REPUBLIC OF IRAQ

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

AL-FURAT AL-AWSAT TECHNICAL UNIVERSITY

ENGINEERING TECHNICAL COLLEGE-NAJAF

DESIGN AND SIMULATION OF FLIGHT CONTROL SYSTEM FOR PARROT MINI-DRONE

ESRAA HADI KADHIM

(B. Sc. Communication Techniques Eng.)

2022

DESIGN AND SIMULATION OF FLIGHT CONTROL SYSTEM FOR PARROT MINI-DRONE

THESIS

SUBMITTED TO THE (COMMENICATION TECHNIQUES ENGINEERING DEPARTMENT)

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF (MASTER DEGREE)

BY

ESRAA HADI KADHIM

Supervised by

Prof. Dr. Ahmad T. Abdulsadda

2022

سورة الانبياء ايه 33

سورة يس اية 40

Dedication

To the greatest person that Allah has ever created, Prophet Mohammad

Peace is on him and his family**.**

To the best person that Allah has ever created after His Prophet, Imam Ali

Peace is upon him**.**

To those who have all the credit on me, to those who were the cause of my existence, to my beloved parent**.**

To my husband, my brothers and my friends**...**

To all who supported and encouraged me to achieve my success

Declaration

I hereby declare that the work in this thesis my own except for quotations and summaries which have been duly acknowledged.

2022 Esraa Hadi Kadhim

Supervisor Certification

I certify that this thesis titled " Design and Simulation of Flight Control System for Parrot Mini-Drone" which is being submitted by Esraa Hadi Kadhim was prepared under my supervision at the Communication Techniques Engineering Department, Engineering Technical College-Najaf, AL-Furat Al-Awsat Technical University, as a partial fulfillment of the requirements for the degree of Master of Technical in Communication Engineering.

Signature:

Name: **Prof. Dr. Ahmad T. Abdulsadda**
(Supervisor)

Date: / / 2022

In view of the available recommendation, I forward this thesis for debate by the examining committee.

Signature:

Name: **Prof. Dr. Ahmad T. Abdulsadda**

(Head of comm. Tech. Eng. Dept.)

Date: / / 2022

Committee Report

We certify that we have read this thesis titled " Design and Simulation of Flight Control System for Parrot Mini-Drone " which is submitted by Esraa Hadi Kadhim and as Examining Committee, examined the student in its contents. In our opinion, the thesis is adequate for the award of degree of Master of Technical in communication Engineering.

Signature:
Name: **Prof. Dr. Ahmad T. Abdulsadda**
(Supervisor)
Date: / / 2022

Signature:
Name: **Asst. Prof. Ali Mohamed Saeed**
(Member)
Date: / / 2022

Signature:
Name: **Dr. Ahmed Fahem ALBAGHDADI**
(Member)
Date: / / 2022

Signature:
Name: **Prof. Dr. Laith Ali Abdul-Rahaim**
(Chairman)
Date: / / 2022

Approval of the Engineering Technical College- Najaf

Signature:

Name: **Asst. Prof. Dr. Hassanain Ghani Hameed**

Dean of Engineering Technical College- Najaf

Date: / / 2022

Linguistic Certification

This is to certify that this thesis entitled "**Design and Simulation of Flight Control System for Parrot Mini-Drone**" was reviewed linguistically. Its language was amended to meet the style of the English language**.**

Signature**:**

Name**:**

Date:

Abstract

The flight management system in an unmanned aircraft represents the entire design of the unmanned aircraft with all its parts and components. The problem with the research is that drones are self-driving and are always used in dangerous places that humans cannot reach. They are made of lightweight materials, so they are not durable, and during their flight, they are exposed to different weather conditions, so it is necessary that the control system in them be solid and appropriately designed for the purpose of maintaining balance and stability of the aircraft during flight.

In this research, we use the Parrot mini drone Mambo design, whose aircraft is characterized by the ability to program it and change its design using the Matlab program. We present a complete study of the flight management system in this aircraft, starting from the sports model and design to the software and hardware components.

In this study, we are designing an aircraft altitude control system, which is, an integrated proportional PD controller, and the ability to work with linear and non-linear systems.

The researchers also improve the size of the aircraft propellers when they apply the new designs in the MATLAB R2021a program. The results show that the new controller makes the aircraft able to withstand a downward drag force of 12.81m/s^2, while the old design does not bear more than the ground force of the body, and the new design for the size of the propellers makes the aircraft capable of carrying an overload of approximately one-third of the weight of the aircraft.

Also, made the plane follow a square-shaped path. The cross-tracking error ratio, which is the ratio between the cross path of the plane's location and the specified path, which is a positive numerical measurement calculated in meters from the plane's vertical height above the path, is 0.3m.

Acknowledgements

First and foremost, I would like to express my sincere gratitude and acknowledgement for Prof. Dr .Ahmad Taha Abdulsadda for his valuable guidance, advice, and timely care extended to me during my research period. I am able to successfully complete this research work and deliver this thesis because of his immense patience and constructive feedback**.**

Also my grateful for my faculty (Department of Communication Engineering, Engineering Technical College/Najaf, Al-Furat Al- Awsat Technical University). I would like to thank all of my friends and colleagues.Special thanks are to my many friends who have given me their support and help MSC: Mustafa Abdul Mahdi Mohammed Hasan and MSC Amal Salman, I would like to thank my parents and the rest of my family; to my Dad and Mum for always being there through the good and the bad times and their everlasting love and support **.**

Without all the people mentioned above, I would not be the person I am today, all giving me the strength and drive to succeed and complete this project.

Contents

List of Table

List of figures

List of Abbreviations/ Nomenclature

Symbol	Definition
ARX	Autoregressive exogenous
CCW	Counterclockwise
CW	Clockwise
DOF	Degree of freedom
FCS	Flight control system
FDD	Fault Detection and Diagnosis
FPI-PD	FuzzyPI-PD controller
IMU	Inertial Measurement Unit
ISE	Indexes to the least
Kd	Derivative gain
Ki	Integrated gain
Kp	Proportional gain
LIDAR	Light Detection and Ranging
LMI	Linear Matrix Inequalities
LPV	Linear parameter-varying
MEMS	Micro-Electro-Mechanical Systems

MMA	Mix motor algorithm
NARX	Non-linear autoregressive exogenous
PD	Proportional and derivative controller
PID	proportional-integral-derivative controller
PMD	Parrot mini-drone
PWM	Pulse width modulation
SDK	software development kit
SISO	Single input single output
SNR	Signal-to-noise ratio
UAVs	Unmanned aerial vehicles
VRS	Vortex Ring State
WBS	Windmill-Brake State
WP	Waypoint

List of symbols

Symbol	Description
α	Angular acceleration
a	Acceleration of quadrotor
A	Area of propeller
C	Lift coefficient
f	Forces on a particle object
g	Constant of gravitation
G(s)	The transfer function of the system
Gc(s)	Transfer function of the controller
GT(s)	The total transfer function
J	The matrix of global inertia
m	Mass of the object
P	Air density constant
q	Quaternions matrix
R	Rotation matrix

U(s)	input signal
v	Velocity of quadrotor
V	Propeller speed
wn	Natural frequency
Y(s)	Output signal
ζ	Damping ratio
τ	Torque
Ω	Skew matrix
ω	Angular velocity
Θ	Roll angle
Φ	Pitch angle
Ψ	Yaw angle

List of Publications

[1] Esraa Hadi Kadhim, Ahmed Tahha Abdulsadda, " Mini Drone Linear and Nonlinear Controller System Design and Analyzing" published in the Journal of robotics and controller (JCR) vol. 3, no. 2, pp. 212–218, 2022, doi: 10.18196/jrc.v3i2.14180. Which is Scopus indexed and has an impact factor.

[2] Esraa Hadi Kadhim, Ahmed Tahha Abdulsadda "The Flight Mangment System in Parrot Mini-Drone " Accepted in the 1st International Conference on Achieving the Sustainable Development Goals ICASDG2022 Conference Proceeding. The accepted manuscripts published in the AIP conference proceedings which is Scopus indexed and has an impact factor.

[3**]** Esraa Hadi Kadhim, Ahmed Tahha Abdulsadda " Improving the Size of the Propellers of the Parrot Mini-Drone and an Impact Study on its Flight Controller System " Submission in the 6st International Conference on New Trend Information and Communication Technology Applications. The accepted manuscripts published Springer in their Communications in Computer and Information Science series CCIS which is Scopus indexed and has an impact factor.

Chapter 1: Background and literature review

1.1 Introduction

Unmanned aerial vehicles (UAVs) have become increasingly widespread and important in both military and civilian settings. Micro UAVs in their current state rely substantially on human intervention to fly. Quadrotor drones have become a regular sight in today's culture. The Federal Aviation Administration of the United States now refers to any size UAV as simply "a UAV." A quadrotor drone is a UAV that is normally flown by a remote control guided by a human or autonomously by an on board computer, according to this description. The difference between the two methods of control is that an autonomous drone does not need human involvement or intervention to complete a task[1].

UAVs have grown more economical and viable for a variety of applications thanks to advancements in sensor technology, control theory, and aerodynamic science. The majority of UAVs on the market are under one meter in length and have the benefit of being portable. The size is a selling element, but it also causes problems. When the environment is unpredictably unstable, stability is essential. Another concern with smaller drones is that the sensors built for the platform are smaller, potentially creating unwanted noise, vibrations, and temperature rises within the tight housing.

As seen in[2],[3] ,[4] and[5] these priority areas have been thoroughly documented and are constantly being developed and improved upon. Autonomous micro UAVs have the mobility and potential to allow the operator to utilize a drone to complete a task without having to keep a continual eye on where and what the drone is doing. Many sectors, both civilian and military, would profit immensely from autonomous drones that can hunt for a specific thing. These drones are both pricey and huge in most circumstances. The application of this technology to smaller UAVs will result in a more intuitive and nimble UAV capable of performing duties without the use of a ground-based controller. MathWorks provides a ready-made generic model of a quadcopter together with a three-dimensional visual simulator that illustrates the system to be managed in this project (Simulink 3D environment).

The technique that defines model-based software design is depicted in the figure below (Figure 1.1)[6]. The control system for calculating the location and altitude of the aircraft's proportional and derivative controller was designed in this study (PD).

This design was employed in the Parrot Mini Drone Mambo, and the findings that explained in Chapter four of the research showed that it was useful in boosting the aircraft's tolerance to the adverse circumstances which may encounter during flight.

In this research, designed the area of the plane's propellers. Also found that the best space for an aircraft propeller is (4 cm - 4.3 cm) as explained in Chapter Four. also studied the impact of the new propeller area on the drone control system and indeed applied this in the Matlab Simulink program, and got good results. Using the new design, we directed Drone to track a specific path as shown in Chapter Four, and calculated the line ratio of this track.

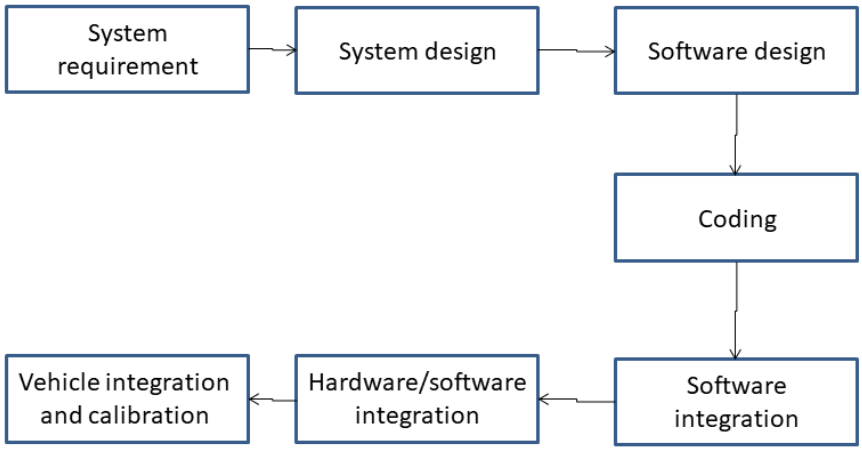


Figure (1. 1) Model-based design workflow

1.2 Literature review

Due to their simplicity and rapid response time, quadrotor drones are one of the most popular aerial vehicles. A flight command that permits the platform to follow a trajectory has been developed after several trials. A proportional-integral-derivative (PID) controller is the most prevalent type of control. This type of control compares the actual measured state by the sensors to the referenced input of where the drone is supposed to be. It determines the mistake based on this comparison and strives to minimize it to achieve its aim. In this chapter, explained the latest previous studies

and the ones closest to our work, which use the Parrot mini Drone, and start mentioning them from the oldest to the most recent.

Juan S. Guerrero, et al. 2015[7]**.** Have done a study on Unmanned Aerial Vehicle (UAV) Data Processing and Instrumentation of an Array of Ultrasonic Sensors for Teaching the Application of the Kalman Filter. It is possible to attenuate and, to some degree, erase irregularities in sensor values by using filters. It's worth noting that the filter's stability takes time to develop; this is something to consider if they want to construct a dynamic and real-time model. They build the filter method for UAV real-time flying in the future and this data will be used as input to the algorithm path planning and obstacle avoidance. As can be seen, this method makes it easier to comprehend the performance of the Kalman filter.

Ahmed Hussein, et al. 2015[8]**.** They looked at low-cost quadcopters' autonomous in door navigation. The output path is broken down into many waypoints, which the drone controller reads and responds to with flying orders. The drone is controlled by a PID Simulink model that manipulates the drone's internal controller for pitch, roll, yaw, and vertical speed. The ability of the drone to follow the waypoint given by the path planning algorithm with a little mistake is the key contribution. The results of the experiments demonstrated the disparities in the quadcopter's manoeuvring ability in various settings. Despite the modest cost of the sensors utilized, the pose and distance errors were negligible, indicating the good performance of the suggested algorithms and their applicability in many navigation applications. Performing further test studies using a visual-based strategy is one of the research's future goals. For the comments on localization and compare it to the results obtained furthermore, dealing with complex surroundings containing dynamic impediments necessitates expanding the problem beyond path planning to trajectory planning.

LEFEBER, A.A.J. 2015[9] **.** Developed the non-linear dynamical model of the Quadrotor, as well as a linearized model for maintaining hovering, are presented in this article. The model is put up such that the controller for the non-linear dynamical model may be developed. Before integrating the dynamical model into the Quadrotor SDK, this project design the controller and implement the controlled model in the non-linear model.

Yunmok Son, et al.2015[10]**.** Studied on gyroscopic sensors, rocking drones with intentional sound noise and used MEMS gyroscopes contain resonance frequencies that reduce their accuracy. It is unknown if this property may be used maliciously to

interrupt drone operations. First, a consumer-grade speaker is used to test 15 different MEMS(Micro-Electro-Mechanical Systems) gyroscopes against sound noise, and the resonance frequencies of seven MEMS gyroscopes were determined by scanning the frequencies under 30 kHz. The resonant output from the gyroscopes had a standard deviation hundreds of times greater than the normal output. After examining the flight control system of a target drone, real-world experiments and a software simulation were used to confirm the effect of the constructed gyroscope output. Experiments in the actual world revealed that in each of the 20 trials, one of two targets was hit. Shortly after began attacking, drones equipped with vulnerable gyroscopes lost control and crashed.

Pengcheng Wang, et al. 2016 [11] **.** This research used a dynamic model of a quadcopter to construct a durable cascade PID control technique. The cascade PID control scheme's key benefit is its high tolerance to external disturbances. In addition, the efficacy of the developed controller was demonstrated by comparing conventional and cascade PID control systems. The suggested control algorithm is supported by simulation results of quadcopter attitude control. The cascade PID control approach gives the quadcopter system a significant performance gain. The focus of future study on creating sliding mode control algorithms using a nonlinear dynamic model, in order to increase the quadcopter system's robustness and performance in the face of parameter uncertainty and external disturbances [9].

Prasant Misra, et al. 2016[12]**.** Developed aerial drones with location-sensitive ears, the goal of this work is to develop a binaural sound source geolocation system. It combines the advantages of sparse (two-element) sensor array design (to meet platform constraints) with proposed mobility-induced beamforming based on intra-band and inter-measurement beam fusion (to overcome severe ego-noise and other complex characteristics) to significantly improve the received signal-to-noise ratio (SNR). Empirical assessments indicate the usefulness of (Drone EARS) in terms of SNR improvement over a variety of frequently utilized approaches.

Gabriele Perozzi, et al 2018[13]**.** Using on-board quadrotor sensors and an inertial tracking position system, three time-changing parameter estimation techniques were developed, investigated, and combined in this study to estimate variable wind velocity (e.g. Optitrack camera, GPS). A completed quadrotor flight dynamics model was provided employing identified aerodynamic coefficients and wind velocity components along the three axes to achieve this goal. Then, to estimate, a

decomposition of dynamical equations in known and unknown terms is done. The estimated algorithms were created using this decomposition, and then evaluated and confirmed numerically against the noise introduced by the sensors.

M. Andreetto, et al. 2019[14]**.** This work presented a data observer capable of making predictions, allowing the controller to be fed at a considerably quicker rate than the sluggish sensor data rate enables. A linear model, whose parameters are identified on-line using a Constrained Kalman Filter, generated the predictions. Extensive testing with actual drones performing altitude stabilization and trajectory tracking tasks has effectively verified the method. Even in the presence of substantial disruptions, the limited model identification provided a stable forecast (which is physically meaningful) and hence safe flying.

Mücahid Rıdvan Kaplan, et al. 2019[15]**.** The PI-PD and Fuzzy PI-PD (FPI-PD) structures were designed and used in this work to tackle the Parrot Mambo Minidrone's position control challenge. In this regard, a nonlinear mathematical model of the Parrot Mambo Minidrone was developed in order to get the minidrone's control models. Then, using control models, PI-PD control systems for altitude and position control systems are constructed. FPI-PD controllers were then created and used in the minidrone's position control loop to tackle the coupled nonlinearities. The findings of the provided comparative real-world experiment reveal that the suggested control systems outperform the built-in ones. Furthermore, the results reveal that the FPI-PD control system outperformed the built-in PD and created PI-PD control systems.

Gennaro Ariante, et al. 2019 [16]**.** The authors developed low-cost sensors, such as a 10-DOF MEMS (Micro Electro-Mechanical System) IMU (Inertial Measurement Unit) and a LIDAR (Light Detection and Ranging), were installed on a small unmanned rotorcraft in this paper and synchronized at a 10-Hz measurement rate to estimate the position of the platform and its distance from an obstacle or a landing field. Kalman filtering was used to correct the IMU data for systematic errors (bias) and measurement noise, as well as to obtain predicted locations from accelerometer data. The technique was created on an on-board microprocessor (Arduino Mega 2560) and enables for low-cost hardware implementations of many sensors for usage in aerospace applications.

Bushra, Tooba Hai and Muhammad Bilal Kadri , 2019[17]**.** Three alternative system identification strategies have been used to mathematically describe the Parrot

AR. Drone 2.0 quadcopter in this article. SISO Continuous-time transfer functions were initially computed independently for lateral, longitudinal, and vertical loops. The parameters for ARX (autoregressive exogenous) and NARX (non-linear autoregressive exogenous) models were then iteratively trained. The results of this study, which are based entirely on real and experimental data, show that the NARX model, which is based on a series-parallel architecture, is the best for quadcopter state estimation. Multiple test flights were used to evaluate the efficiency of this network in contrast to the other two.

Oliviu Mihnea Gamulescu, et al. 2020[18].The use of a drone equipped with a camera and sensors network capable of transmitting data in real-time over post-explosive environmental conditions to the command center for the rapid identification of victims both in open and closed spaces is proposed in this paper as a quick rescue intervention. The strategies for employing drones to investigate and identify accident locations are presented in this research. The control system and the image identification unit were both modelled and simulated.

Noordin, M. A. M. Basri and Z. Mohamed, 2020[19]**.** This study provides a proportional-integral-derivative (PID) flying controller. Because the (Parrot Mambo mini-drones) were so tiny, even a minor change have an impact on it was performance. As a result, a PID control technique was provided for actuation dynamics such as roll, pitch, yaw, and z stabilization. A similar controller approach was also used to regulate the x and y positions of under-actuated dynamics. Simulink is used to simulate the Newtonian model, with external disturbances being normal Gaussian noise of force. The technique was implemented utilizing Bluetooth® Low energy connection through personal area network using Simulink support package for Parrot Minidrones by MATLAB and depending on the simulation parameter personal area network (PAN). During hovering, a little force was imparted to the inquiry system called a disturbance. Finally, the technique is deployed utilizing Bluetooth® Low energy connection via personal area network with Minidrones using MATLAB and based on the simulation parameter (PAN).

Jutarut Chaoraingern. Et al, 2020[20]**.** This research used modified adaptive sliding mode control to monitor the trajectory of mini-drone quadcopter unmanned aerial vehicles, with the purpose of demonstrating the utility of a nonlinear adaptive control approach in achieving the expected performance of the mini-drone quadcopter system. In addition to providing mathematical modelling and nonlinear

dynamic characteristic details of the mini-drone quadcopter actuated system, the modified adaptive sliding mode algorithm was developed using adaptation law based on Lyapunov stability approach and then applied on the attitude loop and altitude loop control system. The findings indicated that even in the presence of parameter perturbations and disturbances, the updated adaptive sliding mode control can reduce error performance indexes to the least ISE of 1.041 m2 and zero per cent overshoot while retaining high stability and resilience.

Amin Talaeizadeh, et al. 2020[21] . In the context of small-scale quadcopters, this work investigated the phenomena of helicopters falling quickly entering the so-called Vortex Ring State (VRS). Combining first-principles modelling and wind-tunnel measurements, the area corresponding to the VRS was discovered. In addition, suggest that the so-called Windmill-Brake State (WBS) or autorotation zone be avoided for quadcopters, which was not always the case for helicopters. A model was suggested for the quadcopter's velocity limitations to avoid these locations. Then there was the difficulty of developing optimal temporal descent trajectories that avoided the VRS and WBS zones. Finally, the best paths were programmed into a quadcopter. The flying experiments reveal that following the specified paths allows the quadcopter to drop much quicker than following completely vertical routes that avoid the VRS and WBS.

Houria Siguerdidjane and Gabriele Sordi, 2020[22] This document describes the many sorts of projects that students at Centrale Supelec, School of Engineering, Université Paris-Saclay work on in the field of automated control and signal processing applications. The courses were given throughout academic teaching semesters, as well as optional courses and laboratory practice sessions, which were geared to helping students to gain information and develop abilities in a variety of disciplines. Some of these tasks, such as indoor tests and organization planning, are mentioned in this study.

Yukai Chen. et al, 2020[23]**.** This work proposed a battery-aware model for determining the energy consumption of drones, which was subsequently used to a drone delivery scenario. In comparison to the usual estimating methodology, the results revealed an accuracy of almost 16 percent.

Adarsh Kumar. et al, 2021[24]**.** This study looked at drone-based systems and COVID-19 pandemic conditions and proposed architecture for dealing with pandemic events in real-time and simulation-based scenarios. In a push-pull data

fetching technique, the suggested architecture uses wearable sensors to capture observations in Body Area Networks (BANs). The proposed design was shown to be beneficial in rural and crowded pandemic locations where wireless or Internet access was a big concern or where the risk of COVID-19 spreading was high. During the installation of a real-time drone-based healthcare system for COVID-19 operations, it was discovered that a significant area may be covered for sanitization, thermal image collecting, and patient identification in a short amount of time (2 KMs in around 10 minutes) through aerial route. The simulation uses the same statistics as the real world. Collision-resistant techniques for indoor and outdoor healthcare operations were reported to be successful.

S. Waitman, H. Alwi and C. Edwards, 2021[25] **.** This research looked at the numerical design and practical implementation of a linear parameter-varying (LPV) sliding mode observer for quadrotor mini drone Fault Detection and Diagnosis (FDD). An LPV model was retrieved for design from a nonlinear model of the mini drone, and the observer synthesis technique was described in detail using Linear Matrix Inequalities (LMI). The observer FDD simulations offer excellent results. The observer was then installed on a Parrot® Rolling Spider mini drone, and a series of flight tests were conducted to evaluate the FDD's capabilities in real time utilizing the mini drone's on-board processing capacity. The results of the flying tests back up the modelling results, demonstrating that the sliding mode observer can reliably reconstruct faults for quadrotor mini drone systems.

Rafael Casado and Aurelio Bermúdez, 2021[26]**.** This paper describes a simulation framework that can assist engineering students who are new to the subject of aerial robotics in gaining the essential competencies and abilities to design autonomous drone navigation systems. Drone behaviour was specified graphically using state machines, while low-level control details were abstracted. Demonstrate how to utilize the framework to create a navigation system proposal that adheres to the guidelines of the "ESII Drone Challenge" student competition. Demonstrated how the proposal may be tested in various settings.

The figure (1.2) shows the parts of the flight management system in the drone and the part that previous studies focused on developing.

