties. Aircraft Engineering and Aerospace Technology, 94(7), 1144-1156.

[104] Perez Paina, G., Gaydou, D., Redolfi, J. A., Paz, C., & Canali, L. (2011). Experimental comparison of Kalman and complementary filter for attitude estimation. *In XII Argentine Symposium on Technology (AST), (2011)*, 205-215

[105] Roberts, J. M., Corke, P. I., & Buskey, G. (2003). Low-cost flight control system for a small autonomous helicopter. *In 2003 IEEE International Conference on Robotics and Automation*, (1), 546-551.

[106] Mahony, R., Hamel, T., & Pflimlin, J.-M. (2005). Complementary filter design on the special orthogonal group SO (3). *In Proceedings of the 44th IEEE Conference on Decision and Control*, 1477-1484.

[107] Baerveldt, A.-J., & Klang, R. (1997). A low-cost and low-weight attitude estimation system for an autonomous helicopter. *In Proceedings of IEEE International Conference on Intelligent Engineering Systems,* 391-395.

[108] Narkhede, P., Poddar, S., Walambe, R., Ghinea, G., & Kotecha, K. (2021). Cascaded complementary filter architecture for sensor fusion in attitude estimation. *Sensors*, 21(6), 1937.

[109] Salwa, M., & Krzysztofik, I. (2022). Application of Filters to Improve Flight Stability of Rotary Unmanned Aerial Objects. *Sensors*, 22(4), 1677.

[110] Kuantama, E., Moldovan, O. G., Ţarcă, I., Vesselényi, T., & Ţarcă, R. (2021). Analysis of quadcopter propeller vibration based on laser vibrometer. *Journal of Low Frequency Noise, Vibration and Active Control*, 40(1), 239-251.

[111] Drozdetskaya, O., & Fidlin, A. (2016). On the dynamic balancing of a planetary moving rotor using a passive pendulum-type device. *Procedia IUTAM*, 19, 126-135.

[112] Wang, C. C., & Yau, H. T. (2008). Application of a hybrid numerical method to the bifurcation analysis of a rigid rotor supported by a spherical gas journal bearing system. *Nonlinear Dynamics*, 51, 515-528.

[113] Yau, H. T., & Yan, J. J. (2009). Adaptive sliding mode control of a high-precision ball-screw-driven stage. *Nonlinear Anal Real World Appl*, 10, 1480-1489.

[114] Wang, C. C., Yau, H. T., Jang, M. J., & Yeh, Y. L. (2007). Theoretical analysis of the non-linear behavior of a flexible rotor supported by herringbone grooved gas journal bearings. *Tribology international*, 40(3), 533-541.

[115] Silvestre, M. A. R., Morgado, J., Alves, P., Santos, P., Gamboa, P., & Páscoa, J. C. (2015). Propeller performance measurements at low Reynolds numbers. *International Journal of Mechanics*, 9, 154-166.

[116] Zhang, X., Li, X., Wang, K., & Lu, Y. (2014). A Survey of Modelling and Identification of Quadrotor Robot. *Abstract and Applied Analysis (2016)*, 1-16.

[117] Shraim, H., Awada, A., & Youness, R. (2018). A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerospace and Electronic Systems Magazine*, 33(7), 14-33.

[118] Legowo, A., Sulaeman, E., & Rosli, D. (2019). Review on System Identification for Quadrotor Unmanned Aerial Vehicle (UAV). *In 2019 Advances in Science and Engineering Technology International Conferences (ASET)* , 1-8.

[119] Ljung, L. (2014). Perspectives on System Identification. *Journal of Intelligent and Robotic Systems*, 74, 129-145.

[120] Gevers, M. (2006). A personal view of the development of system identification. *IEEE Control Systems Magazine*, 26(6), 93–105.

[121] Bergamasco, M., & Lovera, M. (2014). Identification of Linear Models for the Dynamics of a Hovering Quadrotor. *IEEE Transactions on Control Systems Technology*, 22(5), 1696-1707.

[122] Tischler, M., & Remple, R. (2012). Aircraft and rotorcraft system identification Engineering Methods with Flight-Test Examples. *Reston: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/4.868207*

[123] Rabehi, R., Kouzou, A., Saadi, S., & Hafaifa, A. (2019). Parameter selection criteria of Prony method for accurate harmonics and inter-harmonic component identification. *Electrotehnica, Electronica, Automatica (EEA),* 67(1), 46-53.

[124] Pounds, P., Mahony, R., & Corke, P. (2010). Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7), 691-699.

[125] Zawiski, R., & Blachuta, M. (2012, August). Dynamics and optimal control of quadrotor platform. In AIAA Guidance, Navigation, and Control Conference, 4915.

[126] Mokhtari, M. R., & Braham, A. Ch. (2016). Disturbance Observer-Based Approximate Linearization Control of Gun Launched MAV. *Electrotehnica, Electronica, Automatica (EEA),* 64(1), 133-142.

[127] Amir, M. Y., & Abbass, V. (2008). Modelling of Quadrotor Helicopter Dynamics. *International Conference on Smart Manufacturing Application*, 100-105.

[128] Bouabdallah, S., Murrieri, P., & Siegwart, R. (2005). Towards Autonomous Indoor Micro VTOL. *Autonomous Robots*, 18(2), 171-183.

[129] Naidoo, Y., Stopforth, R., & Bright, G. (2011). Quad-Rotor Unmanned Aerial Vehicle Helicopter Modelling & Control. *International Journal of Advanced Robotic Systems*, 8(4), 139-149.

[130] Falkenberg, O., Witt, J., Pilz, U., & Werner, H. (2012). Model Identification and H∞ Attitude Control for Quadrotor MAV's. *Intelligent Robotics and Applications*, 460-471.

[131] Panizza, P., Riccardi, F., & Lovera, M. (2015). Black-box and grey-box identification of the attitude dynamics for a variable-pitch quadrotor. *IFAC-Papers online,* 48(9), 61-66.

[132] Li, Q. (2014). Grey-box system identification of a quadrotor unmanned aerial vehicle (Doctoral dissertation, Delft University of Technology).

[133] Sun, S., Schilder, R., & de Visser, C. C. (2018). Identification of quadrotor aerodynamic model from high speed flight data. *In 2018 AIAA Atmospheric Flight Mechanics Conference*, 0523.

[134] Hernandez, A., Copot, C., Keyser, R., Vlas, T., & Nascu, I. (2013). Identification and path following control of an AR. Drone quadrotor. *17th International Conference on System Theory*, 583-588.

[135] Stanculeanu, I., & Borangiu, T. (2011). Quadrotor black-box system identification. *World Academy of Science, Engineering and Technology*, 5, 276-279.

[136] Jithu, G., & Jayasree, P. R. (2016, March). Quadrotor modelling and control. In 2016 *International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT),* 1167-1172.

[137] Telli, K., & Boumahrez, M. (2022). Black-Box System Identification for Low-Cost Quadrotor Attitude at Hovering. *Electrotehnica, Electronica, Automatica (EEA),* 70(4), 88-97.

[138] Telli, K., & Boumahrez, M. (2023). Quadcopter Control Based on Experimental Dynamic Identification, the First International Conference on Advances in Electrical and Computer Engineering (ICAECE'2023) AIJR Abstracts.

[139] Sarioğlu, A., & Kural, A. (2015). Modelling and ARX identification of a quadrotor Mini UAV. *9th International Conference on Electrical and Electronics Engineering (ELECO)*, 1196-1200.

[140] Ireland, M., & Anderson, D. (2015). System Identification of multi-rotor UAVs using echo state networks. In AUVSI's Unmanned Systems 2015, Atlanta, GA, USA (2015).

[141] Ljung, L. (1998). System identification. *In Signal analysis and prediction, Boston, MA: Birkhäuser Boston,* 163-173.

[142] Hamel, P. G., & Kaletka, J. (1997). Advances in rotorcraft system identification. *Progress in Aerospace Sciences,* 33(3-4), 259-284.

[143] Hoffer, N. V., Coopmans, C., Jensen, A. M., & Chen, Y. (2014). A survey and categorization of small low-cost unmanned aerial vehicle system identification. *Journal of Intelligent & Robotic Systems*, 74, 129-145.