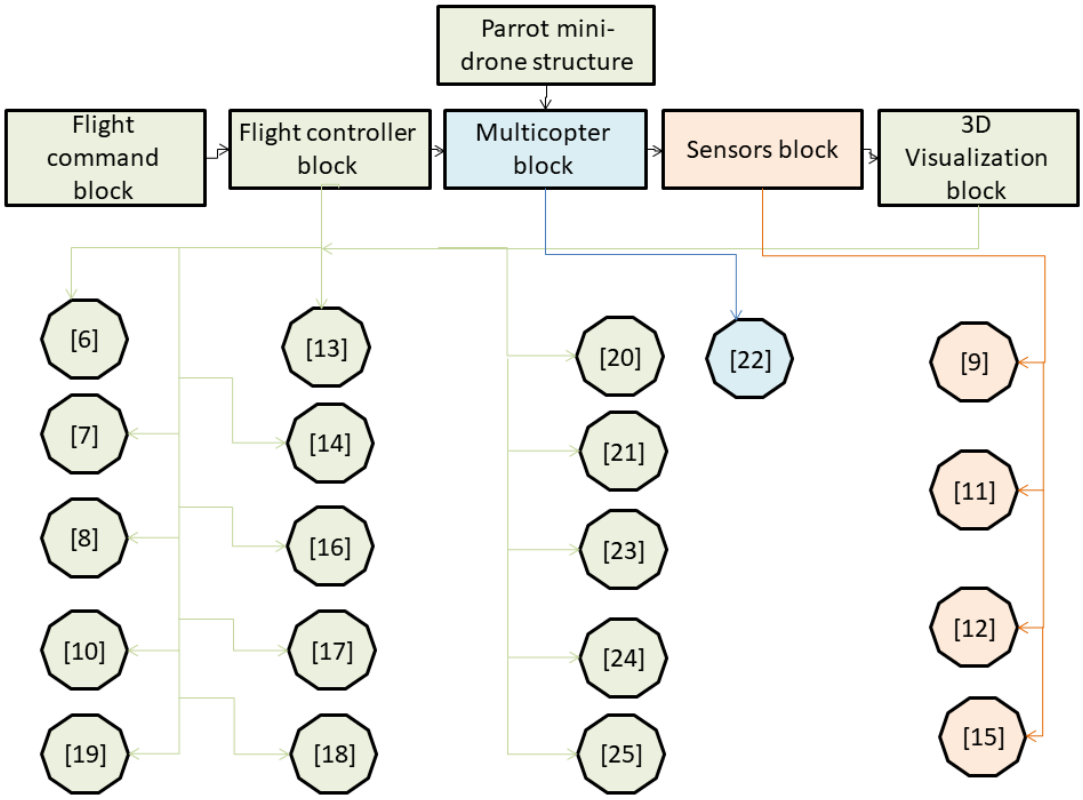


Figure (1. 2) the flight management system of mini drone and the parts that the researchers worked on developing.

The Table (1.1) below summarizes the most important findings and discussion of previous research findings.

Table (1.1) Summary of results reached by researchers in previous studies

No.	Number of references	Results and discussion
1	[7]	The HC-SR04 ultrasonic sensors are connected to the Arduino Yun data acquisition board, which interprets the electronic pulses to calculate the distance in meters and sends the data via serial port to a Raspberry Pi board, which implements the Kalman filter and sends the data via Wi-Fi to a computer connected to the network, allowing visualizing the results of measurements and filter performance. The acquired results reveal an increase in data measurement using distance sensors, with the goal of providing the drone with improved data collection accuracy.
2	[8]	Under various circumstances, the experimental results demonstrated the disparities in the quad-manoeuvring copter's ability. Despite the modest cost of the sensors utilized, the pose and distance errors were negligible, indicating the good

		performance of the suggested algorithms and their applicability in many navigation applications.
3	[9]	The non-linear dynamical model of the quadrotor, as well as a linearized model for maintaining hovering, is presented in this article. The model is set up such that the controller for the non-linear dynamical model may be developed. Because there are no simulation findings, the similarity between the non-linear-dynamical model produced in this study and the models utilized in publications with comparable research is sufficient to proceed with future work.
4	[10]	• They determined that the resonance frequencies of numerous common MEMS gyroscopes are not only in the ultrasonic but also in the audible frequency bands, using a consumer-grade speaker, and examined their resonant output**.** • Using software analysis and simulations, researchers looked at the impact of MEMS gyroscopes' resonant output on drone flight control**.** • They devised an innovative method of assaulting drones with susceptible MEMS gyroscopes by making a deliberate loud noise.
5	[11]	An approach is provided for stabilizing the quadcopter's attitude such that the balanced state can be maintained in the face of disruptions. The Newton-Euler approach is used to create a mathematical model of quadcopter dynamics. It reveals the precise correlations between all of the variables. Following that, both linear and nonlinear state-space equations are developed, which are required for controller design and development. The simulations also show how the cascade PID algorithm outperforms the standard PID control system in terms of efficacy and resilience.
6	[12]	The key observation used in the design was that a single measurement at any given location does not improve the received SNR for a robust location estimate. However, if many such observations are taken from multiple measurement locations, the final consolidated observations improve the SNR level significantly and result in a highly accurate geo-location of the sound source.
7	[13]	Numerical studies are carried out utilizing a nonlinear UAV simulator to represent a quadrotor in the presence of external wind disturbances, which are then confirmed by in-house tests. Even with the inclusion of sensor noise, the algorithms perform well in estimating wind velocities. More research may be done by linking this wind estimator to a robust controller to extend the domain of flight and minimize the control effort on the rotors in real-time when it is required.
8	[14]	The proposed technique has been put to the test in a variety of scenarios. The sample periods of Ts = 510 ms, Tc = 20 ms, and Tv = 100 ms are the same in all of the studies. The identical experiment using the least-squares method usually yields

		unstable results (hence the Mambo crash).
9	[15]	The findings of the reported real-world comparison experiments clearly reveal that the suggested control methods outperform the built-in ones. Furthermore, the findings reveal that FPI-PD outperforms both built-in PD and tailored PI-PD control systems.
10	[16]	In both static measurements and landing simulations, the results demonstrate position estimate inaccuracies of a few millimeters.
11	[17]	The findings of this study, which are based entirely on real and experimental data, show that the NARX (non-linear autoregressive exogenous) model, which is based on a series-parallel architecture, is the best for quadcopter state estimation.
12	[18]	The solution proposed based on drones has the advantage of indirect interaction with the danger zone, as the mini-drone can cross the area affected by a safe altitude and cover a larger area in a much shorter time than ground robots, which can be controlled by the operator outside the danger zone using real-time video streaming. The control system was modeled in Matlab-Simulink to create the control system and for simulation functionality in order to increase the drone's stability and to achieve direct control from the command center's computer.
13	[19]	A PID control was devised to stabilize the mini-drone when it is subjected to minor perturbations. A PID controller with a gravity compensator was used for position control to guarantee that the quadrotor could effectively navigate to the required location by following the path as intended. The PID controller proves that the Parrot Mambo minidrone can achieve attitude stabilization, hover at an appropriate height, and then follow the intended route as designed in simulations and experiments with modest disturbances
14	[20]	The findings show that the updated adaptive sliding mode control can reduce error performance indices to the smallest ISE at 1.041 m2 and zero percent overshoot while maintaining outstanding strength and resilience even in the face of constraint worries and disturbances.
15	[21]	The time needed for various hypothetical fall paths may be estimated using the conclusions reached from this research. Consider the case when the quadcopter must fall 5 meters. Using the obtained maximum velocity of 0.6 m/s for pure descent trajectories and 2 m/s for a VRS(Vortex Ring State) avoiding trajectory, we can predict that a pure descent trajectory will take 8.3 s to fall 5 m, while a VRS avoiding trajectory would take 2.5 s.
16	[22]	The results were satisfactory in depicting the QUAV system. The simulation reveals QUAV dynamic properties such as logical instructions to change the rotor thrust, rotor limitations in processing multiple instructions at once, vehicle reliance of spontaneous control systems, and the fact that a motion in one degree of freedom may readily affect others. The created model shows promise as a foundation for QUAV analysis and control design.

17	[23]	In comparison to the usual estimating methodology, the results reveal an accuracy of more than 16 percent.
18	[24]	With a variety of 3–30 drones, a distance of 1200 kilometers may be covered in 18.900 to 22.93 minutes. Thermal image-based patient identification was proven to be particularly successful for the COVID-19 pandemic in an indoor activity. The drone utilized for indoor imaging and sanitization is expected to be compatible with its operations. In the current world, a tiny drone would be required for comparable tasks.
19	[25]	The results indicate that the observer can estimate system outputs with high precision while also accurately reconstructing actuator and sensor defects.
20	[26]	Worked on studying the flight controller system block and 3D-Virtualization block and programmed the plane to follow a specific path by writing code in the Path plane subsystem. The difference between the two methods is the complexity of the path taken by the aircraft, the height of the aircraft and the turning angle for each of these research.

In this research, we designed a special control system for the height of the drones of the type (PD) controller, so changed the size of the propellers of the drone by an appropriate amount to increase its ability to bear the extra weight, and guided the drones to follow a specific path and calculated the error rate for this path. applied all of the above in the Matlab program and got good results.

1.3 Problem statement

The main problem is how to keep the aircraft safe in bad weather. The problem can be summarized as follows:

1. Creating a control system that promotes better aircraft balance and stability.

2. Improving the size of the flight management system to make the aircraft more responsive to environmental changes.

3. For the purpose of studying the trajectory error rate, the aircraft is directed to follow a certain path.

1.4 Aims of the Study

Safety is always number one in the aviation industry. When designing any part, assembly, or aircraft, it is the primary need. The unmanned aerial vehicle (UAV)

business, which is still in its infancy, is sensitive to the effects of weather on flight. They are just as susceptible to turbulence and bad weather. This flaw poses a significant threat to the safety of UAV flying. This research offers the following:

1- Design the height controller in the drone, thus improving the altitude controller by using (PD) and tuning the values of (Kp and Kd) in the altitude controller of the Parrot Mini Drone Mambo to make it more bearable to external influence and to maintain its altitude.

2- Improve the area of propellers on a UAV to eliminate bias in wind speed, ambient temperature, pressure, humidity, and bearing excess weight.

3- Direct the aircraft to follow a specific route for the purpose of calculating Cross-track error from UAV position to path.

1.5 Contributions

This research contributes to the design and analysis of the flight management system of the Parrot Mini Drone Mambo aircraft and the design of a special control system (Altitude controller) to determine the location and height of the aircraft more tolerant to poor external conditions and choose a suitable propellers area's capable of bearing the excess weight of the aircraft. Guided the drone to follow a specific path.

1.6 Organized Content of Thesis

Chapter 2: This chapter is explaining the problem formulation theory and mathematical model.

Chapter 3: This chapter explains The Simulation Model Description and Methodology

Chapter 4: This chapter includes experimental setup and simulation results.

Chapter 5: It includes the conclusions and summaries reached by this thesis. It also contains future works.

Chapter 2: Problem Formulation Theory

2.1 Introduction

Controlling an unmanned aerial vehicle (UAV) is difficult for several reasons. The Parrot Mambo Drone (PMD) is a non-linear system with several inputs and outputs that are under-actuated. Only four actuators drive the system, which is tracked within another system that does not share the PMD's reference frame[27]. This chapter's goal is to lay the groundwork for the equations of motion that will be used to model quadrotor flight routes, explain the software and the hardware components of quadcopter.

2.2 Mathematical Model of Quadcopter

The operating concept of a quadrotor will be presented in this chapter, as well as its mathematical model, which will be investigated to derive its kinematics and dynamical equations (useful for the implementation on Simulink) [28] , [6],[29] and[30] both employed the same strategy.

Quadrotors are made up of the following primary components:

The frame, which comprises a center and four arms and provides physical support for the other components**.** Electronics and a battery are located in the frame's core**.** Each arm has a motor and a propeller**.**

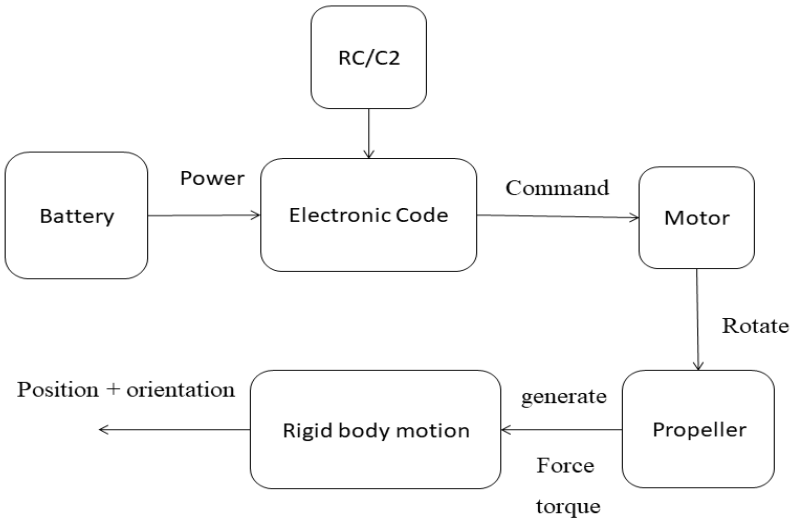


Figure (2. 1) Flowchart of the model of quadcopter

Thrust is created by a spinning propeller, which is a force that is perpendicular to the propeller's rotation plane. A spinning propeller generates a turning effect (or torque) on the quadrotor frame in addition to propulsion. It rotates in the opposite direction of the propeller.

Propellers are divided into two categories:

1- When revolving counters clockwise, a Type 1 propeller, often known as a right-handed propeller, created upward force.

2- When revolving clockwise, a Type 2 propeller, also known as a left-handed propeller, provides thrust in the upward direction.

Two Type 1 and two Type 2 propellers are used in a quadrotor (figure 2.2).

Figure (2. 2) Type of propeller

The quadrotor is a mechanism that is underactuated. It has four propellers, but six degrees of freedom in space.

• Up/down, forward/backward, left/right translational degrees of freedom.

• Degrees of freedom in rotation: heading, pitch, and roll.

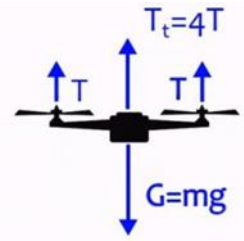
Some directions are not controlled at any given moment because the actuators (4) are smaller than the D.O.F. (6). The quadrotor, for example, cannot move right without spinning in that direction. The same may be said for forwarding and backward motions[28]**.**

Designing a control system that combines rotations and thrusts to achieve the overall goals can solve the underactuation problem.

Assume that the quadrotor's frame is completely level with the ground.

The entire thrust is in the vertical direction if the motors are given the same orders**.**

Adjust for gravity to create upward movement (Figure 2.3).

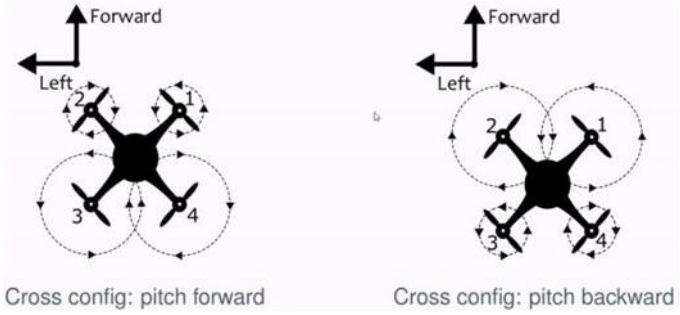


Figure(2. 3) While hovering, forces balance between thrusts and weight[6]**.**

The quadrotor will move down if the entire thrust is less than the force of gravity**.**

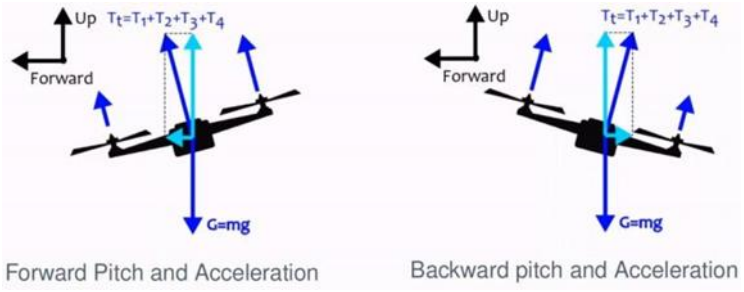
We must produce an imbalance between the forward-side and backward-side pressures to pitch, or rotate around the left-right axis. Pitching forwards is accomplished by reducing forward force and/or increasing backward force**.**

Pitching backward is accomplished by reducing force on the backside while increasing force on the forward side.



Figure(2. 4) Pitch Configuration [6]**.**

Translation in the forward/backward motion is combined with pitching rotation. When the quadrotor pitches forward or backward, it will travel forward or backward as well.



Figure(2. 5) Maintain equilibrium during the pitch movement [6]**.**

To roll, or rotate around the forward-backwards axis, we need to imbalance the forces on the left and right sides. Rolling right is accomplished by reducing force on the right side while increasing force on the left. Rolling left is accomplished by reducing the force in the left half while increasing the force in the right[29].

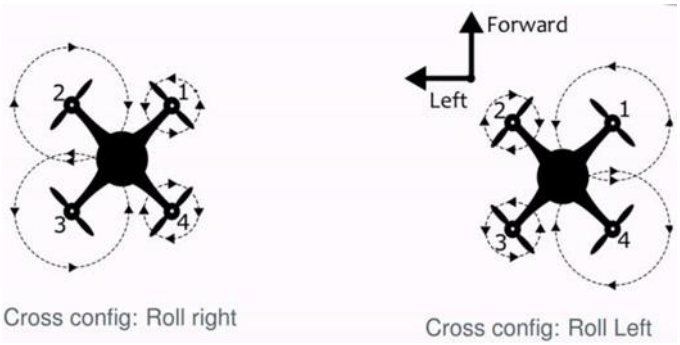
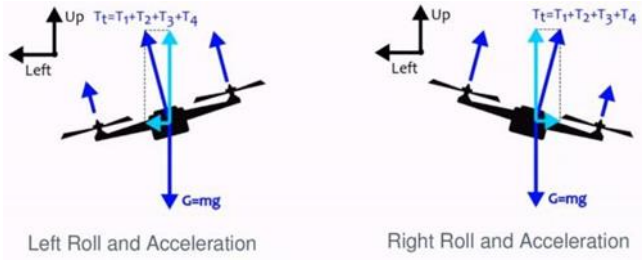


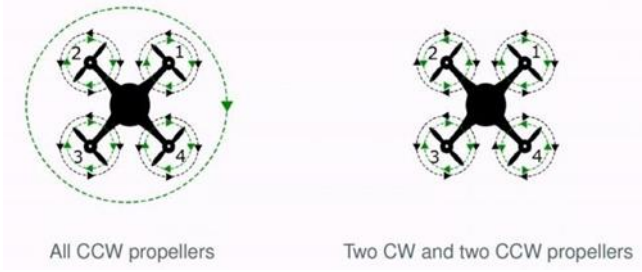
Figure (2. 6) Roll configuration

The left/right translation is synchronized with the rolling revolution. When the quadrotor rolls are left or right, it will travel left or right as well.



Figure(2. 7) During the Roll maneuver, forces balance [6]

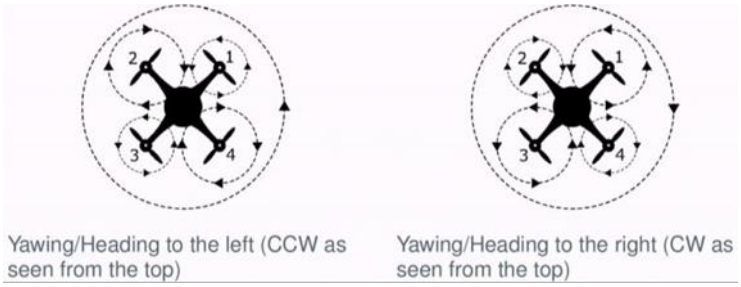
Yawing, on the other hand, is the rotation around the up/down axis. It's vital to note that rotating propellers cause the frame to revolve in the opposite direction. If all propellers rotated in the same direction, the frame would spin in place as it rotated in the opposite way around its up/down axis. The quadrotor's propellers are of two sorts, and they revolve in opposing directions in pairs, cancelling the response effect.



Figure(2. 8) Torque equilibrium [6].

To create a controlled counterclockwise yawing rotation, the input is increased on the pair of clockwise propellers and decreased on the counterclockwise pair, whereas the

input is increased on the pair of counterclockwise propellers and decreased on the clockwise pair to create a controlled clockwise yawing rotation.



Figure(2. 9) Balance is achieved by the yawing maneuver and torques [6]**.**

Pitching and forward/backward motion is coupled, rolling and left/right motion are coupled, up and down motion is independent, and yawing (change of heading) is independent, to summarize**.**

Implementing a motor mixing technique that converts roll, pitch, yaw, and thrust inputs into motor speeds and can be used in Simulink model representation which might be beneficial**.**

According to Douglas [31], the same techniques used to create Thrust, Roll, Pitch, and Yaw maneuvers are used in the same way to the.

$$M_{F/R} = thrust_{cmd} + yaw_{cmd} + pitch_{cmd} + roll_{cmd} \quad (2.1)$$

$$M_{F/L} = thrust_{cmd} - yaw_{cmd} + pitch_{cmd} - roll_{cmd} \quad (2.2)$$

$$M_{B/R} = thrust_{cmd} - yaw_{cmd} - pitch_{cmd} + roll_{cmd} \quad (2.3)$$

$$M_{B/L} = thrust_{cmd} + yaw_{cmd} - pitch_{cmd} - roll_{cmd} \quad (2.4)$$

Where $M_{F/R}$ is the motor front/right, $M_{F/L}$ is the motor front/left, $M_{B/R}$ is the motor back/right, $M_{B/L}$ is the motor back/left, thrust is the force of lifting the body upward, pitch is The lateral or transverse axis is the rotation of a vehicle fixed between the side-to-side axis (on an airplane, wingtip to wingtip), roll The longitudinal axis, also known as the front-to-back axis, is the rotation of the vehicle on the front-to-back axis (nose to tail), and yaw is the rotation of an airplane along its vertical axis, which is perpendicular to the propellers and in the center line[30].

2.2.1 Rigid Body Motion

A rigid body object is a system made up of a large (infinitely large) number of small (infinitely small) point-mass particles with the property that their relative positions stay constant (rigidly. To understand the accompanying graphic, the drone's rigid body motion model is necessary (2.10)**.** How can generalize forces in input be used to determine the location and attitude of the drone in the sky space.

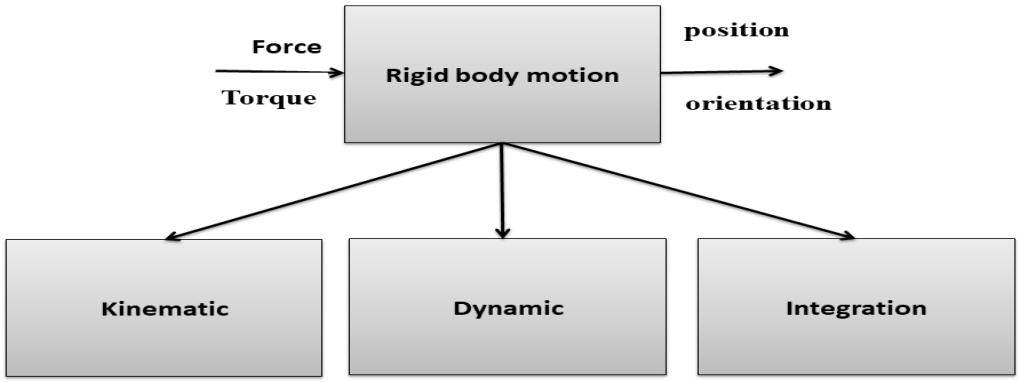
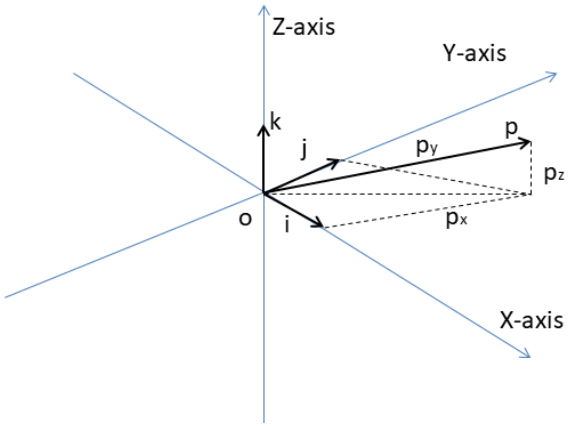


Figure (2. 10) Rigid body motion model.

2.2.2 Kinematics

Different topics are discussed and described in the Kinematics of the Rigid Body. Translational and rotational motion, position, linear velocity, and linear acceleration vectors, the rotation matrix that defines the body frame attitude relative to a fixed inertial frame, angular velocity and angular acceleration, motion composition between frames, and so on are just a few of them**.**

To begin, the Cartesian coordinate system must be specified in order to express geometrical vectors such as p.



Figure(2. 11) In the Cartesian reference frame, vector p (inertial)[6]**.**

The vector p can be described in one of two ways: geometrically, as in the following:

$$p = p_x^0 i + p_y^0 j + p_z^0 k \quad (2.5)$$

Alternatively, an algebraic description may be used:

$$p = \begin{bmatrix} p_x^0 \\ p_y^0 \\ p_z^0 \end{bmatrix}, i^0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, j^0 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, k^0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.6)$$

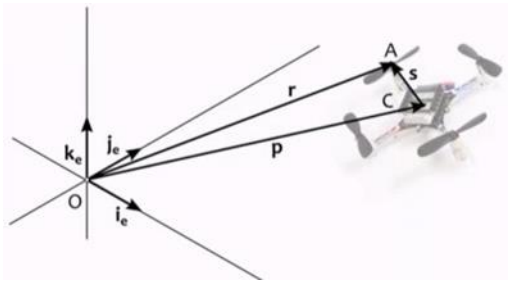
It may be feasible to construct an inertial Cartesian reference frame centered at O, dubbed e-frame (for "external"), and the points A and C, respectively, in the propeller center and the drone's center of mass, which should roughly overlap with the drone frame's center**.**

Three different location vectors might be used:

- C's relative to O's position, p
- A's relative to O's position, r
- A's relative to C's position, s

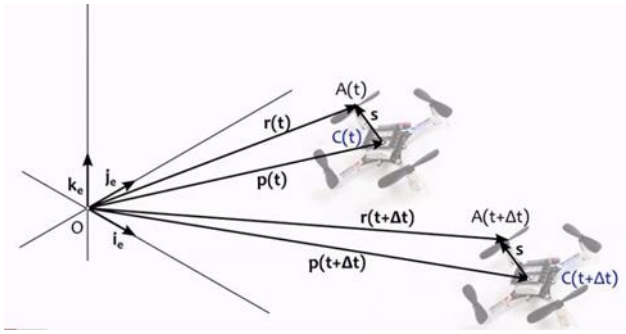
It is also feasible to demonstrate the following connection between these vectors:

$$r = p + s \quad (2.7)$$



Figure(2. 12) Position vectors of the quadrotor in the inertial frame[6]

The position vectors r and p will vary in a translational motion, but not the vector s, which will remain constant.