[144] Hoffer, N. V., Coopmans, C., Jensen, A. M., & Chen, Y. (2013). A Survey and Categorization of Small Low-Cost Unmanned Aerial Vehicle System Identification. *Journal of Intelligent and Robotic Systems*, 74(1–2), 129–145.

[145] Van, D. H., & Schrama, R. J. P. (1995). Identification and control — *Closed-loop issues. Automatica,* 31(12), 1751–1770.

[146] Karimi, A., & Dore, L. I. (1998). Comparison of the closed-loop identification methods in terms of the bias distribution. *Systems & Control Letters*, 34(4), 159–167.

[147] Wahlberg, B., Hjalmarsson, H., & Barenthin, M. (2006). On optimal input design in system identification. *IFAC Proceedings Volumes*, 39(1), 499–504.

[148] Bombois, X., & Gilson, M. (2006). Cheapest identification experiment with guaranteed accuracy in the presence of undermodeling. *IFAC Proceedings Volumes*, 39(1), 505–510.

[149] Bo, L. I. U., Jun, Z. H. A. O., & Jixin, Q. I. A. N. (2006). Design and Analysis of Test Signals for System Identification. *In Computational Science – ICCS 2006,* 593–600.

[150] Nelles, Oliver. (2020). Nonlinear system identification: from classical approaches to neural networks, fuzzy models, and Gaussian processes. *Springer Nature,* (2). https://doi.org/10.1007/978-3-030-47439-3

[151] Aihui, T., & Godfrey, K. R. (2002). The generation of binary and near-binary pseudorandom signals: an overview. *IEEE Transactions on Instrumentation and Measurement,* 51(4), 583-588.

[152] Wei, W., Schwartz, N., & Cohen, K. (2014). Frequency-domain system identification and simulation of a quadrotor controller. In AIAA Modeling and Simulation Technologies Conference, 1342.

[153] Amrani, M. E. H., Dowdeswell, R. M., Payne, P. A., & Persaud, K. C. (1998). Pseudo-random binary sequence interrogation technique for gas sensors. *Sensors and Actuators B: Chemical*, 47(1-3), 118-124.

[154] Forssell, U., & Ljung, L. (n.d.). Identification for control: some results on optimal experiment design. *Proceedings of the 37th IEEE Conference on Decision and Control* (Cat. No.98CH36171), *37th IEEE Conference on Decision and Control*, 3384-3389.

[155] Ghazbi, S. N., Aghli, Y., Alimohammadi, M., & Akbari, A. A. (2016). Quadrotors unmanned aerial vehicles: A review. *International Journal on Smart Sensing and Intelligent Systems,* 9(1), 309–333.

[156] Kim, J., Kang, M.-S., & Park, S. (2009). Accurate modeling and robust hovering control for a quad-rotor VTOL aircraft. *Selected papers from the 2nd International Symposium on UAVs*, 9–26.

[157] Zhang, X., Li, X., Wang, K., & Lu, Y. (2014, October). A survey of modelling and identification of quadrotor robot. *In Abstract and Applied Analysis* (Vol.2014). Hindawi.

[158] Domingues, J. M. B. (2009). Quadrotor prototype. *Instituto superior técnico, Portugal, October (2009)*.

[159] Tayebi, A., & McGilvray, S. (2006). Attitude stabilization of a VTOL quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3), 562–571.

[160] Aboytes Reséndiz, V. M., & Rivas Araiza, E. (2016). System Identification of a Quad-rotor in X Configuration from Experimental Data. *Research in Computing Science*, 118(1), 77–86.

[161] Partovi, A. R., Zong Yao Kevin, A., Lin, H., Chen, B., & Cai, G. (2012, August). Development of a cross style quadrotor. *In AIAA Guidance, Navigation, and Control Conference*, 4780.

[162] Jee, S. H., Cho, H. C., & Kim, J. (2020). The Experimental Modeling of Quad-Rotor Actuators with Undefined Hardware Errors for Safety-Flight. *Electronics*, 9(4), 579.