Figure(2. 13) a quadrotor's translation[6]**.**

The quadrotor's velocity and acceleration are the time derivative of vector p:

$$v_p(t) = \dot{p}(t) = \frac{dp(t)}{dt} \tag{2.8}$$

$$a_p(t) = \dot{v}_p(t) = \ddot{p}(t) = \frac{dv(t)}{t} = \frac{d^2p(t)}{t} \tag{2.9}$$

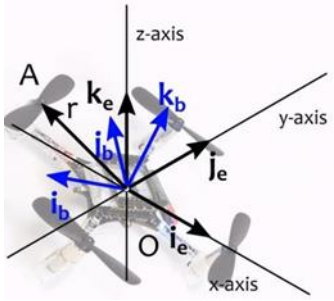
It will be fascinating to observe how the vector r evolves over time:

$$vr = \dot{r} = \dot{p} + \dot{s} = \dot{p} = vp, vr = vp \tag{2.10}$$

$$ar = \ddot{r} = \dot{v}r = \dot{v}p = ap\,, ar = ap \tag{2.11}$$

During a translation movement, all points of the moving item have the same velocity and acceleration; in this example, the drone. In the case of a circular motion,

however, p(t) = 0 can be observed. and the position vector r described by OA, which corresponds to a fixed point A on the body frame that might be the propeller center, and a body reference frame, "b-frame," that rotates with the quadrotor along an axis running through its center of mass.



Figure(3. 1) Body reference frame and inertial frame of quadrotor[30]**.**

The purpose is to express vector r's speed and acceleration. The b-frame can be represented by the following notation in relation to the e-frame basis vectors:

$$i_b = i_b^e\, i_e + i_b^e\, j_e + i_b^e\, k_e \tag{2.12}$$

$$j_b = j_b^e\, i_e + j_b^e\, j_e + j_b^e\, k_e \tag{2.13}$$

$$k_b = k_b^e\, i_e + k_b^e\, j_e + k_b^e\, k_e \tag{2.14}$$

Vector r can be expressed in the b-frame.

$$r = r_x^b\, i_b + r_y^b\, j_b + r_z^b\, k_b \tag{2.15}$$

In the e-frame, this equation is referred to as:

$$r_e = r_x^b\, i_b^e + r_y^b\, j_b^e + r_z^b\, k_b^e \tag{2.16}$$

The vector r can be represented in either b-frame or e-frame, and the transition between both is accomplished by multiplying a rotation matrix R:

$$r_e = R_b^e\, r_b \tag{2.17}$$

Where

$$r_e = \begin{bmatrix} r_x^e \\ r_y^e \\ r_z^e \end{bmatrix}, r_b = \begin{bmatrix} r_x^b \\ r_y^b \\ r_z^b \end{bmatrix}, R_b^e = \begin{bmatrix} i_{b,x}^e & j_{b,x}^e & k_{b,x}^e \\ i_{b,y}^e & j_{b,y}^e & k_{b,y}^e \\ i_{b,z}^e & j_{b,z}^e & k_{b,z}^e \end{bmatrix} \quad (2.18)$$

The rotation matrix has nine entries, however only three are necessary to express a rotation/orientation in three dimensions. The Unit Quaternions (also known as "Euler parameters") and the Euler angles are two alternatives for describing orientation in a more compact manner than the Rotation matrix.

Unit Quaternions are a group of four numbers (4D vectors). They are the most economical and numerically reliable tool for expressing orientations and rotations, and they are identical to rotation matrices. Quaternions are 4-dimensional vectors formed by combining a 3-dimensional vector with a 2-dimensionalvectorscalar

$$q = \begin{bmatrix} s \\ v \end{bmatrix} = [s \quad v_1 \quad v_2 \quad v_3]^T \quad (2.19)$$

A rotation matrix may be expressed as a function of a quaternion in the following way:

$$r_e = [-v \quad sI3 + [v]_x] \begin{bmatrix} -v^T \\ sI3 + [v]_x \end{bmatrix} r_b = R_e^b(q) r_b \quad (2.20)$$

$$R_e^b(q) r_b = \begin{bmatrix} s^2 + v_1^2 - v_2^2 - v_3^2 & 2v_1v_2 - 2v_3s & 2v_1v_3 + 2v_2s \\ 2v_1v_2 + 2v_3s & s^2 - v_1^2 + v_2^2 - v_3^2 & -2v_1s + 2v_3v_2 \\ 2v_1v_2 + 2v_2s & 2v_1s + 2v_3v_2 & s^2 - v_1^2 - v_2^2 + v_3^2 \end{bmatrix} \quad (2.21)$$

$$R_e^b(q) r_b = 2 \begin{bmatrix} s^2 + v_1^2 - 0.5 & v_1v_2 - v_3s & v_1v_3 + v_2s \\ v_1v_2 + v_3s & s^2 + v_2^2 - 0.5 & -v_1s + v_3v_2 \\ v_1v_2 + v_2s & v_1s + v_3v_2 & s^2 + v_3^2 - 0.5 \end{bmatrix} \quad (2.22)$$

$$\dot{r} = 0.5 \begin{bmatrix} -v^T \\ sI3 + [v]_x \end{bmatrix} \omega_b = 0.5 \begin{bmatrix} -v_1\omega_x^b - v_2\omega_y^b - v_3\omega_z^b \\ s\omega_x^b - v_3\omega_y^b + v_2\omega_z^b \\ -v_2\omega_x^b + v_1\omega_y^b + s\omega_z^b \end{bmatrix} \quad (2.23)$$

Another method for representing orientation is to use Euler Angles. They are a set of three numbers that is equivalent to a rotation matrix, but they have an inherent disadvantage in that they contain singularities in operation (divisions by zero in expressions), which must be addressed as exceptional situations.

The frame's rotation may be broken down into three distinct rotations around the coordinate system axes, which is a product of three 2D rotations.

$$R_b^e = R_z(\phi)R_y(\theta)R_x(\psi) \tag{2.24}$$

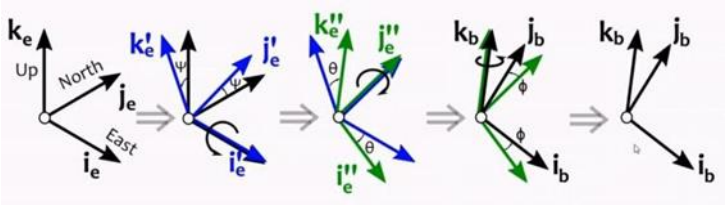
$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ \sin\theta & 0 & \cos\theta \\ 0 & 1 & 0 \end{bmatrix},$$

$$R_x(\psi) = \begin{bmatrix} 0 & cos\psi & -sin\psi \\ 0 & sin\psi & cos\psi \\ 1 & 0 & 0 \end{bmatrix} \tag{2.25}$$

The following is an expression for the relationship between the rotation matrix and the Euler angles:

$$R_b^e = \begin{bmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\psi\sin\theta\cos\phi - \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi - \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix} \tag{2.26}$$

ψ Are the roll angle, $\theta$the yaw angle, $\phi$and the pitch angle, and Cardan angles are another name for this intrinsic sequence.



Figure(2. 14) Angles and rotations defined by Euler[32]

There are a total of 12 extrinsic and 12 intrinsic combinations.

In terms of calculating velocity and acceleration for rotational motion, it can be shown that r b is constant (since the b-frame rotates with the drone) and re varies with time. If a rotation matrix's derivative is employed, the linear velocity in the e-frame is represented as:

$$V_e(t) = \dot{R}_e^b(t)\, r_b(t) \tag{2.27}$$

Instead, the rotation matrix's derivative can be written as:

$$\dot{R}_e^b(t) = \Omega_e R_e^b(t) = R_e^b(t)\Omega_e \tag{2.28}$$

Matrices are skew-symmetric and have the following shape:

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \ , \ \Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.29}$$

Another technique to describe the vector cross-product is to multiply it by a skew-symmetric matrix. The angular velocity, which is a physical vector, is represented by the vector.

The rotation matrix's derivative is written as:

$$\dot{R}_b^e(t) = [\omega_e]_x R_b^e(t) = R_b^e(t)[\omega_b]_x \qquad \text{with} \quad \omega_e = R_b^e(t)\omega_b \tag{2.30}$$

The angular rate in the spinning body frame ω_b may be calculated using data from a MEMS gyroscope sensor mounted on the b-frame.

The linear velocity vector is denoted by the notation:

$$v_e = [\omega_e]_x r_e \tag{2.31}$$

The derivative of the angular velocity vector is the angular accelerationα, and its unit is [rad/s].

The e-frame linear acceleration a_e may be expressed as:

$$a_e = [\alpha_e]_x \, r_e = [\omega_e]_x [\omega_e]_x r_e \tag{2.32}$$

Instead, a_e is the quadrotor's accelerometer sensor output:

$$a_e = R_e^b a_b \tag{2.33}$$

The two motions are merged in a roto-translational motion, which are the generalized case and the most relevant in the drone application.

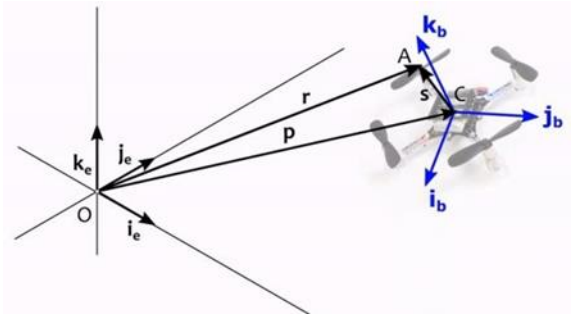


Figure (2. 15) Body reference frame and inertial frame of a quadrotor.

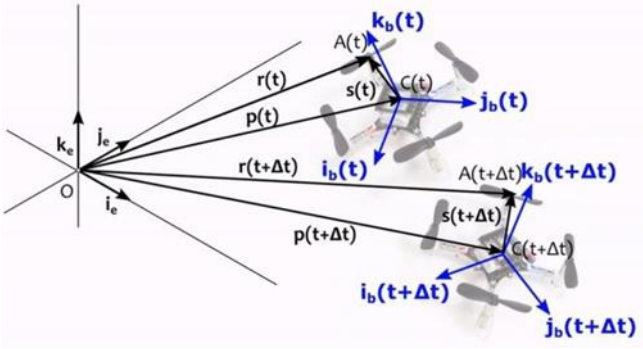


Figure (2. 16) Quadrotor roto translation

Combining translation, rotation, and translation motion yields the following position vector:

$$r_e = p_e + s_e = p_e + R_b^e s_b \tag{2.34}$$

Furthermore, the linear velocity of point r relative to the e-frame is written as

$$v_r^e = v_b^e + R_b^e [\omega_b]_x S_b = v_b^e + R_b^e [\omega_e]_x S_e \tag{2.35}$$

Deriving the linear velocity yields the linear acceleration of point r:

$$a_r^e = a_b^e + [\alpha_b]_x S_e + [\omega_e]_x [\omega_e]_x S_e \tag{2.36}$$

2.2.3 Dynamics

The following rules are used to derive the equations defining the dynamics of quadrotors.

• Newton's rules of motion apply to point masses/particles.

• Euler's rules of rigid body motion (translational and rotational equation of motion) Among the first is the Second Law, which states that the total of forces on a particle object equals the mass of the object multiplied by the acceleration of the object in an inertial reference frame (in this project, the supposition is that the earth is an inertial frame)[28].

$$f_{total} = \sum_k f_k^e = ma_e = m\ddot{p}_e = \sum_k \begin{bmatrix} f_{k,x}^e \\ f_{k,y}^e \\ f_{k,z}^e \end{bmatrix} = m \begin{bmatrix} a_x^e \\ a_y^e \\ a_z^e \end{bmatrix} = m \begin{bmatrix} \ddot{p}_x^e \\ \ddot{p}_y^e \\ \ddot{p}_z^e \end{bmatrix} \quad (2.37)$$

Instead, when a particle provides a force to a second particle by any type of interaction (contact, or at-a-distance), the second particle concurrently applies a force equal in magnitude but opposite to the first particle.

The two forces are directed down the straight line connecting the point masses. If i and j are two particles, and fij is the particle with which i interact with particle j, ri and rj are position vectors, then:

$$f_{ij} = -f_{ji} \quad (2.38)$$

$$f_{ij} = \pm|f_{ji}|(r_i - r_j) \quad (2.39)$$

If a system of particles with fixed distances between them is considered, that is, a rigid body, the dynamical behavior of this system is described by Euler's equations of motion. As a result, the resultant of the internal forces imparted to these particles is null, and the only nonzero forces are those that may be regarded to operate on the object's center of gravity. The sum of external forces equals the acceleration of the center of gravity. Multiplied by the system's total mass[29].

$$p = \frac{1}{m}\sum_i mi\ ri \quad (2.40)$$

$$f_{ext,total}^e = ma_p^e = m\ddot{p}_e \quad (2.41)$$

Here, r is the gravity position vector's center, and m is the body mass. Forces tend to spin a rigid body as well as push or pull it (translation), and torque describes this effect. Multiplied by the system's total mass.

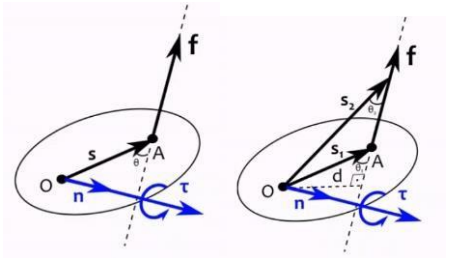


Figure (2. 17) Torque definition

$$\tau_0 = [S]_x f = |S||f|\sin\theta n = |f|.\,d.\,n \qquad (2.42)$$

When a quadrotor is modeled as shown in the diagram, the external torque around O may be stated as follows:

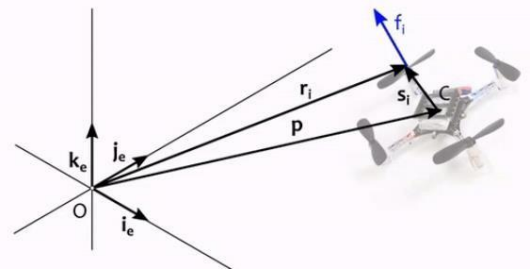


Figure (2. 18) External force exerted on the quadrotor.

$$\tau_{0,ext} = [p]_x f_{ext,total} + \tau_{c.ext} \qquad (2.43)$$

Furthermore, the torque expression, like the resultant of forces on a rigid body, contains just the exterior component because the sum of internal torque is zero.

When the last equation is combined with (2.41), the following equation is obtained:

$$\tau_0^e = \sum_i m_i [r_i^e]_x a_i^e \qquad (2.44)$$

$$\tau_0^e - [p]_x f_{ext,total} = J_e \alpha_e + [\omega_e]_x J_e \omega_e \qquad (2.45)$$

When we combine equations (2.43) and (2.45), we can see that the resultant torque about C (the center of mass of the quadrotor) is a sum of two components, as shown by the following expression:

$$\tau_c^e = J_e\alpha_e + [\omega_e]_x J_e\omega_e \tag{2.46}$$

Where

$$J_e = \sum_i m_i [s_i^e]_x [s_i^e]_x \tag{2.47}$$

That is the matrix of global inertia.

In body-frame coordinates, the rotating equation of motion may be expressed as follows:

$$\tau_c^b = J_b\alpha_b + [\omega_b]_x J_b\omega_b \tag{2.48}$$

In addition, the global inertia matrix is expressed with respect to the body inertia matrix as follows:

$$J_b = R_e^b J_e R_b^e \tag{2.49}$$

$$J_b = -\sum_i m_i [s_i^b]_x [s_i^b]_x = -\sum_i m_i \begin{bmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{bmatrix} \begin{bmatrix} 0 & -z_i & y_i \\ z_i & 0 & -x_i \\ -y_i & x_i & 0 \end{bmatrix} \text{ with } S_i^b = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \tag{2.50}$$

$$J_b = \sum_i m_i \begin{bmatrix} y_i^2 + z_i^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & y_i^2 + z_i^2 & -y_i z_i \\ -x_i z_i & -x_i z_i & x_i^2 + y_i^2 \end{bmatrix} \tag{2.51}$$

$$J_b = \sum_i m_i \begin{bmatrix} \int_v y_i^2 + z_i^2 dm & \int_v -x_i y_i dm & \int_v -x_i z_i dm \\ \int_v -x_i y_i dm & \int_v y_i^2 + z_i^2 dm & \int_v -y_i z_i dm \\ \int_v -x_i z_i dm & \int_v -x_i z_i dm & \int_v x_i^2 + y_i^2 dm \end{bmatrix} \tag{2.52}$$

Furthermore, whereas S_e^b is changeable in time, the vector S_i^b is constant, implying that Je is time-dependent, but Jb in the body frame is time constant.

The differential equations of motion may be summarized and combined to construct the quadrotor's dynamical model, and they can be represented in either rotation matrix notation:

$$\dot{p_e} = v_e \tag{2.53}$$

$$\dot{v_e} = \frac{1}{m} f^e_{ext,total} = \frac{1}{m} R^e_b f^b_{ext,total} \tag{2.54}$$

$$\dot{R^e_b} = R^e_b [\omega_b]_x \tag{2.55a}$$

$$\dot{\omega_b} = (J_b)^{-1}(-[\omega_b]_x J_b \omega_b + \tau^b_c) \tag{2.56}$$

Alternatively, using the Unit Quaternion notation, replace the equation (2.55a) with the following:

$$\dot{q} = \begin{bmatrix} \dot{S} \\ \dot{v} \end{bmatrix} = 0.5 \begin{bmatrix} -v^T \\ sI3 + [v]_x \end{bmatrix} \omega_b = 0.5 \begin{bmatrix} -v_1\omega^b_x - v_2\omega^b_y - v_3\omega^b_z \\ s\omega^b_x - v_3\omega^b_y + v_2\omega^b_z \\ -v_2\omega^b_x + v_1\omega^b_y + s\omega^b_z \end{bmatrix} \tag{2.55b}$$

2.2.4 Forces and Torque

The next schematic (figure 2.19) depicts the section of the system that includes motors and propellers, and the method of force production on the propeller from the motor's torque will be discussed in this paragraph.

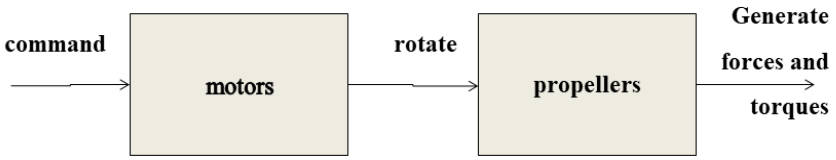


Figure (2. 19) Model of actuators.

The code on the electronics (firmware/embedded code) first provides a digital instruction to the motors, which is then converted into an analog PWM (Pulse Width Modulation) signal that drives the quadrotor's coreless motors. This digital command

is made up of four 16-bit integer integers (ranging from 0 to 65535), one for each motor.

The following are the physical principles that govern the movement of the quadrotor (the same concepts apply to any aircraft):

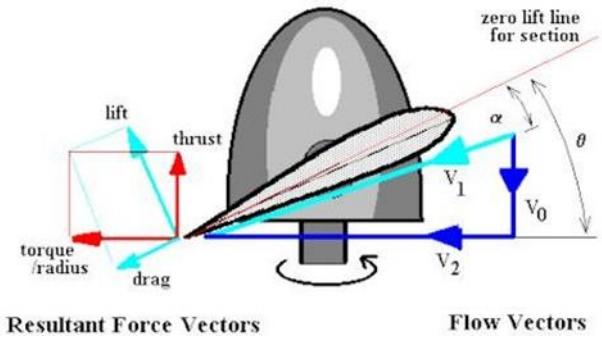
The Theorem of Bernoulli: Daniel Bernoulli discovered that pressure inside a fluid, liquid, or gas falls as the fluid's speed rises, or in other words, "in a moving fluid, the sum of pressure and speed in any place remains constant."

The Venturi effect was discovered by Giovanni Battista Venturi, who demonstrated that as a fluid particle passes through a constriction, its speed rises.

The third Newton's Law states that every action has an equal and opposite response. The idea of lift may be applied to equipment with an airfoil to summarize the preceding principles[30].

A wing or a propeller, for example. It will pass through an airflow, separating it into two flows, if it moves through the air (which has its own atmospheric pressure and speed) at a given speed with a specified upward inclination (called the angle of attack). The one that runs along the top of the profile will do so at a faster rate than the one that runs through the bottom (Venturi effect). Furthermore, according to Bernoulli's theorem, faster speed equals lower pressure. As a result, the upper wing's surface is subjected to less pressure than the inferior wing's.

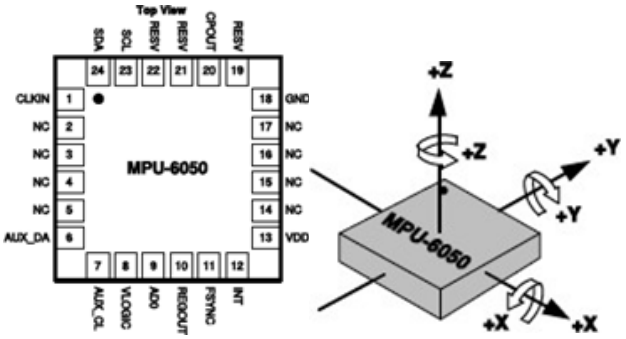
An airfoil is normally made out of a body with a particularly designed shape to capture the majority of the force created by the fluid's speed and pressure fluctuations as it flows through an air stream.



Figure(2. 20) Air propulsion and a propeller model[33]**.**

2.3 The main technical component

• IMU sensors to calculate accelerations and speeds. In mobile robotics, developing a sensory system capable of detecting an object's mobility in its surroundings is a significant challenge. These sensory data are used to construct control algorithms, and their availability and dependability are crucial criteria. An inertial navigation system (INS) is a system capable of obtaining object navigation information (linear and angular locations and speeds) in its body frame and translating it into a fixed (inertial) reference frame **.**



Figure(2. 21) IMU with accelerometer and gyroscope[34]**.**

• Ultrasound sensor In mobile robotics, SONAR (sound navigation and ranging) devices are used to estimate the distance between the robot and the surrounding environment at a minimal cost (in the case of this project, to measure the distance between the drone and the ground). They are both signal generators and transducers, and their functioning concept is based on calculating the flight time of an ultrasonic wave to travel the distance back and forth from the sensor. It is used by the quadcopter to determine altitude. It sends out a high-frequency sound wave and detects how long it takes for the wave to reflect off the ground and return to the sensor. The distance between the floor and the drone may be determined using the measured time. It's possible to estimate a maximum altitude of roughly 13 feet**.**

• Pressure sensor it assists the ultrasonic sensor in estimating the drone's altitude. The air pressure lowers as the drone rises in height. The change in altitude may be estimated using the pressure variation**.**

• Camera sensor for speed measurement and horizontal stabilizer.

• The motors they are coreless motors figure(2.22). It is divided into numerous sections**.**

- A permanent magnet is commonly used as a stator**.**

- Rotor: a crown-shaped ring of ferromagnetic plates around which the various components rotate**.**

- Wrapping the copper windings

- Collector: a cylindrical drum that connects the coil terminals to a set of blades**.**

- Graphite-based brushes maintain contact with the collector even when it is spinning, and they supply the motor with its main power.

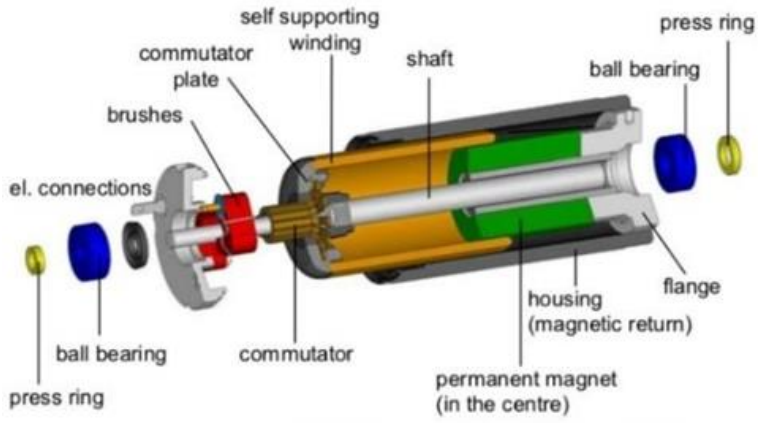


Figure (2. 22) DC motor construction without a core (coreless) [36].

Chapter 3: The Simulation Model Description and Methodology

3.1 Introduction

In this chapter, was described and explained the simulation model. It also designed and analysed the Linear and Nonlinear Controller Systems of altitude in the Parrot mini-drone. Improved the size of the parrot's propeller to make it more capable of bearing the additional weights. So studied this design's effect on the aircraft's control system. also made the aircraft follow a specific path, and calculated the error rate on this path. Used the simulation design model of the Parrot plane from the mini-drone[37].

3.2 The Simulation Model Description

The design is a closed-loop system, with the plant represented by the block Multicopter Model (the dynamic content of the system). The controller is represented by the flight controller system. The sensor block is put on the feedbag line. The command block represents the original signal that is used by the controller to compare it with the feedbag signal. The Environment Model contains constants and mathematical equations. The flight visualization is represented by the output 3D Simulink visualization.

By entering the command window with the phrase "ParrotMinidroneCompetitionStart," the project environment of the Mathworks mini-drone competition conducted by Mathworks may be used. Miniature Drone Simulator (figure 3.1).