[163] Lu, S. (2018). Modeling, control and design of a quadrotor platform for indoor environments. *Master Thesis, Arizona State University, USA*. https://core.ac.uk/download/pdf/195380024.pdf

[164] B Samir, B. (2007). Design and control of quadrotors with application to autonomous flying. *Doctoral dissertation No 372 , Ecole Polytechnic Fédérale De Lausanne*. https://doi.org/10.5075/epfl-thesis-3727

[165] Hasseni, S. E. I. (2020). Commande Robuste Non-linéaire d'un Quadrotor, *Doctoral dissertation, Université Mohamed Khider–Biskra.*

[166] Ajmera, Juhi, and V. Sankaranarayanan. (2016). Point-to-point control of a quadrotor: Theory and experiment. *IFAC-Papers On Line*, 49(1), 401-406.

[167] Sabatino, F. (2015). Quadrotor control: modeling, nonlinear control design, and simulation, Master Thesis, KTH, School of Electrical Engineering (EES), Automatic Control .

[168] Ataka, A., Tnunay, H., Inovan, R., Abdurrohman, M., Preastianto, H., Cahyadi, A. I., & Yamamoto, Y. (2013, November). Controllability and observability analysis of the gain scheduling-based linearization for UAV quadrotor. *In 2013* IEEE *International conference on robotics, biomimetics, intelligent computational systems*, 212-218.

[169] Ordaz, P., et al. (2018). Nonlinear Control and Trajectory Tracking of an Unmanned Aircraft System Based on a Complete State Space Representation. *IFAC-PapersOnLine*, 51(13), 561-566.

[170] Okyere, Emmanuel, et al. (2019). LQR controller design for quad-rotor helicopters. *The Journal of Engineering*, 2019.17, 4003-4007.

[171] Amin, Roohul, Li Aijun, and Shahaboddin Shamshirband. (2016). A review of quadrotor UAV: control methodologies and performance evaluation. *International Journal of Automation and Control*, 10(2), 87-103.

[172] Liu, P., Chen, A. Y., Huang, Y. N., Han, J. Y., Lai, J. S., Kang, S. C., Tsai, M. H. (2014). A review of rotorcraft unmanned aerial vehicle (UAV) developments and applications in civil engineering. *Smart Struct System*, 13(6), 1065-1094.

[173] Nguyen, Hoa T., et al. (2020). Control algorithms for UAVs: A comprehensive survey. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 7(23), e5-e5.

[174] Gutierrez Soto, Mariantonieta, and Hojjat Adeli. (2017). Recent advances in control algorithms for smart structures and machines. *Expert Systems*, 34(2), e12205.

[175] Khalil, K. H. (1996). Nonlinear Systems,"2nd Edition, Englewood Cliffs, NJ, Prentice-Hall.

[176] El-Sharif, Ibrahim A., Fathi O. Hareb, and Amer R. Zerek. (2014). Design of discrete-time PID controller. *International Conference on Control, Engineering & Information Technology (CEIT'14)*, 110-115.

[177] Jatsun, S., Emelyanova, O., Leon, A.S.M., Stykanyova, S. (2017). Control flight of a UAV type tricopter with a fuzzy logic controller. *In Proceedings of the 2017 Dynamics of Systems, Mechanisms and Machines (Dynamics),* 1–5.

[178] Dong, J., He, B. (2019). Novel fuzzy PID-type iterative learning control for quadrotor UAV. *Sensors*, 19, 24.

[179] Andrade, F.A., Guedes, I.P., Carvalho, G.F., Zachi, A.R., Haddad, D.B., Almeida, L.F., de Melo, A.G., Pinto, M.F. (2022). Unmanned Aerial Vehicles Motion Control with Fuzzy Tuning of Cascaded-PID Gains. *Machines*, 10, 12.

[180] Melo, A. G., Andrade, F. A., Guedes, I. P., Carvalho, G. F., Zachi, A. R., & Pinto, M. F. (2022). Fuzzy gain-scheduling PID for UAV position and altitude controllers. Sensors, 22(6), 2173.

[181] Rodríguez-Abreo, O., Rodríguez-Reséndiz, J., Fuentes-Silva, C., Hernández-Alvarado, R., Falcón, M.D.C.P.T. (2021). Self-tuning neural network PID with dynamic response control. *IEEE Access,* 9, 65206–65215.