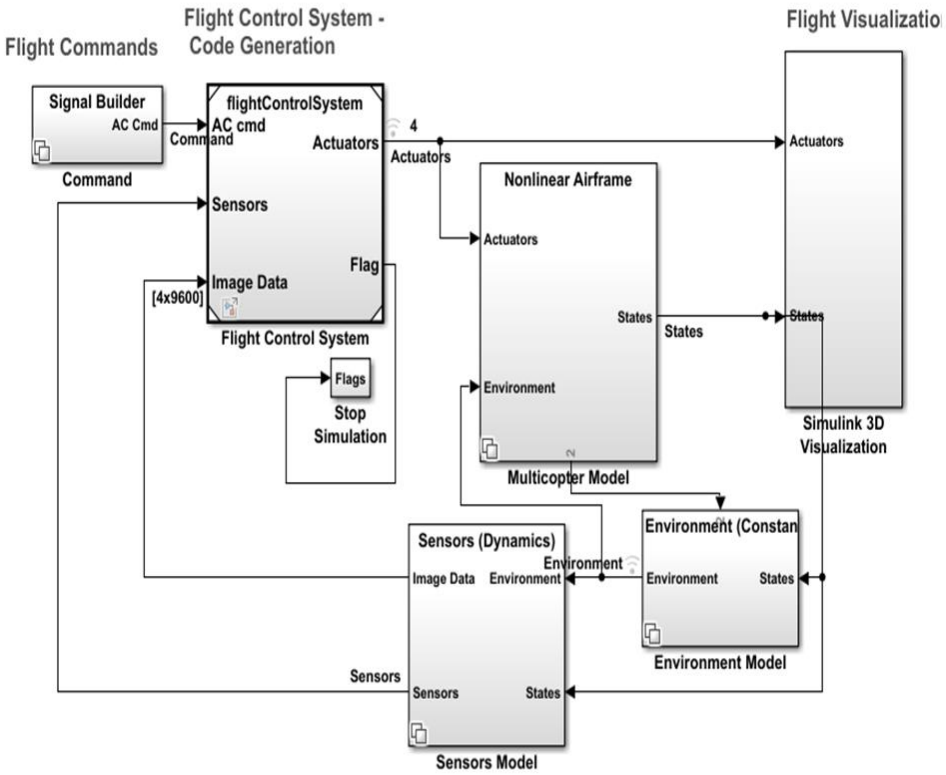


Figure (3.1) Simulation model of parrot mini drone-Mambo

Below, explain the work of each part of the model shown in Figure (3.1) and briefly explain its components.

3.2.1 Flight command

In the simulation, the "Command" subsystem may be used to adjust the micro drone input signals (figure 3.2). This one is made up of four distinct subsystems: data input, signal builder, joystick, and spreadsheet file reading.

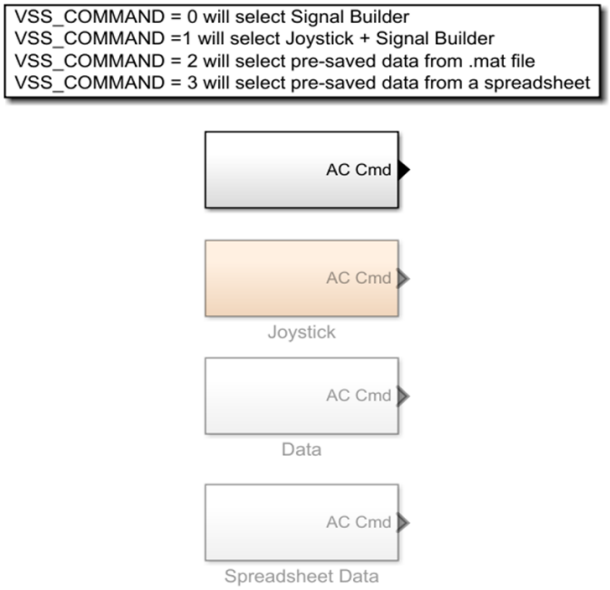


Figure (3.2) Command subsystem

The command subsystem has signal builder (Figure 3.3) set as the default. However, in this project, the signals controlling the drone originate from the path planning subsystem in the Flight controller which gets data from the image processing subsystems.

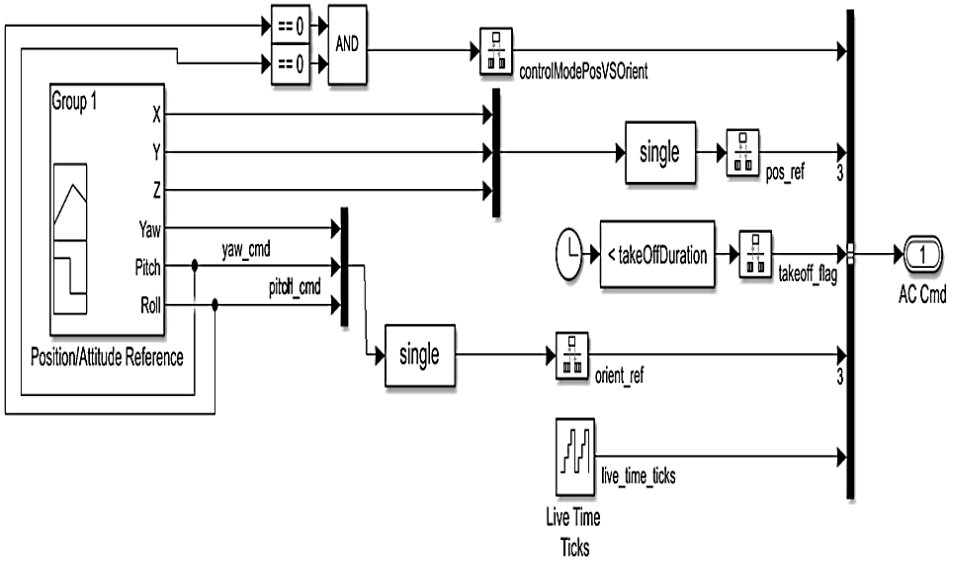


Figure (3.3) Signals builder.

The mode used to control the drone is based on position coordinates X-Y-Z and yaw, but it may also be driven by pitch and roll signals from the image processing subsystem. However, in this final scenario, the camera orientation must be maintained to point directly to the track, which is readily done by employing position coordinates rather than pitch and roll angles. The simulation model is the focus point of the Model-Based Design process since it helps to enhance control development before deploying it on hardware, hence minimizing damages and crashes.

3.2.2 Flight control system

The FCS subsystem is based on a model created at MIT by Professor Sertac Karaman and Fabian Riether. Because the project's purpose is to create a flight controller that can monitor the path while remaining stable, the controller subsystem is designed to tune the gains of the PID controllers, and the image processing and path planning subsystems are designed to achieve that goal**.** There are two subsystems within the Flight Control System (Figure 3.4)

- Image Processing System: the graphic processing component (the green block) is included in this system**.**
- Control System: This is where the flying logic is stored (the orange block).

The two subsystems operate at different speeds: the Image Processing System operates at almost 20 milliseconds (the camera frame rate is 60 Hz), whereas the Control System operates at 5 milliseconds (200 Hz)**.**

As a result, they are linked via the "rate transition" block, which enables data transfer across systems with differing rates**.**

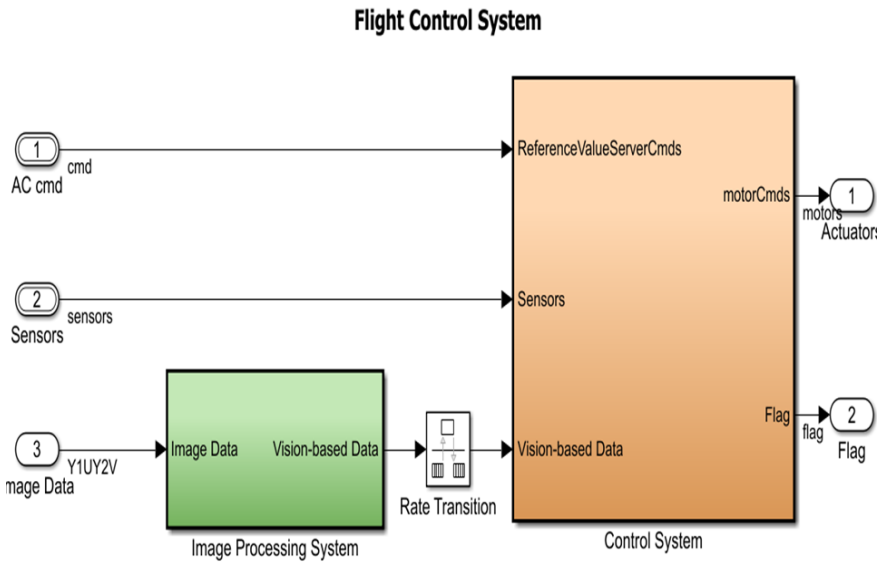


Figure (3.4) the flight control system.

There are several separate blocks inside the Control system subsystem block (Figure 3.5)

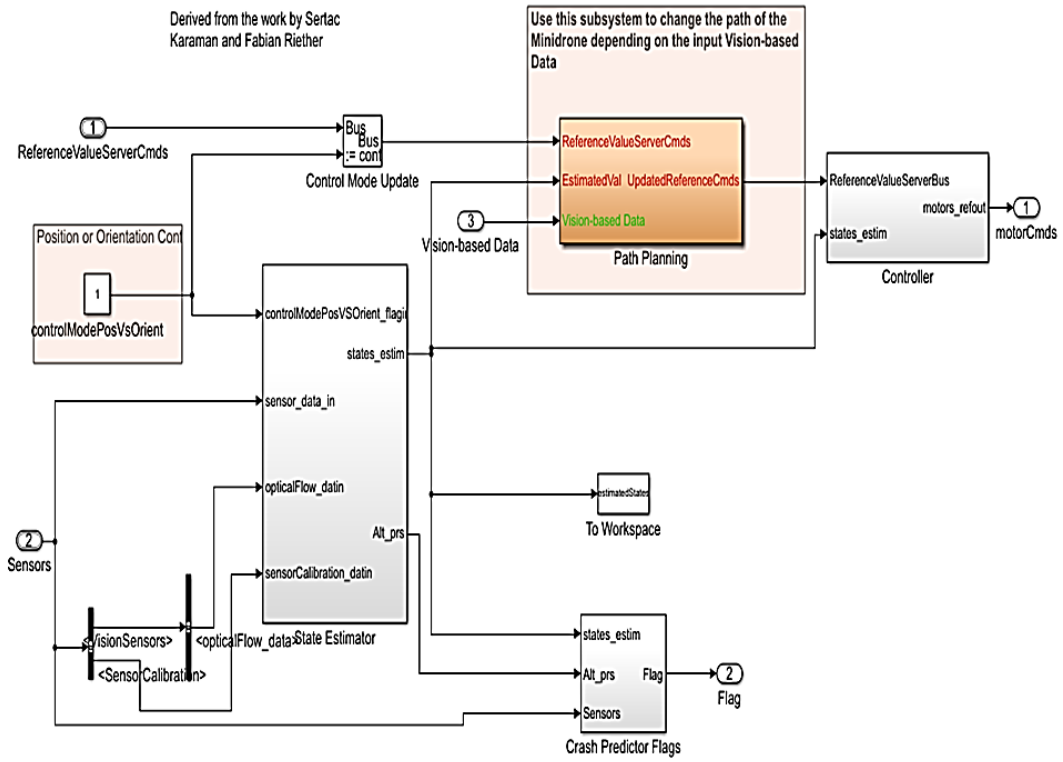


Figure (3.5) Subsystems in flight control system block.

Figure (3.5) Path planning (orange): this is where the logic for the line-tracking method will be constructed**.**

1- Controller: This is where all of the flight controller's PIDs are stored**.** The Flight Control System (FCS) is separated into colour-coded control subsystems and contains the position and orientation control systems as well as take-off logic. Attitude, yaw, roll, and pitch controllers are three low-tier control systems. All four signals are sent into the motor mixing algorithm after the attitude control is processed through the timed take-off control logic. If just orientation control is necessary, the user can apply a logic flag to have the FCL ignore all other systems and only activate these three blocks. If position control is necessary, the position controller activates the upper-tier MMA block, resulting in a cascading system with roll and pitch control in the inner loop and position control in the outer loop. The control logic commands are converted into actuator signals sent out from the FCS by the motor mixing algorithm and thrust to the motor command block.

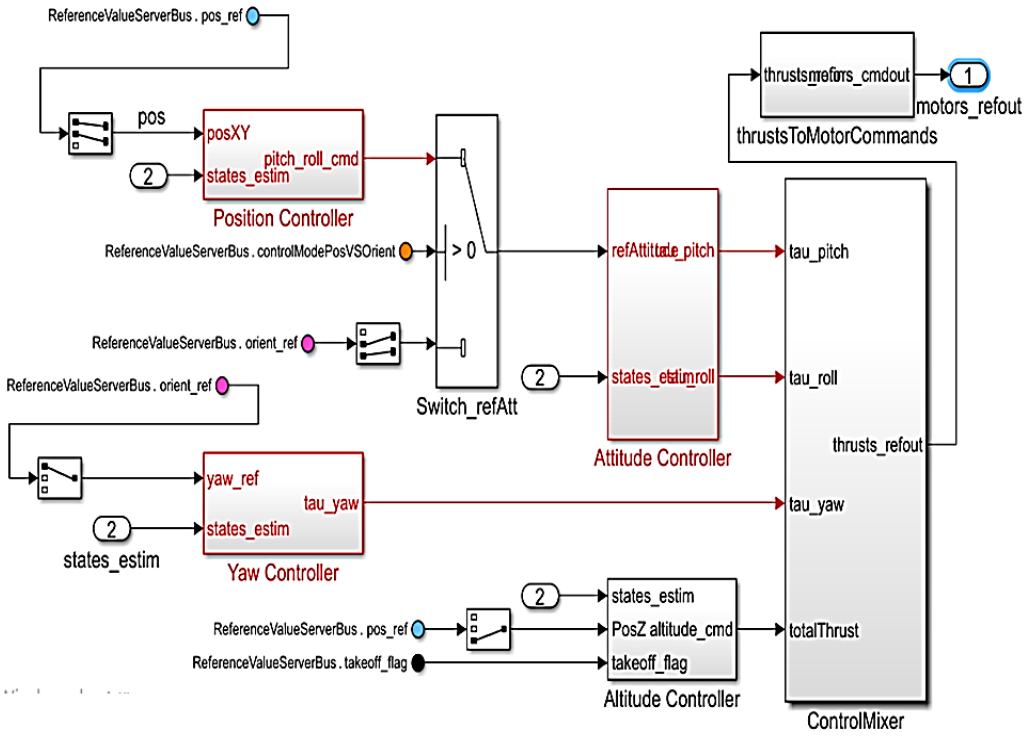


Figure (3.6) The controller subsystem.

2- State estimator: the state observer is included in this class**.** Within the state estimator subsystem, Figure (3.7), a twelve-dimensional state vector for the PMD is produced, consisting of location, orientation, velocity, and angular velocity[x y z vx vy vz $\emptyset$ $\psi\theta$ p q r]. The state estimator block collects data and compiles it using the four PMD sensors.

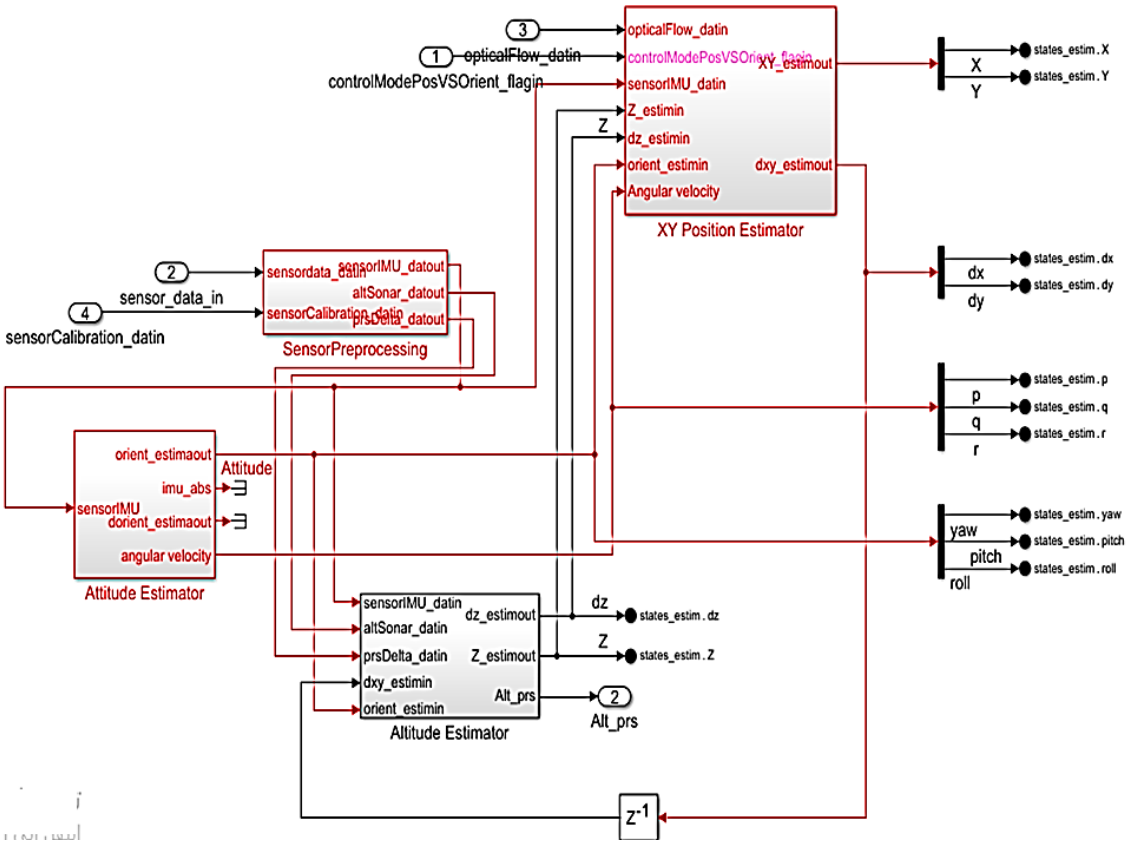


Figure (3.7) original internal Simulation components for the parrot mini drone's flight control state estimation subsystem

Figure (3.8) shows the state estimator, which is made up of two subsystem blocks: a velocity estimator and an XY position estimator.

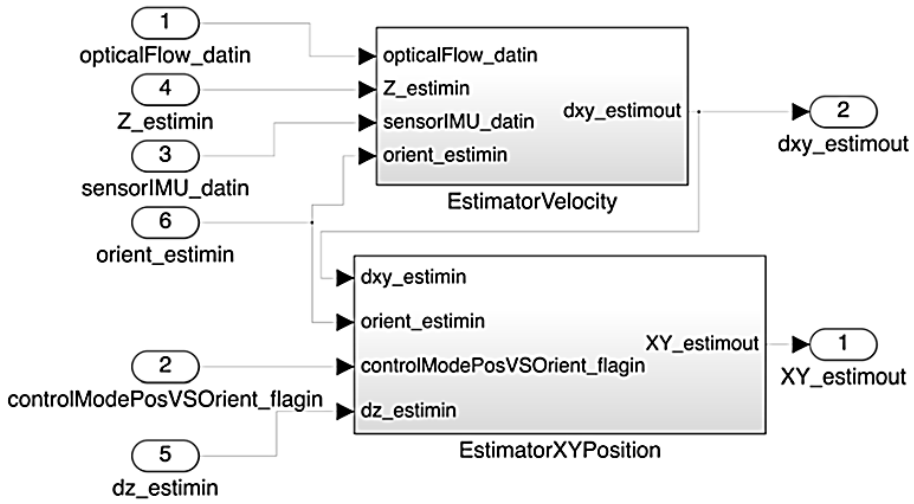


Figure (3.8) The original upper level estimator's simulation subsystem for the parrot mini drone's XY location.

The initial state is intended to prevent the PMD from executing instructions until communication with the reference host has been established. While the PMD is in this state, the system thinks the PMD is in the original XY state of [0;0] and calculates the current position by integrating the velocity in the X and Y locations. The system then advances to the lost or confirmed state [38][36] after determining the state. When the PMD has not received a status update for the X and Y locations, the lost state is activated. When this is activated, the Kalman filter block's prior time step estimate of the previous location is employed, and the system integrates the X and Y velocity to produce the current estimated position in the X and Y. The starting conditions are reset to the falling edge of the status signal from the TCP/IP Receive block. The confirmed state is the one that activates the Kalman filter block. The Kalman filter [36] is based on noise from the plant and measurements. The noise levels are computed using the original SSPPM model. With each estimation, the model is meant to lower the size of the uncertainty bubble[36]. When the system exits the confirmed state and must enter the lost or initial state, the uncertainty bubble shrinks to the size of the lost or initial state. The system utilizes the received position as the measured position and the X and Y velocity as the position commands to reduce the size of the uncertainty bubble. The projected position from the preceding time step from the beginning or lost state is the initial condition for this status.

3- Crash Predictor Flags: this section contains the logic for turning off the drone if there is a flight anomaly.

3.2.3 The multicopter (or airframe) model

In figure (3.9), It is built as a variant subsystem, which means that before running the model, may choose which version of the airframe we want to run with: either the nonlinear model for flight simulation or the linear airframe model for controller tuning. When looking within the nonlinear model, there are two primary blocks that can be seen. The 6DOF block and the AC model subsystem. All internal and external forces and torques operating on the drone are calculated by the AC model block. The quaternion representation for six-degrees-of-freedom equations of motion with regard to the drone's body frame and the earth frame body axes is used. This subsystem's output is a bus system that connects the status signals.

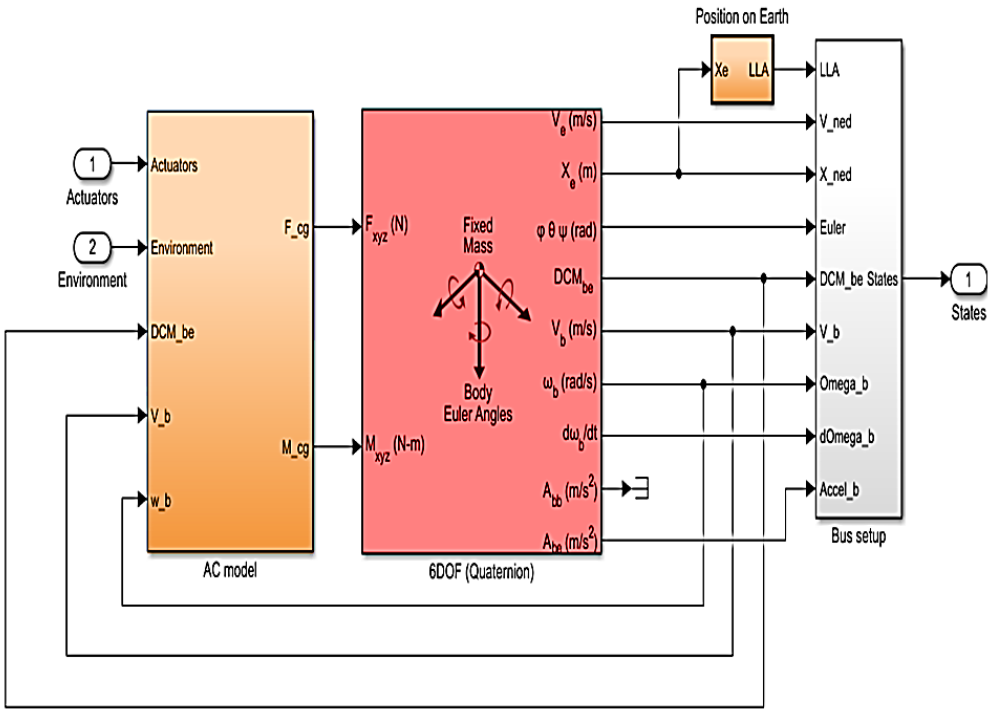
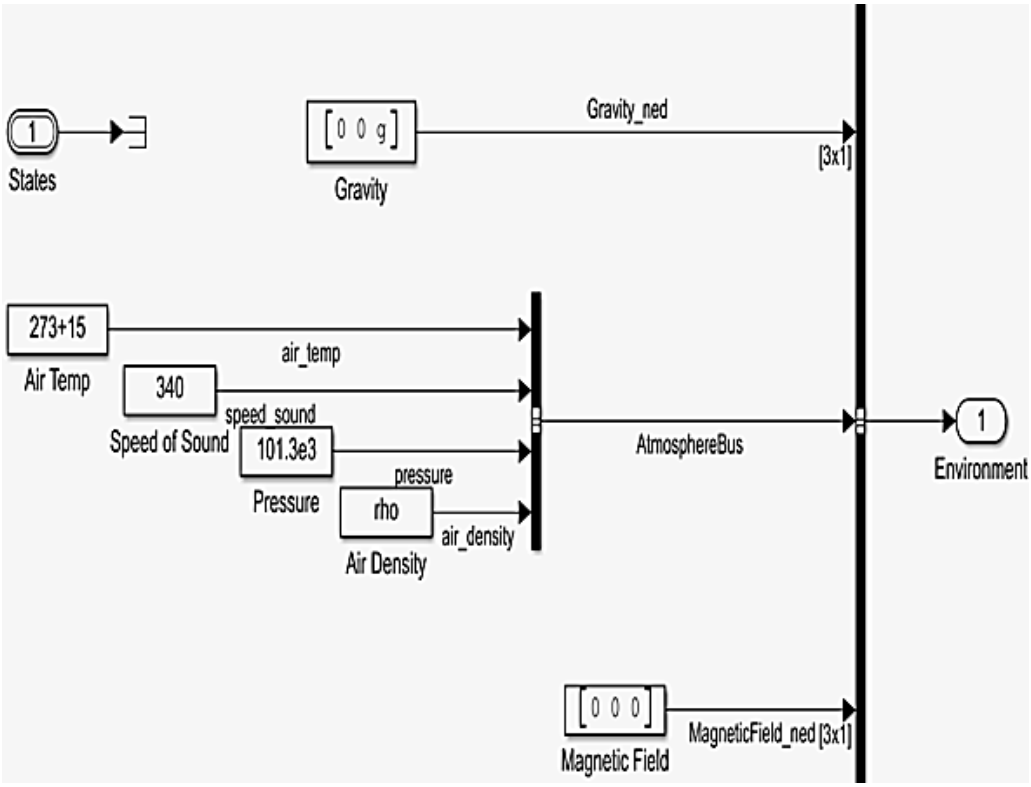


Figure (3.9) Nonlinear quadcopter model.

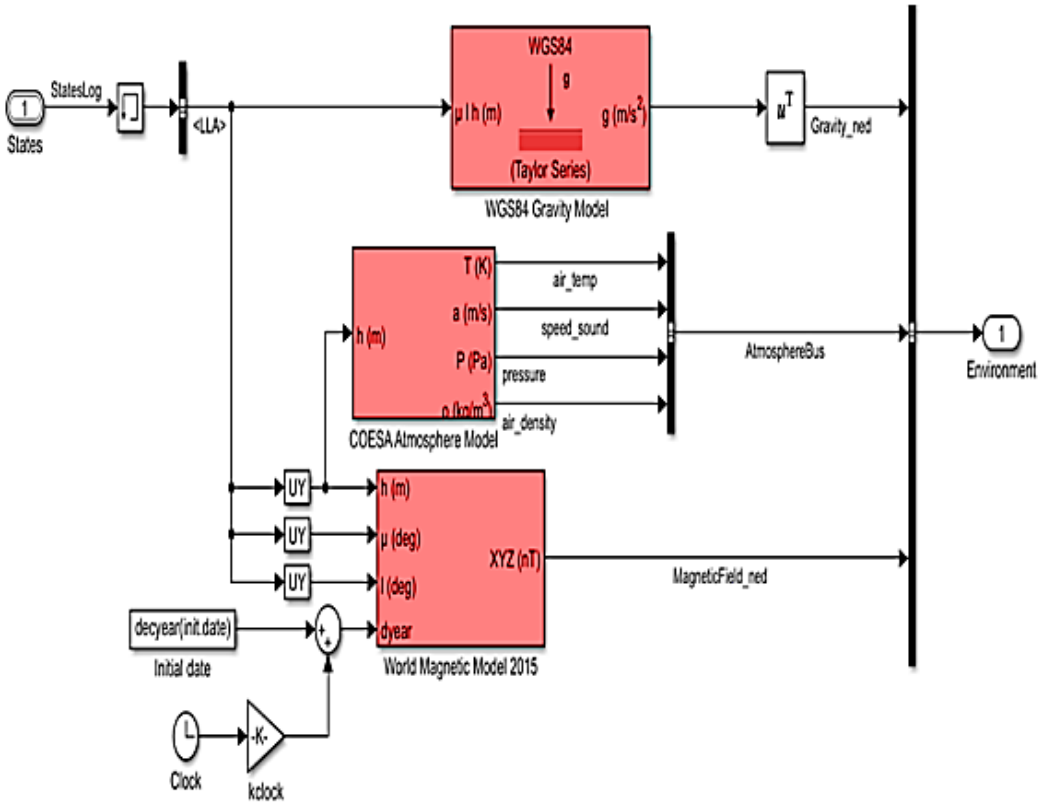
3.2.4 The environment block

The (Figure 3.10) is a variation subsystem that allows us to choose between constant and position-dependent environmental variables. However, because things like gravity and air pressure do not fluctuate by less than a meter between ground level

and altitude, the constant variables will be used for this project. However, if the goal is to see how high the little drone can fly, using the changing environment would drop air pressure and density as it rises, eventually causing the drone to stall at a certain height. So, which model to use depends on the purpose of the test. We'll utilize the Constant Environment model for our needs.



(**a**)



(b)

Figure (3.10) (a)and(b) are Constant environment and Variable environment.

3.2.5 The sensors block

The (Figure 3.11), which is also a variant subsystem, are dynamic sensors with noise or feedthrough sensors. The sensors should be chosen to behave as closely as possible to real-world objects. There is some hardcoded sensor calibration data and a sensor system block that stores the models for the camera, the IMU, the ultrasound, and the pressure sensor inside this subsystem. The condition of the system is determined by the following sensors:

• An IMU (Inertial Measurement Unit) is a device used to measure inertial forces. It calculates angular rates and translational accelerations.

• A camera with a lens angled downward It's required for image processing and estimating optical flow.

• An ultrasound sensor's required for altitude calculations.

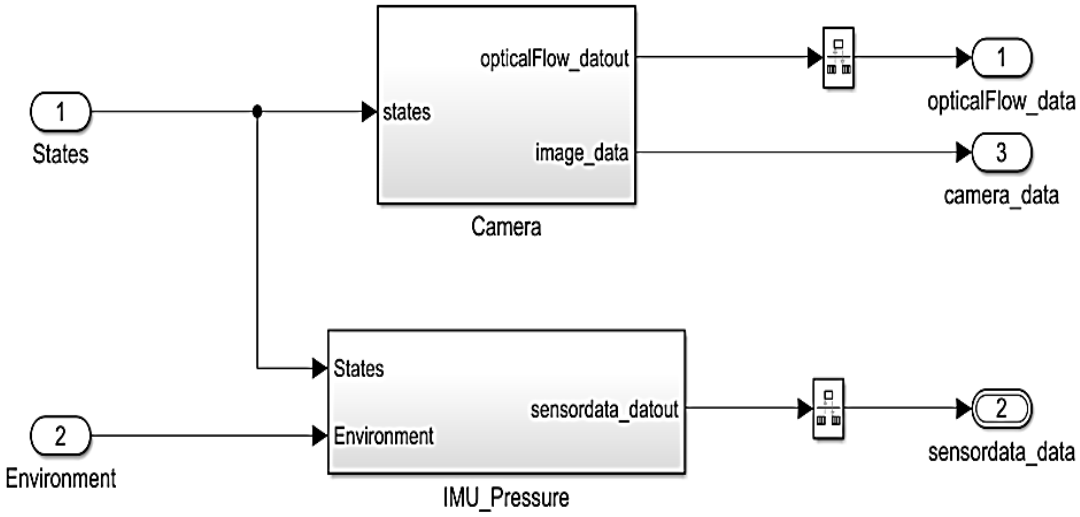


Figure (3.11) Sensor system.

3.2.6 The flight visualization

The signals can be shown on an Aerospace Blockset toolbox-provided cockpit display, which includes typical flying instruments such as heading, percent RPM, airspeed and climb-rate indications, and an altimeter. The suitable states are found in the "extract flight instruments" (figure 3.12) subsystem.

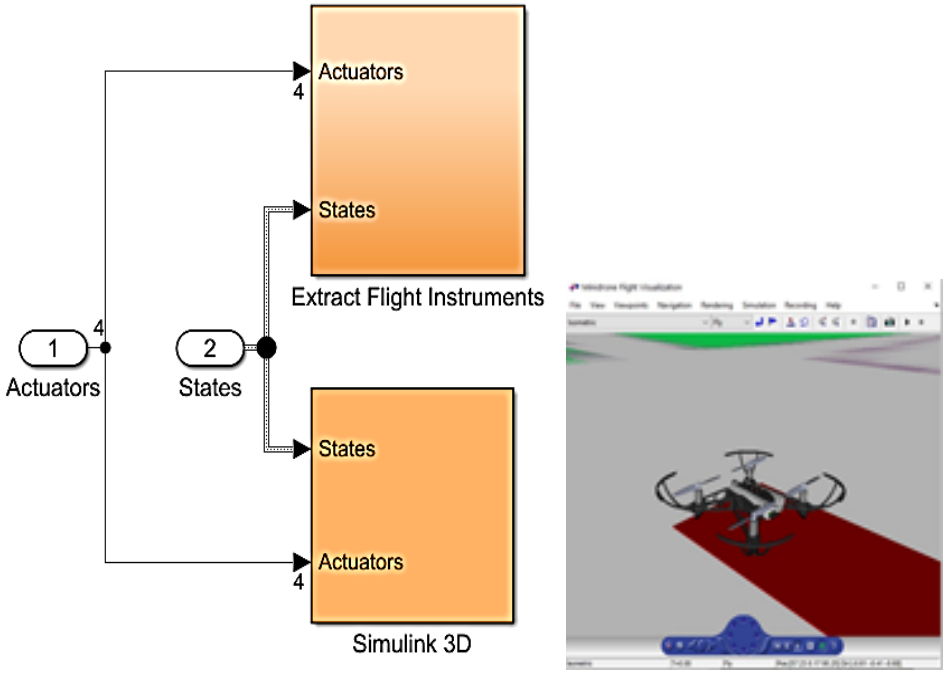


Figure (3.12) the Simulink 3D environment.

3.3 Mini Drone Linear and Nonlinear Controller System Design and Analyzing

In this section, have improved the height controller in the drone, thus improving the altitude controller by using (PD) and tuning the values of (Kp and Kd) in the altitude controller of the Parrot Mini Drone Mambo to make it more bearable to external influence and to maintain its altitude. We assumed that the aircraft was exposed to bad weather conditions, such as snowfall and dust, which led to an increase in the speed at which the drone fell.

3.3.1 Design Altitude controller

Used Parrot mini drone-Mabo, the figure (3.13) The Mambo is controlled by a computer. A built Simulink model is utilized to simulate the desired flight route for this study. This program enables simulated runs with various parameters to identify the Parrot mini drone's intended response. This is performed by controlling the Simulink model's numerous subsystems[27].

Figure (3. 13) Parrot mini-drone fly

Maintaining control of a UAV is necessary for a variety of reasons. UAVs must fewer independent control inputs than grades of freedom, which causes a controller difficulty when tiresome to retain control of wholly six degrees of freedom. This opens up the possibility of including design elements to regulate the axes, as well as yaw, pitch, and roll. Figure (3.14) shows a simple block chart of the needed inputs and wanted outputs that a controller will require to successfully managing a UAV.

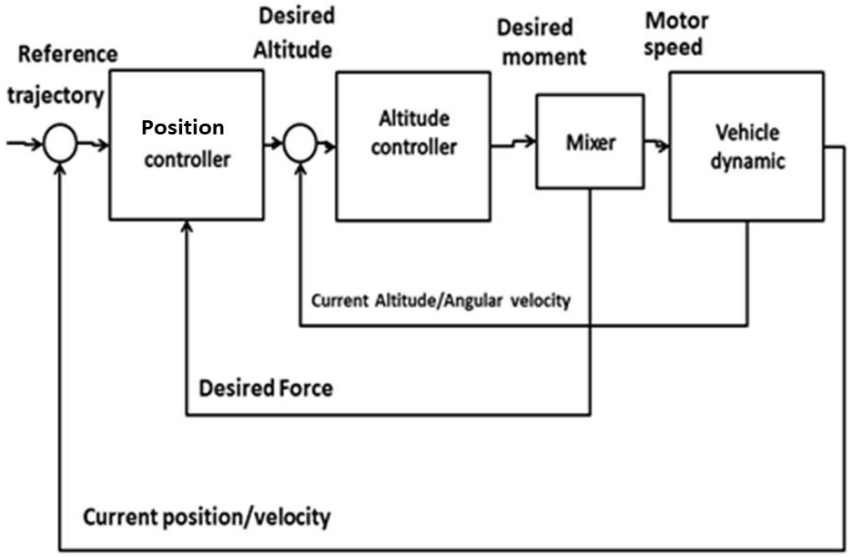
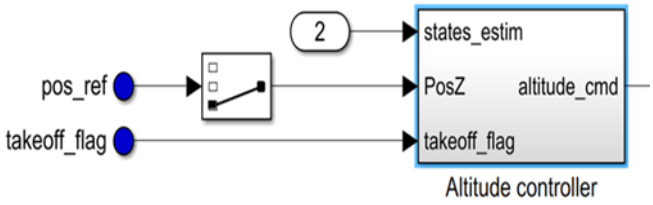


Figure (3.14) Quadrotor control system design.

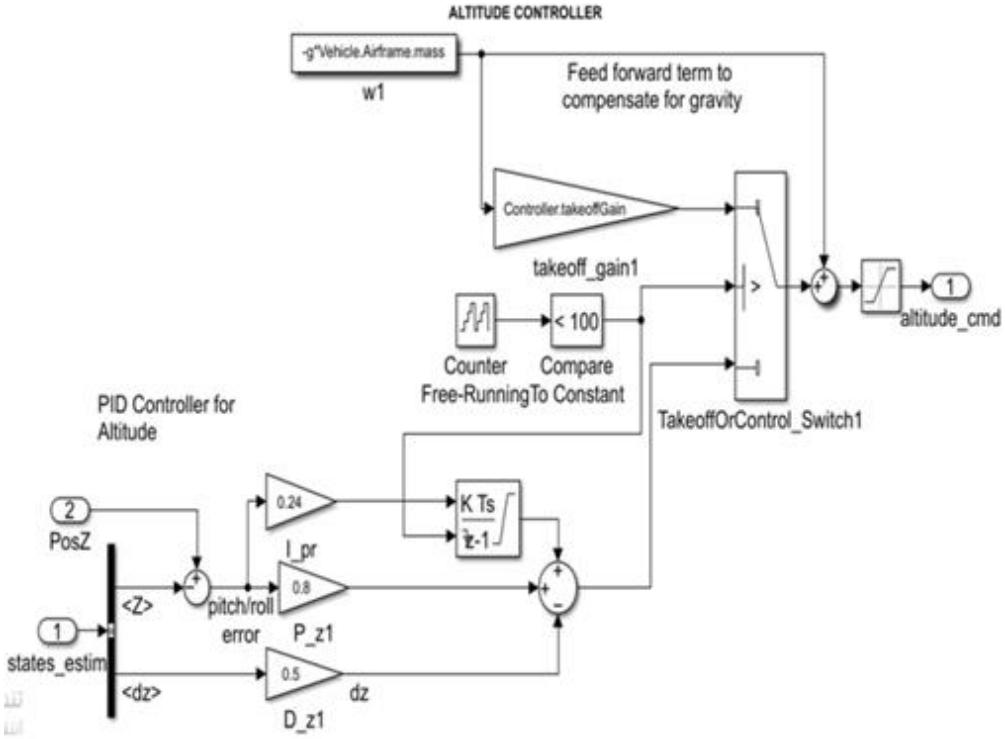
The focus of this study is on the Parrot mini drone-Mabo control implementation. Simulink is used to create the programmed controller based on a Parrot model. The block diagram for the procedure for each flight alteration made by the Parrot mini drone Mabo is shown in Figure (3.5). Two control rings, an external loop and an internal loop flow continually into apiece throughout the system. The system inputs are the location reference, estimated yaw, yaw reference, and altitude reference. The Simulink simulation's state estimator is divided hooked on numerous filter blocks. A complementary filter and a Kalman filter remain employed[27].

To identify inaccuracies, it compares the reference signals generated by the path planning algorithm to the estimated states. These are fed into the PID controllers, which generate the commands for the actuators. The signals are then sent to the pitch/roll (or attitude) internal loop controller by the X-Y position outer loop controller. There is also a yaw controller and a height controller that work independently of these controllers. A total of six PID controllers control the position and attitude of the micro drone. Figure (3.15) and Figure (3.16) illustrate how to set up the altitude controller as PID. In this approach, the proportional gain is multiplied by the altitude error generated from the sonar sensor, while the derivative gain is multiplied by the rate of altitude of the gyroscope, which is a less noisy signal than

the ultrasound signals. It is important to note that the z-axis in the coordinate system of drone points downwards, which means the altitude value in the control system will always have a negative sign in front of it (expressed in meters)[6].



Figure(3.15) Altitude controller block[39]**.**



Figure(3.16) Altitude controller structure[40]**.**

The Mathematical Model of PID (Proportional Integrated and Derivative) and PD (Proportional and Integrated) controller:

3.3.2 PD controller

It is series controller, proportional and derivative controller. If we assume that we have the system shown in Figure (3.17), the PD controller is connected in series with the system.

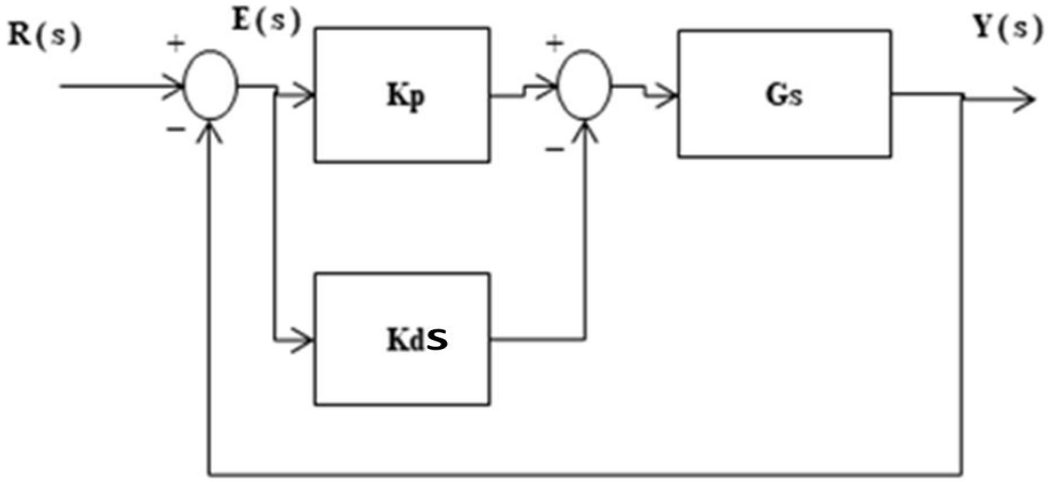


Figure (3.17) PD Controller block diagram.

$$\text{oversG(s)} = \frac{w_n^2}{s^2 + 2\zeta Wns} \quad (3.57)$$

Where G(s) is the transfer function of the system, ζ is the damping ratio and wn is the natural frequency.

$$Gc(s) = K_{p+}K_d\, s \quad (3.58)$$

Where Gc(s) is the transfer function of the controller Kp and Kd are constant value (gain).

$$GT(s) = Gc(s)G(s) = \frac{w_n^2\, Kds + w_n^2 Kp}{s^2 + 2\zeta w_n\, s} \quad (3.59)$$

Where GT(s) is the total transfer function.

$$\frac{Y(s)}{R(s)} = \frac{Gc(s)G(s)}{1 + Gc(s)G(s)} = \frac{w_n^2 Kds + w_n^2 Kp}{s^2 + (2\zeta w_n + w_n^2 Kd)s + w_n^2 Kp} = \frac{w_n^2}{s^2 + 2\zeta wns + w_n^2} \quad (3.60)$$

Where the Y(s) is the output signal and the R(s) is the input signal.

To control the damping coefficient and natural frequency, two variables (Kp and Kd) were selected in two equations to allow for full control over the system[29].

3.3.3 PID Controller

It is a cascade controller, proportional, integrated, and derivative controller. If we assume that we have the system shown in Figure (3.18), the PID controller is connected in series with the system.

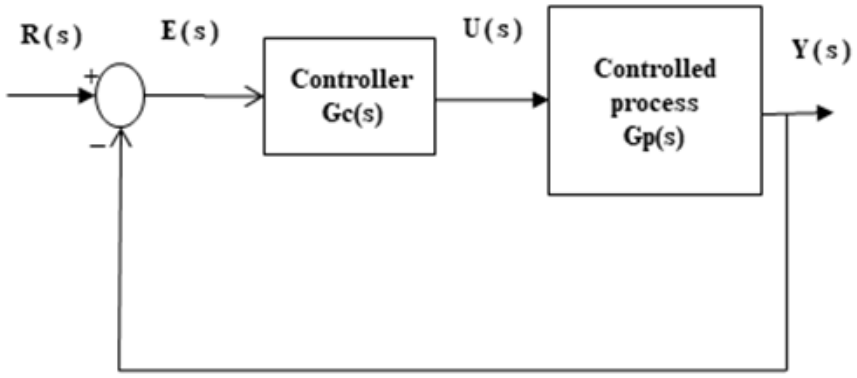


Figure (3.18) PID Controller block diagram.

$$G(s) = \frac{w_n^2}{s^{2+2\zeta w_n} s} \quad (3.61)$$

$$Gc(s) = K_{p+} K_d s + \frac{Ki}{s} \quad (3.62)$$

$$GT = Gc(s)G(s) = \frac{w_{nKds^2+}^2 w_n^2 Kp\, s + w_n^2 Ki}{s^3 + 2\zeta w_{n^s}\, s^2} \quad (3.63)$$

$$\frac{Y(s)}{R(s)} = \frac{Gc(s)G(s)}{1 + Gc(s)G(s)} = \frac{w_n^2 Kds^2 + w_n^2 Kps + w_n^2 Ki}{s^3 + (2\zeta w_n + w_n^2 Kd)s^2 + w_n^2 Kp}$$
$$= \frac{w_n^2}{s^3 + (a + 2\zeta wn)s^2 + (2a\zeta w_n + w_n^2)s + aw_n^2} \quad (3.64)$$

To control the damping coefficient and natural frequency. Three variables are to be selected (Kp, Kd, Ki) in three equations, thus we have full control over the system[29].

3.3.4 The method that used in the design of altitude controller

In this chapter, we developed and improved the altitude control structure shown in Figures (3.15) and (3.16).These PID controllers contain force and torque commands as outputs, which are then communicated to the mix motor algorithm (MMA) (Figure(3.5)), which generates the required motor thrusts and converts the orders into motor speeds. We replaced the PID controller with a type PI controller and then a type controller PD in the first step and found the best of the three types in system stability. The controller was simplified, as figure (3.19). The " auto-tuning " method found the value of KP and Kd were found by the "auto-tuning" method.

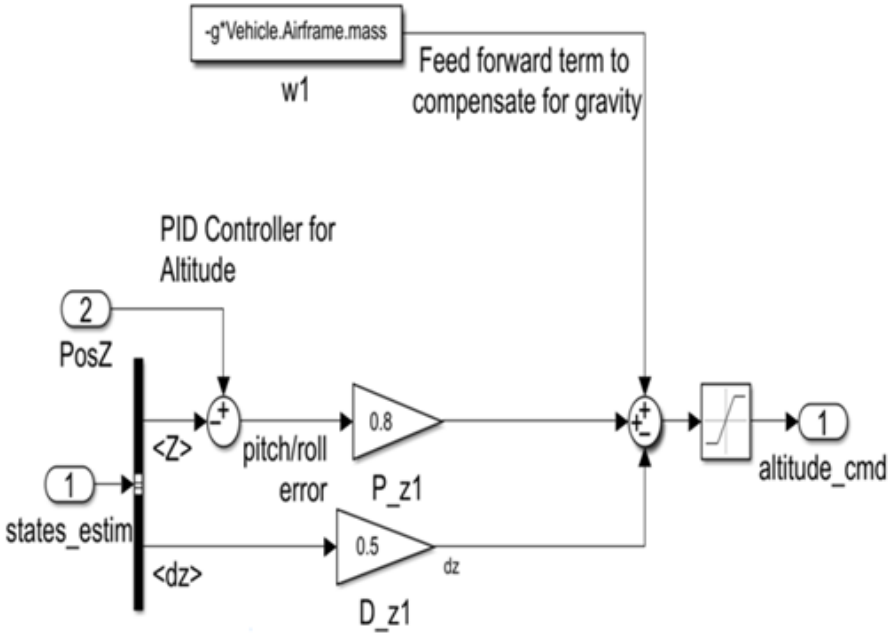


Figure (3.19) Altitude controller.

The second method to increase the efficiency of the system was to apply a disturbance such as dust or snow on the vehicle, thereby increasing the value of the block named (g*vehicle*mass) representing the mass of the vehicle. In doing so, the system became unstable, which led us to tune the value of Kp and Kd until we could improve performance through trial and error. We eventually obtained a good result as the system was able to remain stable even as it was affected by the bad weather. Figure (3.20) summarizes the steps of the work that was undertaken.

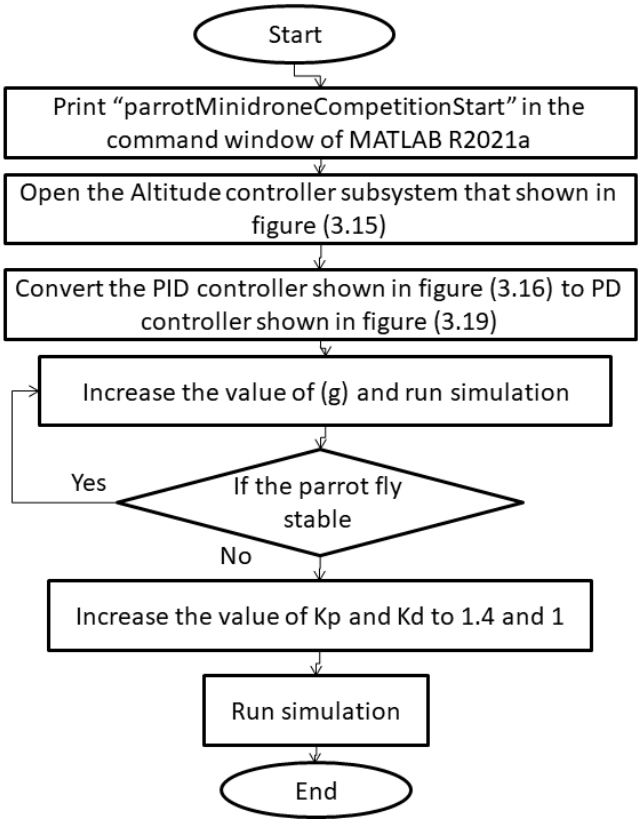


Figure (3.20) Work flow algorithm.

Tuning the PD controller: a linear model is required to tune the controllers since nonlinear models, notwithstanding their simulation accuracy, are not ideal for controller design. The height controller will be tuned in this section using Simulink's ‘PD tuner’ tool. The controller simply indicates the height to climb or drop to using a positive or negative command. The linearised controller model used for tuning is shown in Figure (3.21) below[41].

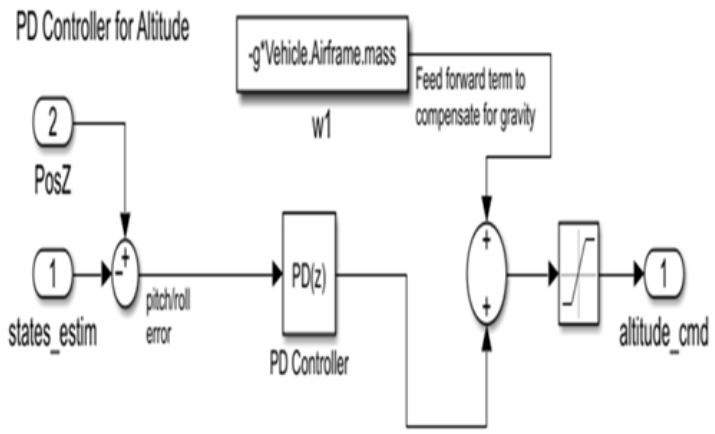


Figure (3.21) Tuning is done via a Simplified Altitude Controller.