[182] Sawyer, S. (2015). Gain-scheduled control of a quadcopter UAV. *Master's thesis, University of Waterloo*. *UWSpace* ,http://hdl.handle.net/10012/9488

[183] Abdollahi, A., Tabar, A. F., & Khodadadi, H. (2015). Optimal controller design for quadrotor by genetic algorithm with the aim of optimizing the response and control input signals. *Cumhuriyet Üniversitesi Fen Edebiyat Fakültesi Fen Bilimleri Dergisi*, 36(3), 135-147.

[184] Rodríguez-Abreo, Omar, et al. (2020). Genetic algorithm-based tuning of backstepping controller for a quadrotor-type unmanned aerial vehicle. *Electronics*, 9(10), 1735.

[185] Hasseni, Seif-El-Islam, Latifa Abdou, and Hossam-Eddine Glida. (2021). Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms. *EvolutionaryIntelligence*, 14, 61-73.

[186] Cowling, I., Yakimenko, O., Whidborne, J., Cooke, A. (2010). Direct method-based control system for an autonomous quadrotor. *Journal of Intelligent & Robotic Systems*, 60(2) ,285-316.

[187] Valenti, M., Bethke, B., Fiore, G., How, J., & Feron, E. (2006, August). Indoor multi-vehicle flight testbed for fault detection, isolation, and recovery. *In AIAA guidance ,navigation, and control conference and exhibit*, 6200.

[188] Heng, X., Cabecinhas, D., Cunha, R., Silvestre, C., & Qingsong, X. (2015, November). A trajectory tracking LQR controller for a quadrotor: Design and experimental evaluation. *InTENCON 2015 IEEE region 10 conference*, 1-7.

[189] Liu, C., Pan, J., & Chang, Y. (2016, July). PID and LQR trajectory tracking control for an unmanned quadrotor helicopter: Experimental studies. *In 2016 35th Chinese IEEE controlconference (CCC)*, 10845-10850.

[190] Ghaffar, A. A., & Richardson, T. (2015). Model reference adaptive control and LQR control for quadrotor with parametric uncertainties. *International Journal of Mechanical andMechatronics Engineering*, 9(2), 244-250.

[191] Raffo, G. V., Ortega, M. G., Rubio, F. R. (2010). An integral predictive nonlinear control structure for a quadrotor helicopter. *Automatica*, 46(1), 29-39.

[192] Lee, D., Kim, H. J., Sastry, S. (2009). Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter. *International Journal of Control, Automation, andSystems,* 7(3), 419-428.

[193] Rashad, R., Aboudonia, A., & El-Badawy, A. (2015, October). Backstepping trajectory tracking control of a quadrotor with disturbance rejection. *In 2015 XXV International IEEEConference on Information, Communication and Automation Technologies (ICAT),* 1-7.

[194] Amin, R. U., & Aijun, L. (2017). Design of mixed sensitivity H∞ control for four-rotor hover vehicle. *International Journal of Automation and Control,* 11(1), 89-103.

[195] Satici, A. C., Poonawala, H., & Spong, M. W. (2013). Robust optimal control of quadrotor UAVs. *IEEE Access*, 1, 79-93.

[196] Ramírez, L., Zúñiga, M., Romero, G., Lara, D., Rabhi, A., & Pegard, C. (2015). Fuzzy Control of a Quadrotor for Reconstructed States Feedback Using Multiobserver. Applied Mechanics and Materials, 811, 172-178.

[197] Gonzalez-Hernandez, I., Palaciossn, F. M., Cruz, S. S., Quesada, E. S., Leal, R. L. (2017). Real-time altitude control for a quadrotor helicopter using a super-twisting controller based on high-order sliding mode observer. *International Journal of Advanced Robotic Systems* ,14(1), 1-15.

[198] Ye, Q. (2018). Event-driven PID control of autonomous quadrotor helicopters. *Journalof Algorithms & Computational Technology,* 2(2), 159-164.

[199] Yang, Y., Yan, Y. (2017). Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerospace Science and Technology*, 54, 208-217.