By opening the PD block, the 'autotuner' is launched. It linearises the control loop. The program then shows the linearised version's closed-loop response, allowing you to tweak the system's reaction time and transient behaviour (as shown in Figure (3.22)).

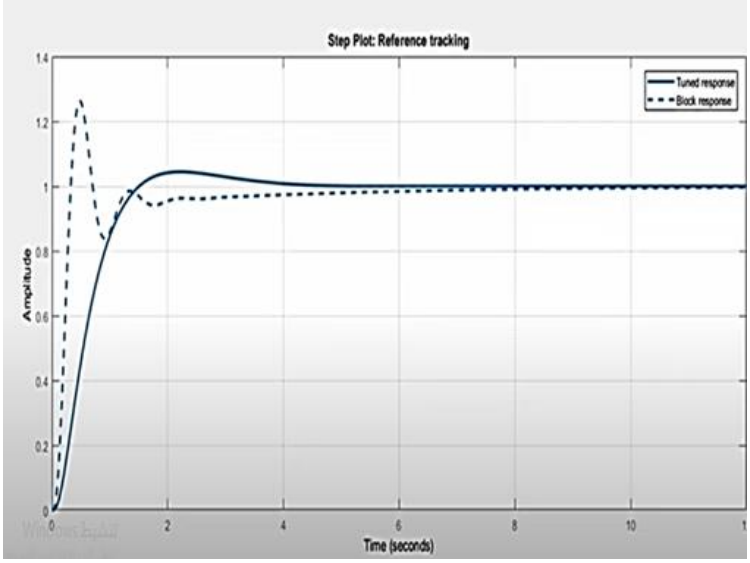


Figure (3.22) PID Tuner App on Simulink

Due to the elimination of nonlinear components, the dashed line of the response signal does not have the same simulation behaviour as the solid line, but it is still useful for tweaking purposes. Following gain selection and brief hardware testing, it became clear that the hardware does not behave as planned as it is unable to take off correctly. In this scenario, the issue has an impact on the feedforward term. If it is too low, the algorithm assumes that the drone's weight is lower than it actually is, or that the thrust is greater than it actually is. As a result of the reduced proportional route, the controller has more trouble controlling the remaining weight and it is unable to

lift off. The drone can eventually take off if the value is increased by approximately 25%[42].

3.4 Improving the Size of the Propellers of the Parrot Minidrone and an Impact Study on its Flight Controller System

In this section, projected turbulence onto the UAV. So imposed the increase in the weight of the drone caused by this disturbance. In order to make the plane withstand these weather conditions without falling and crashing, calculated an appropriate increase in the area of the plane's wing and indeed applied this in the MATLAB-R2021a Simulink program, and got good results. The new design of the propellers demonstrated the plane's ability to carry an extra payload of approximately one-third of the weight of the plane.

3.4.1 Methods of the propellers size design

The reference mass of vehicle is (0.063 kg) plus motor (0.030 kg) get total mass equal (0.093 kg)[43],[37],[43] increase to (0.105 kg) in Simulink block. The drone is flying and is stable if the force of lift (L) that gets from the rotation of the propellers is equal to the force of attraction of the earth to the body (F) as shown in Figure (3.23), where

$$L = 0.5 \times \rho \times A \times V^2 \times C \tag{3.65}$$

$$\text{and } F = mg \tag{3.66}$$

P = Air density constant (1.3 kg/m^2).

A= area of propeller (Area of the orthogonal projection of the propellers)(0.0036 m).

V= propeller speed.

C= lift coefficient.

m= mass of drone.

g= constant of gravitation (9.81 m/s^2).

Figure (3.23)

Under normal flight conditions the vehicle mass is (0.093kg)

F = mg = 0.093*9.81 = 0.91233 N = lift force

$$L = 0.5 \times \rho \times A \times V^2 \times C$$

$$V^2 \times C = \frac{0.91233}{0.5 \times 1.3 \times 0.0036} = 390$$

Under the influence of bad weather conditions, when we assumed an increase in vehicle weight from 0.093 to 0.105 kg), we can determine the area of propeller as below:

$$A = \frac{L}{0.5 \times \rho \times V^2 \times C} = \frac{0.105 \times 9.81}{0.5 \times 1.3 \times 390} = 0.004\ m^2$$

Note, we have changed the weight of the drone and the area of the propellers from the icons shown in Figure (3.24).

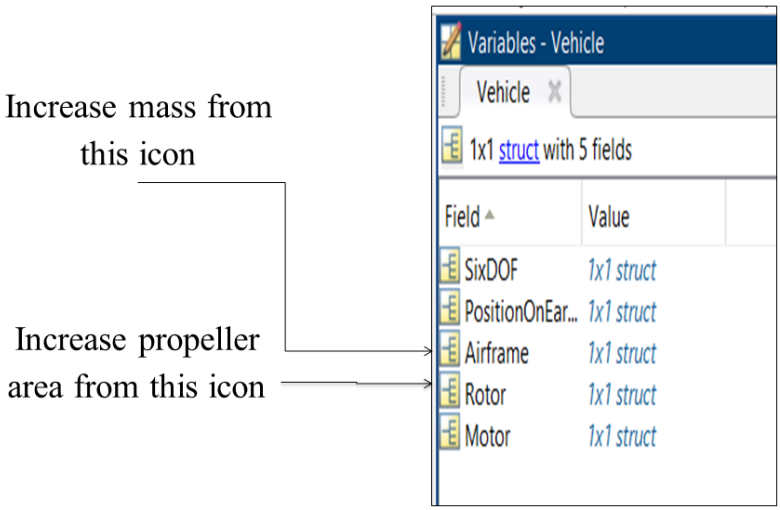


Figure (3.24)

The Table (4.2) shows other experiments. We repeated the same procedure to calculate the area of the new propeller in each case, and we got the same good results after applying the new results in Simulink. We discovered that the propellers area cannot be increased by more than 0.0045 m^2, because the size of the propellers must be proportional to the size of the frame[44].

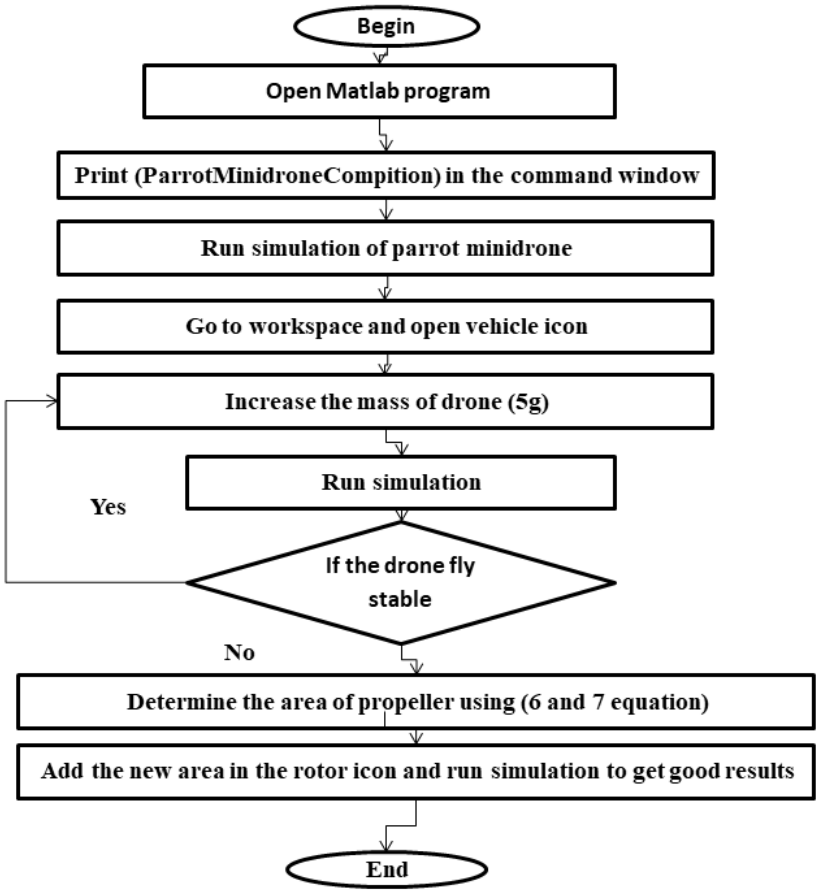


Figure (3.25) Algorithm of work

3.5 Path design using simulation

Path planning, Figure (3.26), in the Simulation support package for parrot mini drone subsystem, has two block designs to assure appropriate Parrot mini-drone flight. This initial design block contains the logic for landing the Parrot mini-drone, which will be triggered if a warning flag is engaged or if the flight time is shorter than five seconds. The waypoint follower system was adjusted in the second half of this study [39].

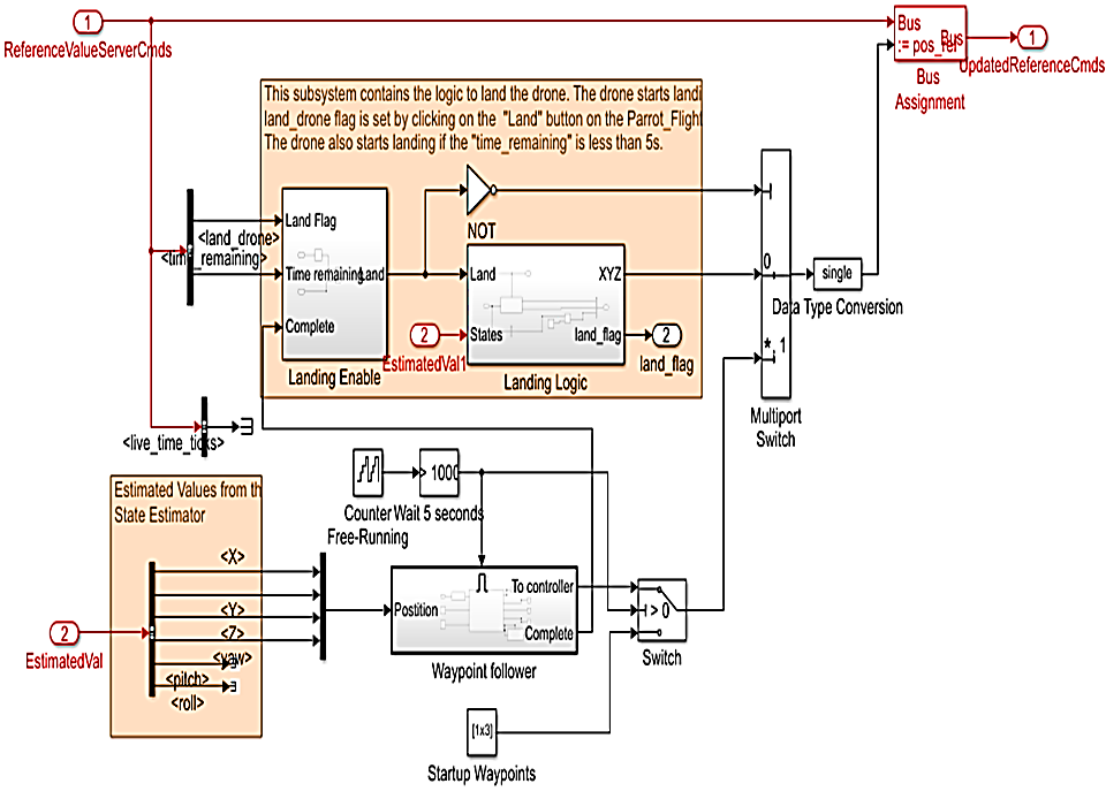


Figure (3.26) Path planning

The waypoint follower from the Robotics System Toolbox UAV library calculates the required yaw, heading, and look ahead point [39]based on the PMD location, WPs, and look ahead distance [39], as shown in Figure (3.27). The way point field is filled up with the desired way points for this research, as specified by:

[0 0 -1; 1.5 0 -1; 1.5 1.5 -1; 0 1.5 -1; 0 0 -1; 1.5 0 -1; 1.5 1.5 -1; 0 1.5 -1; 0 0 -1].

The height (1 meter) at which the drone navigates the square route is represented by the z axis value of -1.

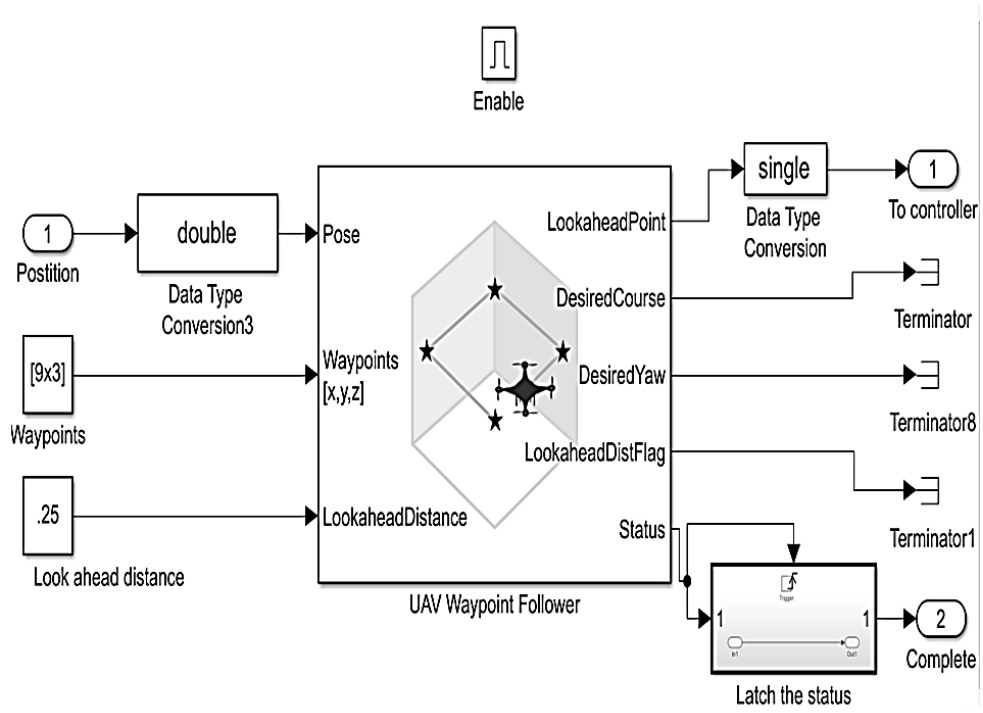


Figure (3.27) Simulation support package mini-drone of the Parrot mini-drone modified lower level waypoint follower.

The chosen waypoints, Figures (3.28), are intended to simulate all flight movements and rotations for the Parrot. Each waypoint has various requirements for intended position, yaw, pitch, and roll. The order and direction of flight for the intended Waypoint objective sites are shown in Figure (3.28).

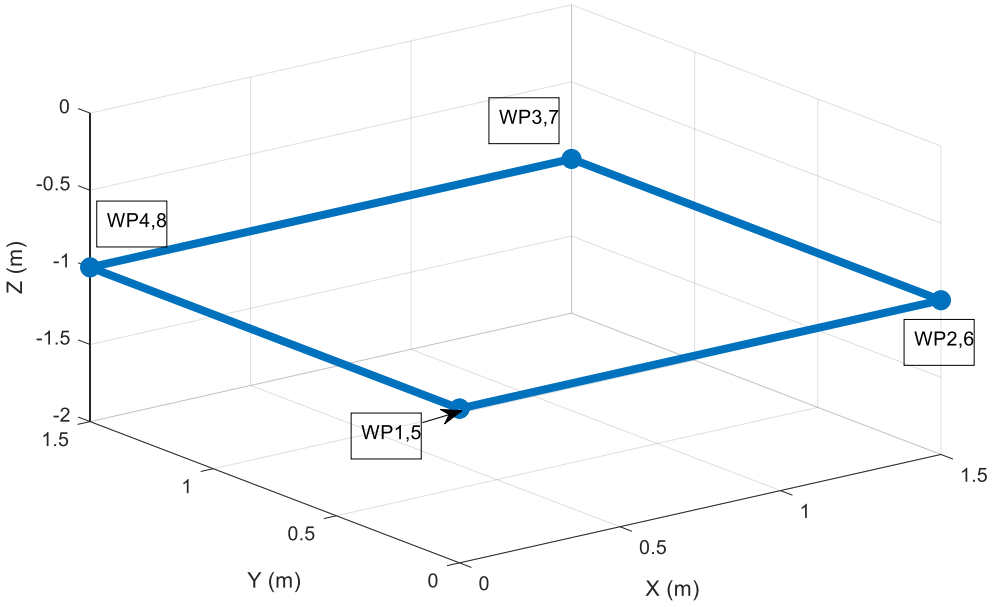


Figure (3.28) A waypoint for the Parrot mini-drone flight path planning.

We also set the transition radius parameter to 0.5 in the waypoint follower block. This number is used as the radius around a waypoint within which the drone begins moving towards the next point when it approaches it. If the transition radius is set to 0.5m and the drone is 0.5m away from the waypoint, for example, the drone will begin traveling towards the next waypoint. With a decrease in the value of the transition radius, the accuracy of reaching a chosen waypoint improves.

Chapter 4: Experimental Setup and Simulation Results

4.1 Introduction

In this chapter, we have explained and clarified all the results we reached in this research; and the researchers discussed each part of them in detail; at the end of the chapter, made a comparison between the results we achieved with the results of previous researches.

4.2 Results of Altitude controller design

In the reference conditions when the drone flies in suitable weather, we obtained the following results (Figure (4.1)):

Kp = 0.8, Kd = 0.3, ζ=0.707, wn= 190 Hz

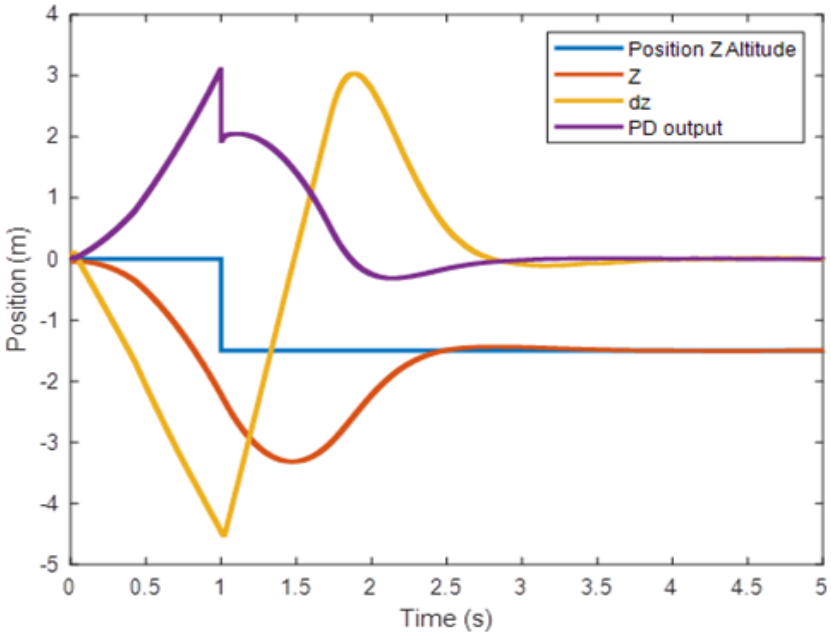


Figure (4.1) Reference results of PD controller

We increased the drone's weight to impose the falling of dust or snow due to bad weather conditions by changing the value in the block (-g*vehicle. airframe. mass). This is the plane's weight multiplied by the body's free fall constant (g = 9.81 m/s^2). The negative sign indicates that the body is increasing.

We increased the value of the constant (g) to -12.81 to impose an increase in the weight of the aircraft. This resulted in the system becoming unstable, and we attained the results shown in Figure (4.2).

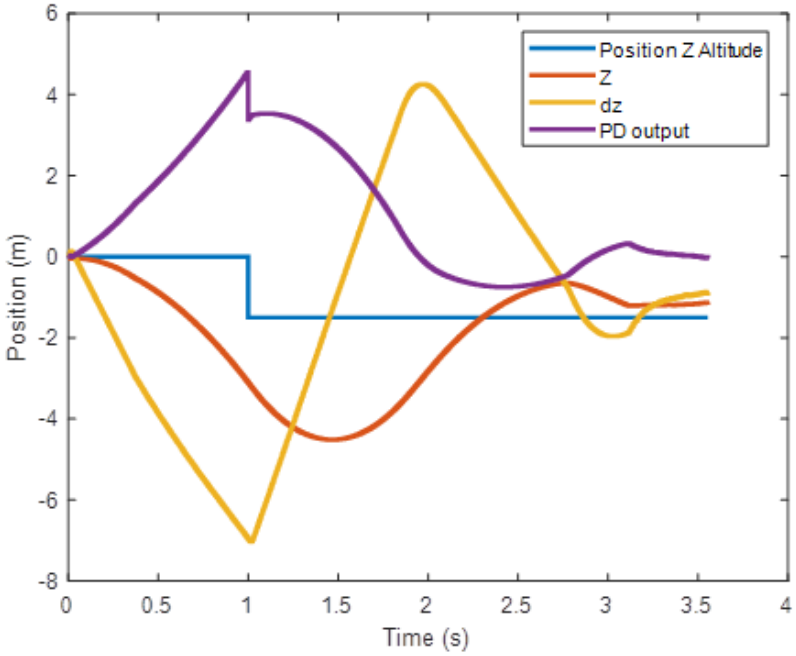


Figure (4.2) The result with disturbance .

To improve the system's response, we changed the values of (Kp) and (Kd) via the 'autotuner' method. The system responded the best when (Kp) was 1.4 and (Kd) was 1 (as shown in Figures (4.3) and (4.4)).

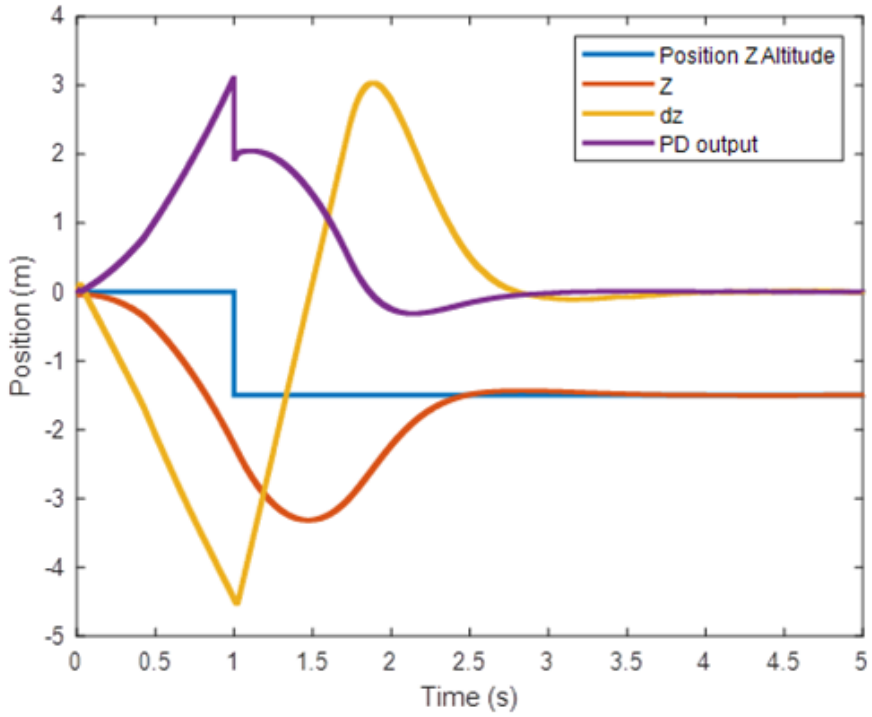


Figure (4.3) The new result with disturbance after convert the value of Kp and Kd.

We can summarize the results as shown in the Table (4.1) below:

Table (4.1) The table below shows the system's response to the different controller states with their gains values and their relationship to the value of (g).

No.	**Type of control-ler**	**Kp**	**Ki**	**Kd**	**System response**	**-g*vehicle.airframe.mass**
1	PID	0.8	0.24	0.5	Stable	-9.81* vehicle.airframe.mass
2	PD	0.8	0	0.3	Stable	-9.81* vehicle.airframe.mass
3	PD	0.8	0	0.3	Unstable	-12.81* vehicle.airframe.mass
4	PD	1.4	0	1	Stable	-12.81* vehicle.airframe.mass

4.2.1 Discussion of the results

Unmanned aerial vehicles (UAVs) may be a valuable asset in search and release missions. However, to realise their full potential, all parameters that can touch the flight of UAVs must be properly accounted for, such as the excellence of sensory operations (which can vary depending on the location of the UAVs). Most previous studies in the literature focused on the parts of the controllers in drones, especially the altitude controller. Many researchers, as explained in the introduction chapter, employed many modern and complex techniques. In this research, we used a parrot mini drone in our experiment, where we tested the performance of the plane using PD, and after subjecting the drone to some disturbance and changing the values (Kp and Kd) by auto-tuning in the simulation, we obtained good results by applying that in the MATLAB simulation. Figure (4.1) shows the altitude of aircraft Z and the estimation altitude dZ as well as the output of PD, where it can be noted that the drone is flying at a fixed altitude until the end of the specified implementation time, but when we introduce disturbance in the altitude controller (increase g to -12.81), the drone falls down after 3.5 seconds of flying, as shown in Figure (4.2). Figure (4.3) and (4.4) presents the results obtained after a change of values. KP and KD are exactly identical to the original results before the disturbance was added.

4.3 Results of the propellers size design

The experiments are listed in the Table (4.2) below. We used the same approach in each case to compute the area of the new propeller, and we received the same good results in Simulink after applying the new results. We observed that because the size of the propellers must be proportionate to the size of the frame, the area of the propellers cannot be raised by more than 0.0045 m^2.

Table. (4.2) The table below shows the area of the new propellers for each increase in the weight of the aircraft. We note that the best space for the propellers is (0.0043m2) which fits all the weights in the table as well as the original weight of the aircraft

New weight (kg)	Calculation propellers area (m^2)	The results
0.105	0.004	The drone fly and stable
0.110	0.0042	The drone fly and stable
0.115	0.0043	The drone fly and stable
0.120	0.0045	Unstable system

We was draw the input and output signals for all the controllers in the flight management system in the presence and absence of turbulence. Drawing the results of turbulence after increasing the area of the propellers to make the plane able to bear some extra weight due to weather conditions. As follows:

4.3.1 X-Y position controller

Outer loop controller for X-Y positions. It makes the drone return to its original position and the original location (0, 0). Delivering the signals to the internal loop controller for pitch/roll (or attitude).