[200] Guldner, J., Utkin, V., & Shi, J. (1999). Sliding mode control in electromechanical systems. (2nd) *Taylor & Francis, London*. https://doi.org/10.1201/9781420065619

[201] Bouadi, H., Cunha, S. S., Drouin, A., & Mora-Camino, F. (2011, November). Adaptive sliding mode control for quadrotor attitude stabilization and altitude tracking. *In 2011 IEEE 12th international symposium on computational intelligence and informatics (CINTI),* 449-455.

[202] Barbot, J., Djemai, M., & Boukhobza, T. (2002). Sliding mode observers. *Sliding mode control in engineering,* 11, 33.

[203] Guisser, M., & Medromi, H. (2009). A High Gain Observer and Sliding Mode Controller for an Autonomous Quadrotor Helicopter. *International Journal of Intelligent Control and Systems,* 14(3), 204-212.

[204] Dávila, A., Moreno, J. A., & Fridman, L. (2009). Optimal Lyapunov function selection for reaching time estimation of Super Twisting algorithm. *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, Shanghai*, 8405-8410.

[205] Shtessel, Y., Edwards, C., & Levant, A. (2014). Sliding mode control and observation. (10), *Springer, New York,* 159-253.

[206] Ahmed, M. F., Zafar, M. N., & Mohanta, J. C. (2020, February). Modeling and analysis of quadcopter F450 frame. *In 2020 international IEEE conference on contemporary computing and applications (IC3A),*196-201.

[207] APC propeller. (2014). PERFORMANCE DATA (advance ratio and MPH) of APC 10x4.7SF. APC 10x47SF Datasheet. Dec 2022. www.apcprop.com/files/PER3_10x47SF.dat

[208] McCrink, M., & Gregory, J. W. (2015). Blade element momentum modeling of low-Re small UAS electric propulsion systems. *In 33rd AIAA Applied Aerodynamics Conference*. 3296.

[209] Kumar, V. A., Sivaguru, M., Janaki, B. R., Eswar, K. S., Kiran, P., & Vijayanandh, R. (2021). Structural Optimization of Frame of the Multi-Rotor Unmanned Aerial Vehicle through Computational Structural Analysis. *Journal of Physics: Conference Series*, 1849(1), 012004. IOP Publishing.

[210] Mishra, A., Pal, S., Malhi, G., & Singh, P. R. A. B. H. A. T. (2020). Structural analysis of UAV airframe by using FEM techniques: A review. *International Journal of Mechanical and Production ISSN*, (10), 2249-6890.

[211] Martinez Leon, A. S., Rukavitsyn, A. N., & Jatsun, S. F. (2021). UAV airframe topology optimization. *In Proceedings of the 6th International Conference on Industrial Engineering (ICIE 2020) , Springer International Publishing* ,(6) , 338-346.

[212] Barenthin, M. (2006). On input design in system identification for control. Doctoral dissertation, Stockholm: Signaler, sensorer och system , 2006. , p. x, 121.

[213] Söderström, T., & Stoica, P. (1989). System Identification. *Prentice Hall International Series in Systems and Control Engineering*. *Hemel Hempstead, UK*.

[214] Lohani, A. K., Goel, N. K., & Bhatia, K. K. S. (2006). Takagi–Sugeno fuzzy inference system for modeling stage–discharge relationship. *Journal of Hydrology,* 331(1-2), 146-160.

[215] Nwobi-Okoye, C. C., Ochieze, B. Q., & Okiy, S. (2019). Multi-objective optimization and modeling of age hardening process using ANN, ANFIS, and genetic algorithm: Results from aluminum alloy A356/cow horn particulate composite. *Journal of Materials Research and Technology,* 8(3), 3054-3075.

[216] Baloch, M. A., Ismail, I., & Baloch, T. M. (2010, June). Anfis identification model of an advanced process control (apc) pilot plant. *In 2010 International IEEE Conference on Intelligent and Advanced Systems*, 1-5.

[217] Jang, J. S. (1996, September). Input selection for ANFIS learning. *In Proceedings of IEEE 5th international fuzzy systems,* (2), 1493-1499.

[218] Yeom, C. U., & Kwak, K. C. (2018). Performance comparison of ANFIS models by input space partitioning methods. *Symmetry*, 10(12), 700.

[219] Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., ... & Lin, C. T. (2017). A review of clustering techniques and developments. *Neurocomputing*, 267, 664-681.