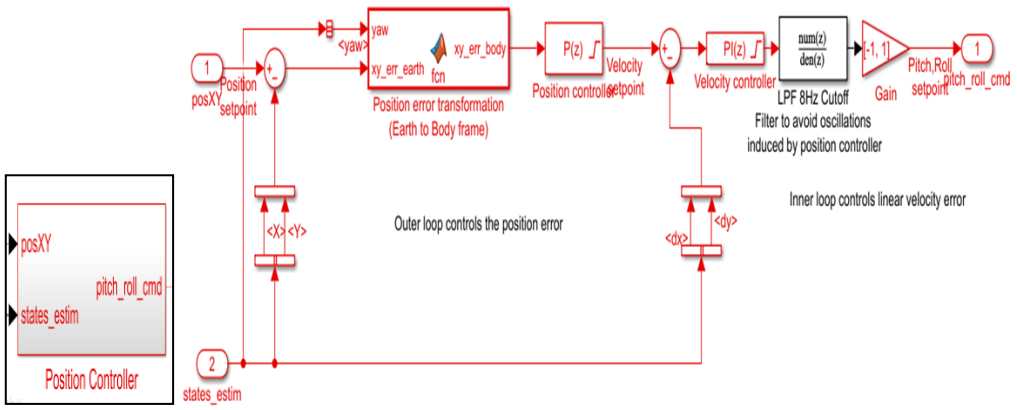


Figure (4. 5). X-Y position blocks and structure controller.

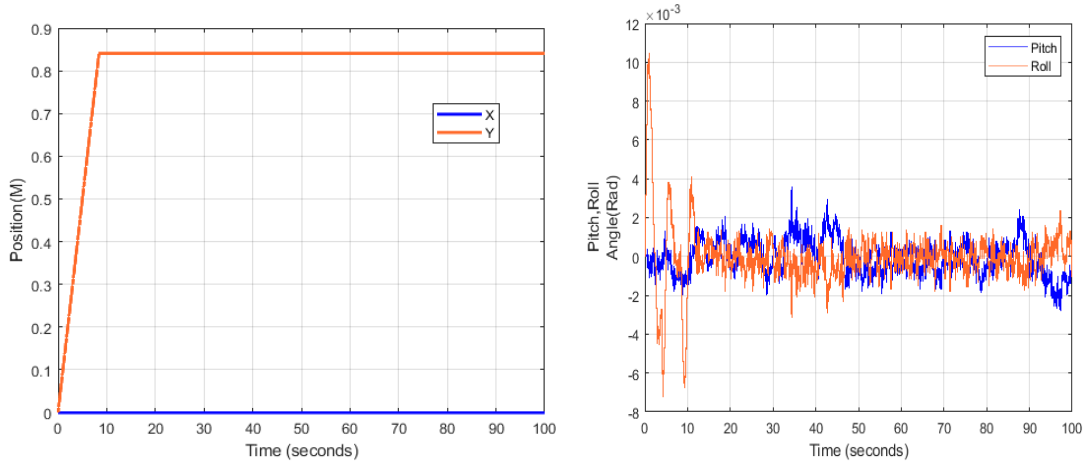


Figure (4. 6) The input and output signals of X-Y position controller without disturbance

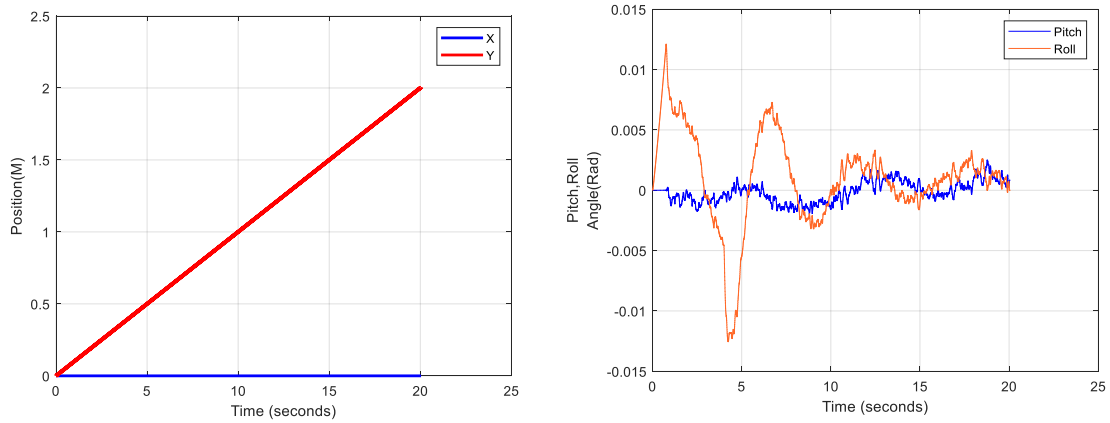


Figure (4. 7) Input signals of X-Y position controller and output signals of X-X position controller with disturbance.

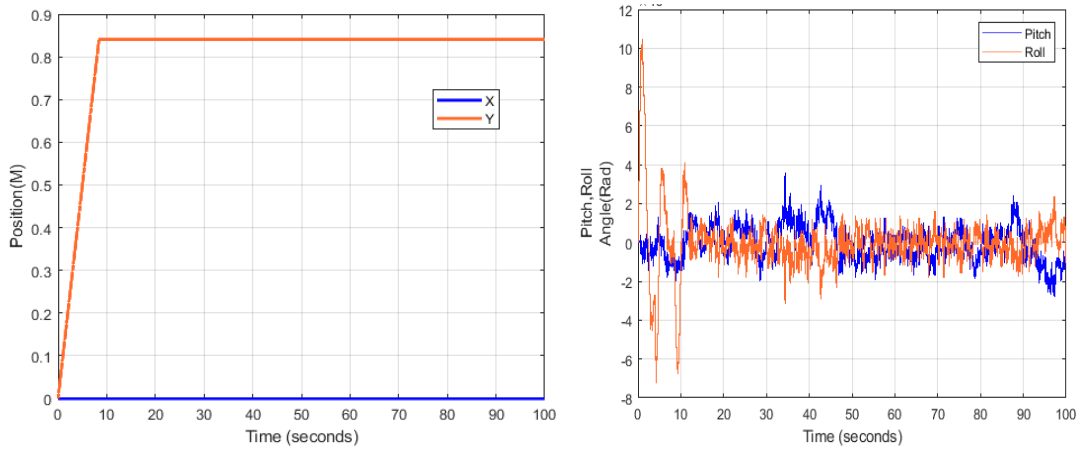


Figure (4.8) The input and output signals of XY position controller after increase the area of propeller.

4.3.2 Attitude position controller

Controller for the internal loop. These PID controllers contain force and torque commands as outputs, which are then communicated to the (Motor Mixing Algorithm) MMA (figure 3.5).

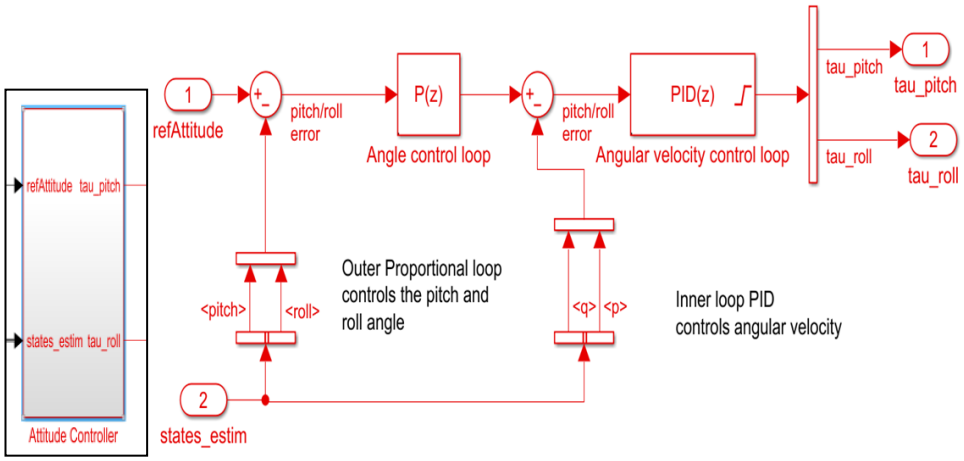


Figure (4. 9) Attitude blocks and structure controller.

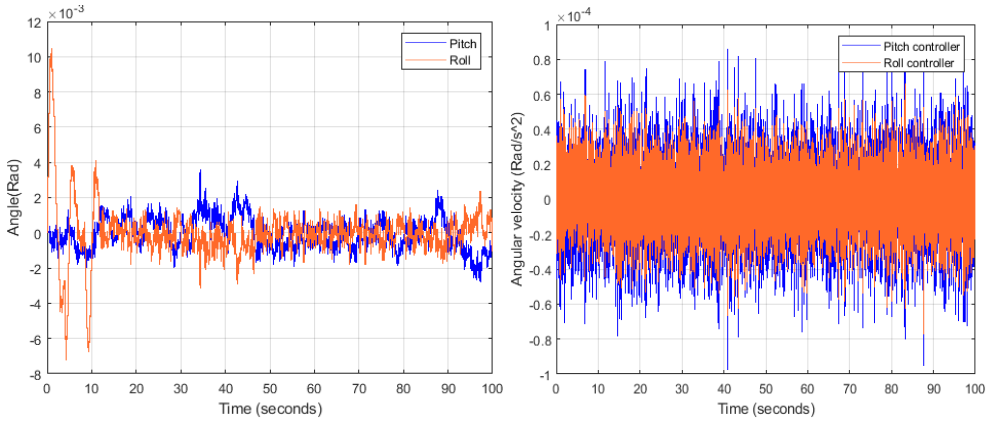


Figure (4. 10) The Altitude controller input and output signals without disturbance.

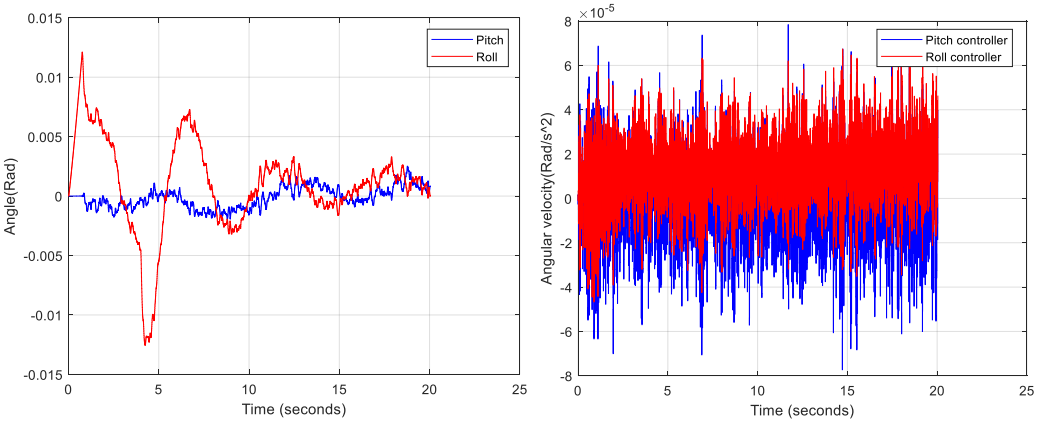


Figure (4. 11) Attitude controller input and output signals with disturbance.

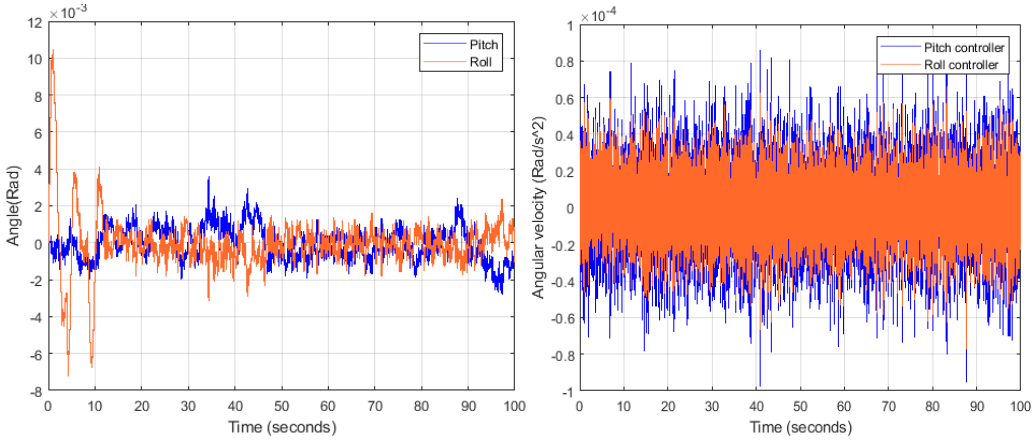


Figure (4. 12) The input and output of Attitude controller signals after increase the area of propeller.

4.3.3 Yaw controller

These PD controllers contain force and torque commands as outputs, which are then communicated to the (Motor Mixing Algorithm) MMA (figure 3.5).

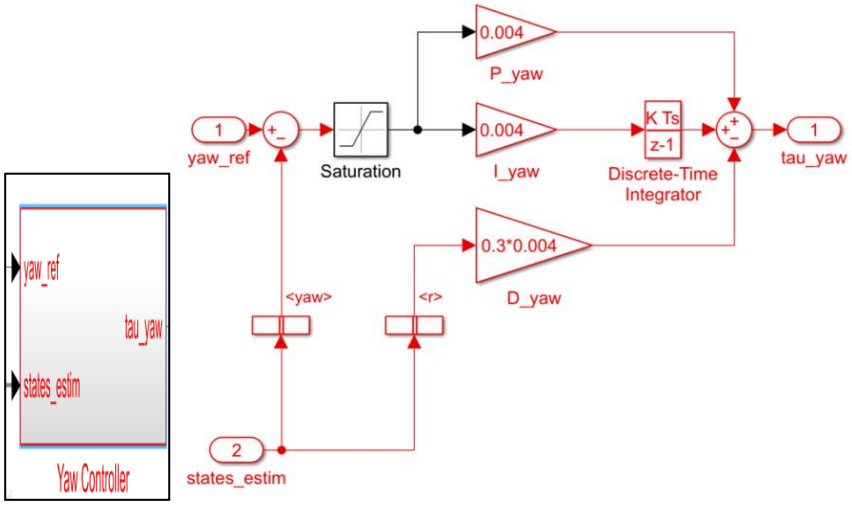


Figure (4. 13) Yaw blocks and structure controller.

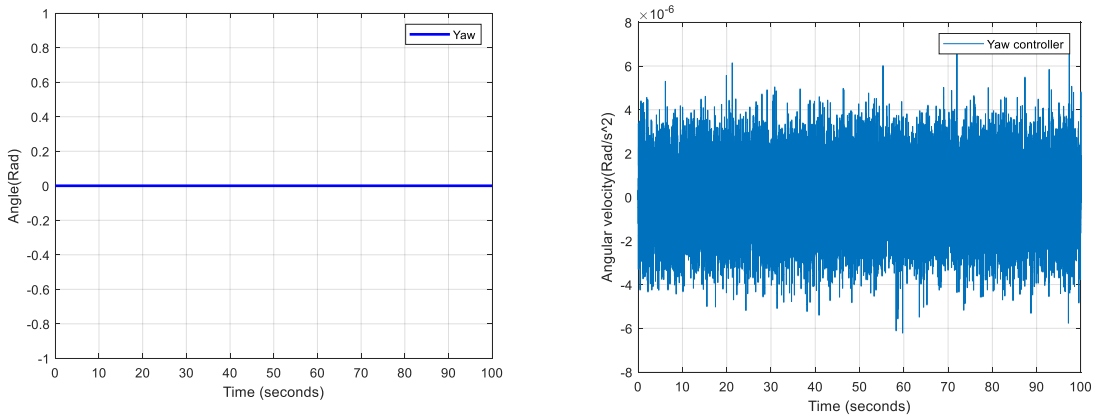


Figure (4.14) The input and output of Yaw controller signals without disturbance.

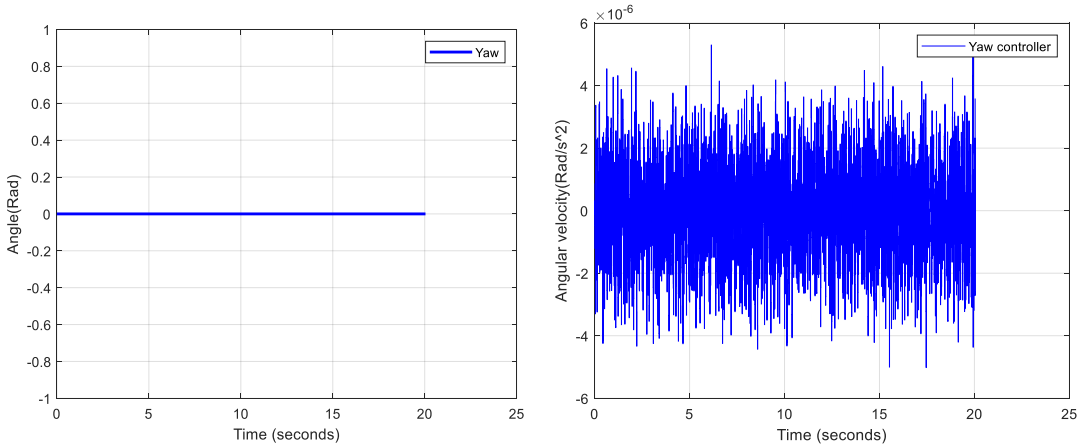


Figure (4.15) The input and output of Yaw controller signals with disturbance.

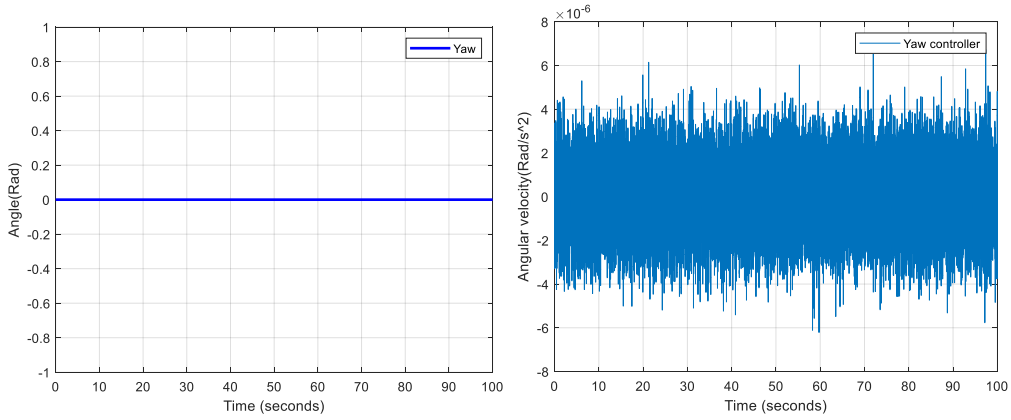


Figure (4. 16) The input and output of yaw controller signals after increase the area of propeller.

4.3.4 Altitude controller

The altitude controller is set up with a PID controller. In this approach, the proportional gain is multiplied by the altitude error generated by the sonar sensor, and the derivative gain is multiplied by the gyroscope's altitude rate measurement, which is a less noisy signal than the ultrasound signals. It's important to note that the z-axis in the drone's coordinate system points down, so the altitude value will always have a negative sign in front of it in the control system (expressed in meters).

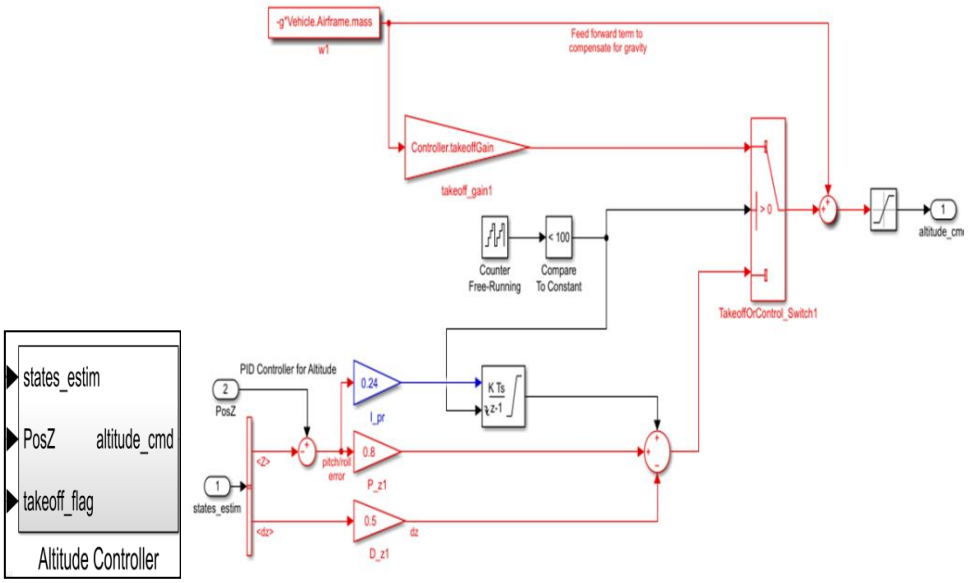


Figure (4. 17) Altitude blocks and structure controller.

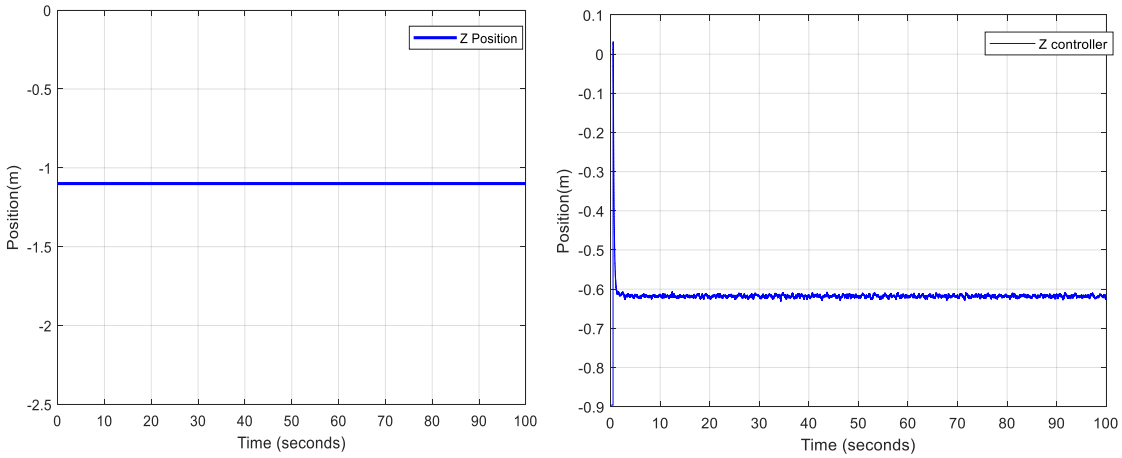


Figure (4. 18) Altitude controller input and output signals without disturbance.

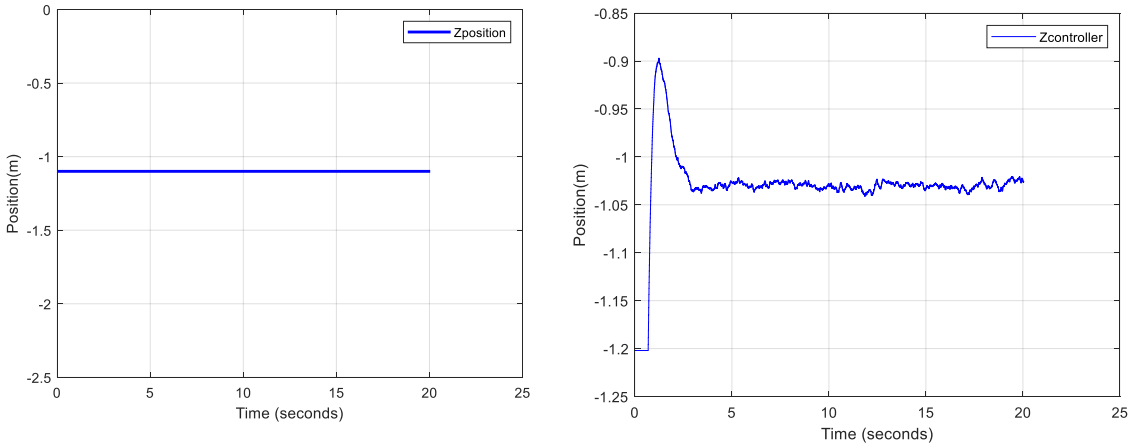


Figure (4.19) Altitude controller input and output signals with disturbance.

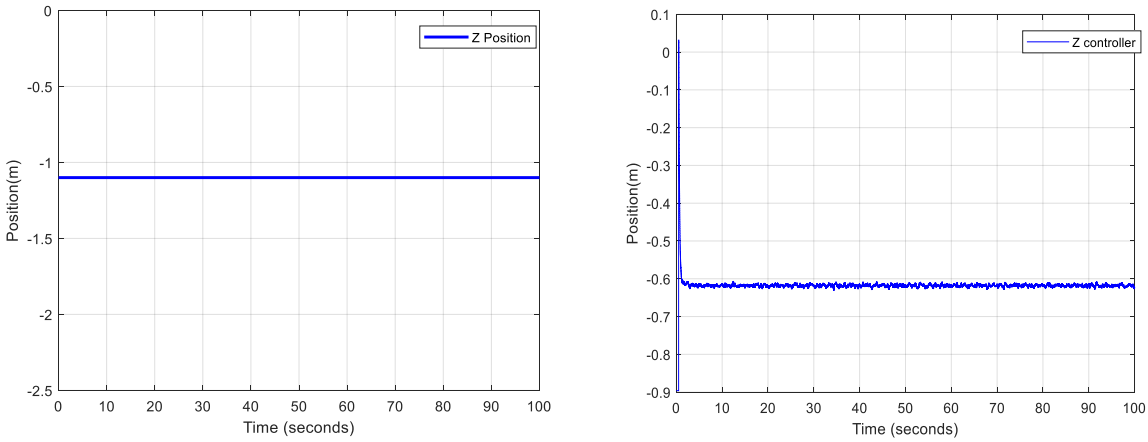


Figure (4. 20) The input and output of altitude controller signals after increase the

4.3.5 Discussion the results of propeller's size

Most of the previous research focused on how to maintain the safety of the drone in various weather conditions. Some research has developed control units for drones and made them fly as stable as possible. Some research has developed and added sensors for the plane. Others are concerned with battery life. We explained this in detail at the beginning of the research. In this study, we have imposed an increase in the aircraft's weight by bad weather. We have extracted the results for each controller in the case of good conditions and bad conditions. We have noticed that the increase in the reference weight of the aircraft is leading to a significant defect in the flight of the aircraft, which causes its collision and crash, as illustrated in the method section

in Table (4.2). We did the experiment using the Matlab program and got good results that were completely identical to the original results, as shown in the results section. We chose to extract the results of one trial because we got the same results in all the different cases **.** As shown in the Table (4.2).