[220] Kyriakopoulou, A. (2008). Text classification aided by clustering: a literature review. *IntechOpen*.

[221] Widodo, I. D. (2018). Fuzzy subtractive clustering based prediction model for brand association analysis. *In MATEC Web of Conferences EDP Sciences* (154), 01082.

[222] Rao, U. M., Sood, Y. R., & Jarial, R. K. (2015). Subtractive clustering fuzzy expert system for engineering applications. *Procedia Computer Science*, 48, 77-83.

[223] Benmouiza, K., & Cheknane, A. (2019). Clustered ANFIS network using fuzzy c-means, subtractive clustering, and grid partitioning for hourly solar radiation forecasting. *Theoretical and Applied Climatology*, 137, 31-43.

[224] Tudoroiu, R. E., Zaheeruddin, M., & Tudoroiu, N. (2018, September). MATLAB Implementation of an Adaptive Neuro-Fuzzy Modeling Approach Applied on Nonlinear Dynamic Systems-a Case Study. *In 2018 Federated IEEE Conference on Computer Science and Information Systems (FedCSIS)*. 577-583.

[225] Jang, J. S. R., & Gulley, N. (1995). Fuzzy logic toolbox user's guide. *The Mathworks Inc*, 1(995), 19.

[226] Jalali-Heravi, M., & Kyani, A. (2007). Comparison of Shuffling-Adaptive Neuro Fuzzy Inference System (Shuffling-ANFIS) with Conventional ANFIS as Feature Selection Methods for Nonlinear Systems. *QSAR & Combinatorial Science,* 26(10), 1046-1059.

[227] Nakariyakul, S., & Casasent, D. P. (2008, August). Improved forward floating selection algorithm for feature subset selection. *In 2008* IEEE *International Conference on Wavelet Analysis and Pattern Recognition,* (2), 793-798.

[228] Gibson, J., & Hadi, M. U. (2022). Modeling and Optimal Control for Rotary Unmanned Aerial Vehicles in Northern Ireland Climate. *Applied Sciences,* 12(15), 7677.

[229] Maghfiroh, H., Nizam, M., Anwar, M., & Ma'Arif, A. (2022). Improved LQR control using PSO optimization and Kalman filter estimator. *IEEE Access,* 10, 18330-18337.

[230] Su, M. C., Liu, C. W., & Tsay, S. S. (1999). Neural-network-based fuzzy model and its application to transient stability prediction in power systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 29(1), 149-157.

[231] Aghazamani, A. M., Khodabandeh, M., Razavi-Far, R., Zarei, J., & Saif, M. (2022). Quadrotor Attitude and Altitude Tracking Control Using Finite Discrete-Time Linear Quadratic Tracking Controller. *IFAC-PapersOnLine*, 55(10), 2962-2967.

[232] Yu, Y., & Ding, X. (2012). A quadrotor test bench for six degree of freedom flight. *Journal of Intelligent & Robotic Systems*, 68, 323-338.

[233] Barbosa, F. D. S. (2017). 4DOF quadcopter: development, modeling and control, Doctoral dissertation, Universidade de São Paulo. https://doi.org/10.11606/D.3.2017.tde-23102017-144556

[234] ifm electronic gmbh. (2022), *Time domain*. IFM. Access *Augst* 2022. https://www.ifm.com/us/en/us/real-time-maintenance/vibration/technology/time-domain

[235] Kumar, J. P., Chauhan, P. S., & Pandit, P. P. (2022). Time domain vibration analysis techniques for condition monitoring of rolling element bearing: A review. *Materials Today: Proceedings,* 62, 6336-6340.

[236] Hanly, S. (2018). The Ultimate Guide to Vibration Analysis. *Design World, WTWH Media,* LLC, 12.

[237] Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics,* 23(3), 665-685.

[238] Walia, N., Singh, H., & Sharma, A. (2015). ANFIS: Adaptive neuro-fuzzy inference system-a survey. *International Journal of Computer Applications*, 123(13).

[239] Tokat, S., Fadali, M. S., & Eray, O. (2015). A classification and overview of sliding mode controller sliding surface design methods. *Recent Advances in Sliding Modes: From Control to Intelligent Mechatronics,* 417-439.