When the weight and lift are in opposite and equal directions, the aircraft is in equilibrium in the air; it does not rise or fall. When the lift and resistance are in opposite and equal directions, the aircraft is at a constant speed. Lift is the force generated by the area of low pressure along with the upper layer of an aircraft's wing when compared to the area of high pressure along with the lower layer of the same wing. The pressure difference between the top and bottom of the wing results in a force that pushes the wing toward the area of least pressure - that's lifting. According to the experiment that we performed in this study, we proved that if the weight of the aircraft is increased more than the lift force, the increase in the propeller area by a certain amount has a significant impact on maintaining the stability of the aircraft and preventing it from falling and crashing.

We notice in Figures ((4.6), (4.10), (4.14) and (4.18)) that the drone has completed its flight to the end of the time period specified in the simulation, which is 100 seconds without disturbances. But after increasing the weight of the drone by a certain amount, the results showed in Figures ((4.7), (4.11), (4.15), and (4.19)) that the drone, after (19-20) seconds from the time of take-off, a defect occurred in the plane's system and was subjected to collision and fall, and after improving the size of the drone's propellers by a certain amount, the drone continued its flight normally as shown in Figures ((4.8), (4.12), (4.16) and (4.20)).

4.4 Results of path planning

Figure (4.21) shows how the command inputs for yaw, pitch, and roll for the Parrot mini-drone are compared to the overall time in seconds taken for the simulated flight**.** When the system reaches the target Waypoint, it marks it with a green line**.** At the end of every Waypoint destination objective, there is a noticeable increase in input. This is due to the Parrot making the necessary changes to orient the location of the next Waypoint. After the initial command inputs to the motors, the graphs reveal very little noise.

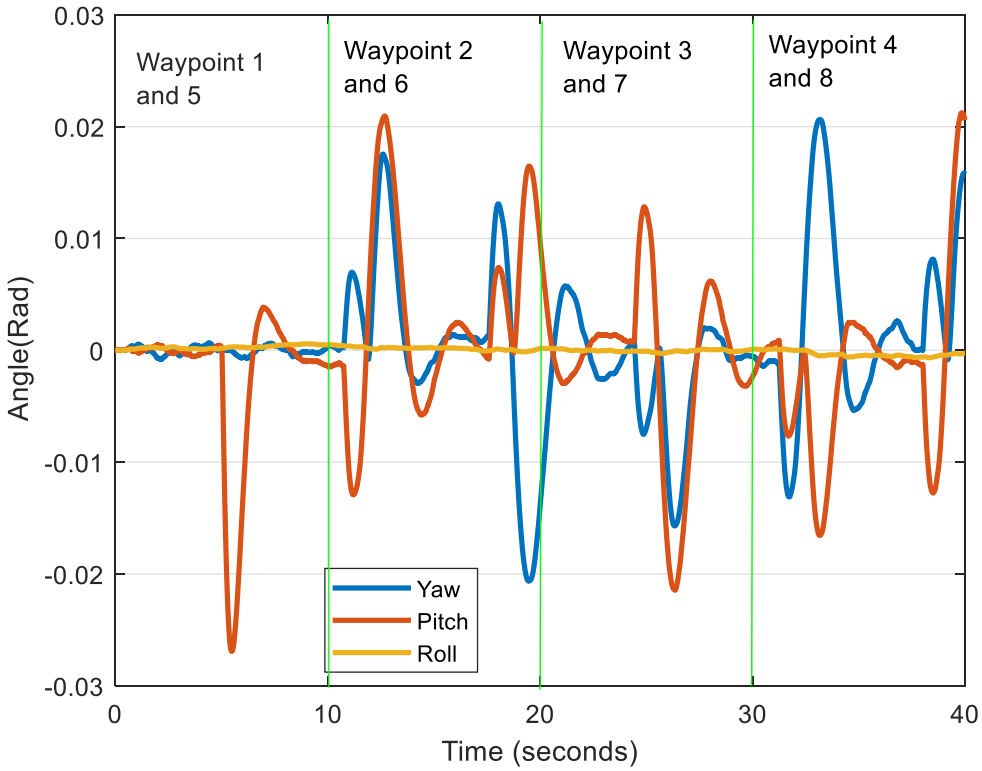


Figure (4. 21). The three rotation of Parrot mini-drone (Yaw, Pitch and Roll).

The response for each axis and range of motion, as well as the errors associated with each axis, are provided in this section. The X, Y, and Z-axis responses during the Parrot's full flight are shown in Figure (4.22). Time is shown against the relevant axis location in meters in the figures. A green line runs through each Waypoint. Figures (4.21) and (4.22) separate the axes and rotations into separate plots, allowing you to see the reaction for each one separately. There is a little disruption and a delay in flight from the beginning to Waypoint2 in all plots. The delay is caused by the SSPPM design, which has been modified to start the simulated model one meter off the ground when it is initialized. Because the model adapts to the right location using the Kalman filter that is incorporated, as well as the reprogrammed control systems developed inside the Simulation support package of Parrot mini-drone, the disruption is more noticeable at the start of the simulation.

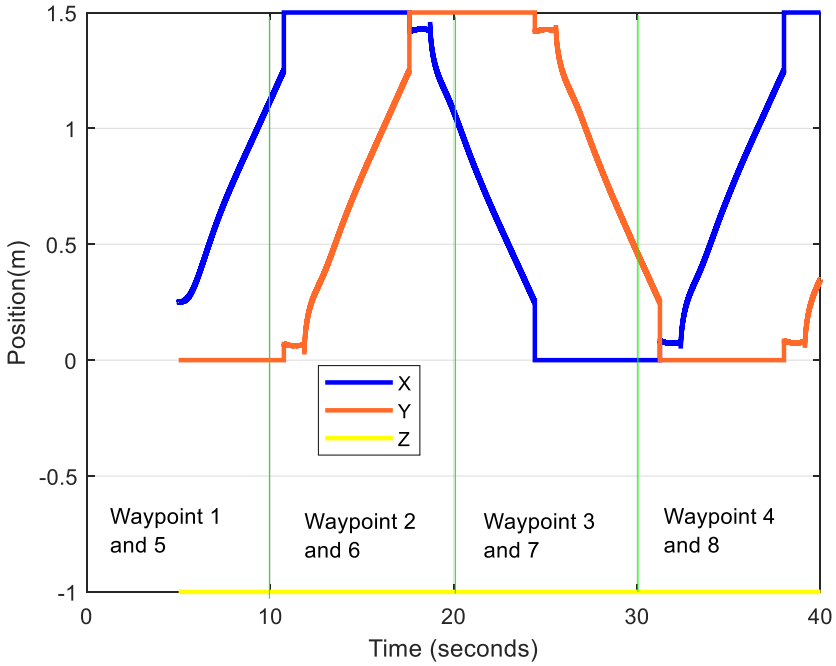


Figure (4. 22) The PMD's distance response on three axes (X, Y, and Z).

4.4.1 Cross tracking error

Cross track error from UAV position to path, returned as a positive numeric scalar in meters. The error measures the perpendicular distance from the UAV position to the closest point on the path (as shown in Figure (4.23)). The cross error of this path is 0.3 m (as shown in Figure (4.23)).

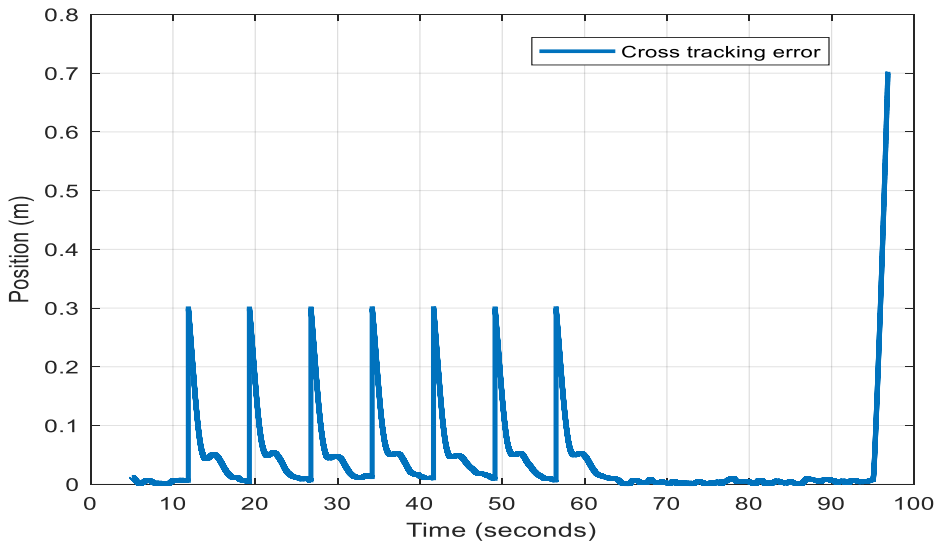


Figure (4. 23) The tracking error in the waypoint follower of PMD.

Table (4.3) A comparison between the results we have reached in our study with the results of similar studies.

No.		Our research results	J. Chaoraingern, et al[45]	Mücahid Rıdvan Kaplan, et al[15]
1	Design (PD) altitude controller	Kp=1.4 Kd=1 The results we obtained in the case of bad conditions correspond one 100% to the case of flying under normal conditions	Kp=1.15 Kd=0.42 The results were obtained in the case of bad conditions correspond one 100% to the case of flying under normal conditions	When compared to its PD equivalent, the FPI-PD control system improved the x-axis settling time value by 58 %.
2	Path planning	The error of the path=0.3 m	The error of the path=0.000 m	The error of the path=0.1 m
3	Improved the propellers size	A=430mm The results we obtained in the case of bad conditions correspond one 100% to the case of flying under normal conditions	E. Kuantama, et al [44]	S. Yadav, et al[46]
			Adding a cover to a 406 (mm) propeller, also known as a ducted propeller, can improve its performance.	In RC drones, maximum propeller sizes range from 4 to 12 inches (305 mm).

Chapter 5: Conclusions and Future work.

5.1 Conclusions

Design implementation and analysis the flight management system of drones and improving their size and function is one of the important and modern issues that concern researchers at the present time because drones are characterized by their speed and ease of movement, as they perform many important functions for humans.

In this study, we designed a control unit for the height of the drone for a control device of type (PD) and we set to calculate the values of (Kp and Kd) using the Matlab Simulation program. With this controller to withstand a pulling force downward greater than the force of Earth's attraction to the body, while previous studies worked to develop the control unit by making the drone more stable in normal weather conditions.

Also improved the size of the propellers of the drone, and the results proved that the drone is able to carry an additional weight equivalent to one-third of the weight of the plane. Using a coordinate follower from the Robotics System Toolbox UAV library, we mapped the path of the UAV and found the error in this path, and the error rate we reached was close to the results of previous research as shown in the Table (4-3).

The first chapter of this research is an introduction to the topic, an explanation of the research problem, its objectives and contributions, and a review of the most important studies and previous research on this topic. In the second chapter, present a complete study of the parrot drone, explaining the mathematical model. In Chapter Three, explain the parrot mini-drone simulation model and the methodology used to design the altitude control system for this aircraft that works with linear and non-linear designs and has the ability to withstand greater downward resistance than the older design, the methodology used to optimize the size of the drone's propellers and we made the drone to trace the layout of a particular path and we determined the error rate of this path. In the fourth chapter, presented our findings in this research and discussed these results. The fifth chapter represents the summary and future work. Applied the new results in MATLAB and got good results.

5.2 Future Works

In this section, we have proposed a number of important topics for future work that can be done as the follows in the future:

1. Applying the steps we've taken in this search on micro drones and larger drones
2. Design and implementing the sensor's block in the Parrot mini drone by using sensors to skip barriers, location sensors, and other sensors.
3. Studying the impact of noise on sensors.
4. Creating new parrot mini drone controllers to achieve the best results.
5. Programming the drone to follow different and complex paths, calculating the error percentage in these paths, and choosing the best path.

References

[1] I. Rubio Scola, G. A. Guijarro Reyes, L. R. Garcia Carrillo, J. Hespanha, and J. Xie, “Translational model identification and robust control for the parrot mambo UAS multicopter,” 2019 IEEE Globecom Work. GC Wkshps 2019 - Proc., 2019, doi: 10.1109/GCWkshps45667.2019.9024528.

[2] S. Bouabdallah, A. Noth and R. Siegwan, “PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor” , Proceedings of 2004 1EEElRS.J Internationel Conference On Intelligent Robots and Systems September 28. October 2,2004, Sendal, Japan

[3] R. Ji and J. Ma, “Mathematical modeling and analysis of a quadrotor with tilting propellers,” Chinese Control Conf. CCC, vol. 2018-July, pp. 1718–1722, 2018, doi: 10.23919/ChiCC.2018.8482899.

[4] R. Debevec, “A Smart UAV Platform for Railroad Inspection,” pp. 2004–2019, 2019.

[5] R. L. Allen III, “Quadrotor Intercept Trajectory Planning and Simulation,” , NAVAL POSTGRADUATE SCHOOL MONTEREY, CALIFORNIA no. March, 2009.

[6] P. Ceppi, “Model-based Design of a Line-tracking Algorithm for a Low-cost Mini Drone through Vision-based Control” , University of Illinois at Chicago 2020.

[7] J. S. G. Guerrero, A. F. C. González, J. I. H. Vega, and L. A. N. Tovar, “Instrumentation of an Array of Ultrasonic Sensors and Data Processing for Unmanned Aerial Vehicle (UAV) for Teaching the Application of the Kalman Filter,” Procedia Comput. *Sci.*, vol. 75, no. Vare, pp. 375–380, 2015, doi: 10.1016/j.procs.2015.12.260.

[8] A. Hussein, A. Al-Kaff, A. De La Escalera, and J. M. Armingol, “Autonomous indoor navigation of low-cost quadcopters,” 10th IEEE Int. Conf. Serv. Oper. Logist. Informatics, SOLI 2015 - conjunction with ICT4ALL 2015, no. November, pp. 133–138, 2015, doi: 10.1109/SOLI.2015.7367607.

[9] A. A. J. LEFEBER, “CONTROLLING OF AN SINGLE DRONE: Hovering the drone during flight modes,” 2015.

[10] Y. Son *et al.*, “Rocking drones with intentional sound noise on gyroscopic sensors,” Proc. 24th USENIX Secur. Symp., pp. 881–896, 2015.

[11] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, “Dynamics modelling and linear control of quadcopter,” *Int.* Conf. Adv. Mechatron. *Syst. ICAMechS*, vol. 0, pp. 498–503, 2016, doi: 10.1109/ICAMechS.2016.7813499.

[12] A. R. Deevi, P. Misra, and P. Balamurali, “Aerial Drones with Ears,” no. July, pp. 338–339, 2016, doi: 10.1145/2994551.2996695.

[13] G. Perozzi, D. Efimov, J. M. Biannic, L. Planckaert, and P. Coton, “Wind estimation algorithm for quadrotors using detailed aerodynamic coefficients,” Proc. Am. Control Conf., vol. 2018-June, pp. 1921–1926, 2018, doi: 10.23919/ACC.2018.8431879.

[14] M. Andreetto and L. Palopoli, “Constrained Kalman Filter for Adaptive Prediction in Minidrone Flight,” 2019 IEEE Int. Instrum. Meas. Technol. Conf., pp. 1–6, 2019.

[15] M. R. Kaplan, A. Eraslan, A. Beke, and T. Kumbasar, “Altitude and Position Control of Parrot Mambo Minidrone with PID and Fuzzy PID Controllers,” ELECO 2019 - 11th Int. Conf. Electr. Electron. Eng., pp. 785–789, 2019, doi: 10.23919/ELECO47770.2019.8990445.

[16] G. Ariante, U. Papa, S. Ponte, and G. Del Core, “IMU sensors data : Kalman filtering and integration,” 2019 IEEE 5th Int. Work. Metrol. Aerosp., no. July, pp. 522–527, 2019.

[17] Bushra, T. Hai, and M. B. Kadri, “Comparison of different techniques for experimental modeling of a Quadcopter,” 2019 2nd Int. Conf. Latest Trends Electr. Eng. Comput. Technol. INTELLECT 2019, 2019, doi: 10.1109/INTELLECT47034.2019.8955470.

[18] O. M. Gamulescu, S. D. Rosca, F. Panaite, A. Costandoiu, and S. Riurean, “Accident sites management using drones,” MATEC Web Conf., vol. 305, p. 00004, 2020, doi: 10.1051/matecconf/202030500004.

[19] A. Noordin, M. A. M. Basri, and Z. Mohamed, “Simulation and experimental study on pid control of a quadrotor MAV with perturbation,” Bull. Electr. Eng. Informatics, vol. 9, no. 5, pp. 1811–1818, 2020, doi: 10.11591/eei.v9i5.2158.

[20] J. Chaoraingern, “Modified Adaptive Sliding Mode Control for Trajectory Tracking of Mini-drone Quadcopter Unmanned Aerial Vehicle,” vol. 13, no. 5, pp. 145–158, 2020, doi: 10.22266/ijies2020.1031.14.

[21] A. Talaeizadeh, D. Antunes, H. N. Pishkenari, and A. Alasty, “Mechatronics Optimal-time quadcopter descent trajectories avoiding the vortex ring and,” vol. 68, no. March, 2020, doi: 10.1016/j.mechatronics.2020.102362.

[22] H. Siguerdidjane and G. Sordi, “Various student ’ s projects related to aerospace control education,” 2020.

[23] Y. Chen, “A Case for a Battery-Aware Model of Drone Energy Consumption,” no. August, 2020, doi: 10.1109/INTLEC.2018.8612333.

[24] A. Kumar, K. Sharma, H. Singh, and S. Gupta, “A drone-based networked system and methods for combating coronavirus disease (COVID-19) pandemic,” Futur. Gener. Comput. *Syst.*, vol. 115, pp. 1–19, 2021, doi: 10.1016/j.future.2020.08.046.

[25] S. Waitman, H. Alwi, and C. Edwards, “Flight evaluation of simultaneous actuator/sensor fault reconstruction on a quadrotor minidrone,” IET Control Theory Appl., vol. 15, no. 16, pp. 2095–2110, 2021, doi: 10.1049/cth2.12180.

[26] R. Casado and A. Bermúdez, “A simulation framework for developing autonomous drone navigation systems,” *Electron.*, vol. 10, no. 1, pp. 1–17, 2021, doi: 10.3390/electronics10010007.

[27] S. DeBock “GUIDANCE, NAVIGATION, AND CONTROL OF A QUADROTOR DRONE WITH PID CONTROLS,” , NAVAL POSTGRADUATE SCHOOL no. September, 2020.

[28] M. Orsag, C. Korpela, P. Oh, and S. Bogdan, Aerial Manipulation. 2018, https://doi.org/10.1007/978-3-319-61022-1.

[29] R. Prasad, *Slotine* "APPLIED NONLINEAR CONTROL". 2014, Applied nonlinear control / Jean-Jacques E. Slotine, Weiping L. ISBN 0-13-040890-5

[30] “Introduction to Quadrotors and Control Theory | Udemy.” https://www.udemy.com/course/quadrotors/ (accessed Feb. 02, 2022).

[31] “Programming Drones with Simulink Video - MATLAB & Simulink.” https://ch.mathworks.com/videos/programming-drones-with-simulink-1513024653640.html (accessed Feb. 15, 2022).

[32] Y. I. Jenie, A. Fathurrahman, O. Arifianto, and R. A. Sasongko, “Mathematical modelling and simulation of a quadrotor unmanned aerial vehicle with automatic altitude and speed control,” *AIP Conf. Proc.*, vol. 2226, no. April, 2020, doi: 10.1063/5.0002805.

[33] “Blade Element Propeller Theory | Aerodynamics for Students.” http://www.aerodynamics4students.com/propulsion/blade-element-propeller-theory.php (accessed Feb. 14, 2022).

[34] “MPU6050 datasheet.” https://datasheetspdf.com/pdf-file/735134/InvenSense/MPU6050/1 (accessed Mar. 29, 2022).

[35] “MathWorks - MATLAB and Simulink Conferences - MATLAB & Simulink.” https://ch.mathworks.com/ (accessed Apr. 08, 2022).

[36] “Parrot Mambo drone downloads | Parrot Support Center.” https://www.parrot.com/us/support/documentation/mambo-range (accessed Mar. 27, 2022).

[37] “Simulink ® Support Package for Parrot ® Minidrones User ’ s Guide.” https://www.parrot.com/us/support/documentation/mambo-range (accessed Mar. 27, 2022).

[38] “AA 203: Optimal and Learning-based Control.” https://stanfordasl.github.io/aa203/ (accessed Mar. 27, 2022).

[39] “Simulink Support Package for Parrot Minidrones - File Exchange - MATLAB Central.” https://ch.mathworks.com/matlabcentral/fileexchange/63318-simulink-support-package-for-parrot-minidrones (accessed Feb. 19, 2022).

[40] “Simulink Support Package for Parrot Minidrones - File Exchange - MATLAB Central.”

https://ch.mathworks.com/matlabcentral/fileexchange/63318-simulink-support-package-for-parrot-minidrones (accessed Feb. 01, 2022).

[41] Y. Chen, "OptimPID : A MATLAB Interface for Optimum PID Controller Design ⋆," no. March, pp. 28–30, 2012.

[42] C. S. Veerappan, P. K. K. Loh, and R. J. Chennattu, Smart Drone Controller Framework—Toward an Internet of Drones, vol. 1002, no. February. Springer Singapore, 2022.

[43] F. Riether, "Getting started with MIT ' s Rolling Spider MATLAB Toolbox MIT ' s Rolling Spider MATLAB Toolbox," pp. 1–31, 2015.

[44] E. Kuantama and R. Tarca, "Quadcopter thrust optimization with ducted-propeller," MATEC Web Conf., vol. 126, pp. 1–4, 2017, doi: 10.1051/matecconf/201712601002.

[45] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, "Mini-drone quadrotor altitude control using characteristic ratio assignment PD tuning approach," Lect. Notes Eng. Comput. Sci., vol. 2019-Octob, pp. 337–341, 2019.

[46] S. Yadav, M. Sharma, and A. Borad, "Thrust efficiency of drones (quadcopter) with different propellers and and their payload capacities," *Int. J.* Aerosp. Mech. Eng., vol. 4, no. 2, pp. 18–23, 2017, [Online]. Available: http://www.ijamejournals.com/pdf/rpm11174.pdf.

الخلاصة

يمثل نظام إدارة الطيران في طائرة بدون طيار التصميم الكامل للطائرة بدون طيار بكل أجزائها ومكوناتها. تكمن مشكلة البحث في أن الطائرات بدون طيار ذاتية القيادة وتُستخدم دائمًا في الأماكن الخطرة التي لا يستطيع الانسان الوصول إليها. إنها مصنوعة من مواد خفيفة الوزن ، لذا فهي ليست متينة ، وأثناء رحلتها تتعرض لظروف جوية مختلفة ، لذلك من الضروري أن يكون نظام التحكم فيها صلبًا ومصممًا بشكل مناسب لغرض الحفاظ على التوازن والاستقرار أثناء الرحلة.

نستخدم في هذا البحث تصميم طائرة Parrot mini-drone Mambo ، والتي تتميز طائرتها بالقدرة على برمجتها وتغيير تصميمها باستخدام برنامج Matlab. نقدم دراسة كاملة لنظام إدارة الطيران في هذه الطائرة ، بدءًا من النموذج الرياضي والتصميم إلى مكونات البرامج والأجهزة.

في هذه الدراسة ، نقوم بتصميم نظام التحكم في ارتفاع الطائرة ، وهو وحدة تحكم PD تناسبية متكاملة لها القدرة على العمل مع الأنظمة الخطية وغير الخطية.

أيضًا قمنا بتحسين حجم أجنحة الطائرات عند تطبيق التصميمات الجديدة في برنامج MATLAB R2021a. تظهر النتائج أن وحدة التحكم الجديدة تجعل الطائرة قادرة على تحمل قوة جر هابطة تبلغ $12.18\ m/s^2$، بينما التصميم القديم لا يتحمل أكثر من القوة الأرضية للجسم ، والتصميم الجديد لحجم الأجنحة يجعل الطائرة القادرة على حمل حمولة زائدة تقارب ثلث وزن الطائرة.

أيضًا ، جعل الطائرة تتبع مسارًا مربع الشكل. نسبة خطأ التتبع المتقاطع ، وهي النسبة بين المسار المتقاطع لموقع الطائرة والمسار المحدد ، وهو قياس رقمي موجب محسوب بالأمتار من الارتفاع العمودي للمستوى فوق المسار ، هو 0.3 متر.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة الفرات الاوسط التقنية

الكلية التقنية الهندسية– نجف

تصميم ومحاكاة نظام التحكم في الطيران لطائرة Parrot Mini–Drone

رسالة مقدمة الى

قسم هندسة تقنيات الاتصالات

كجزء من متطلبات نيل درجة ماجستير تقني في هندسة الاتصالات

تقدم بها

اسراء هادي كاظم

بكالوريوس في هندسة الاتصالات

إشراف

الأستاذ الدكتور احمد طه عبد السادة الجياشي

2022 /