JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

GARUDA-"Guided Autonomous Rotorcraft Depth Analysis"

Sangeetha N, Dr. Shwetha N, Dr. Roopesh R, Lisha S Gowda, Manasa G, Gagana P, S V Shrigandha ,

Assistant Professor, Assistant Professor, Assistant Professor, Student, Student, Student,

Electronics and Communication Department
Dr. Ambedkar Institute of Technology, Bangalore, Karnataka, India

Abstract : This paper describes the development of GARUDA (Guided Autonomous Rotorcraft Using Depth Analysis), a custom-built quadcopter flight controller designed for flexibility and research experimentation. Instead of relying on commercially available closed-source controllers, the proposed system is developed from the ground up using an STM32 microcontroller. The controller combines data from multiple onboard sensors, including an IMU, GPS, barometer, magnetometer, and ultrasonic sensors, to maintain stable flight and situational awareness. A PID-based control strategy is implemented and carefully tuned to ensure smooth and reliable attitude control. To support monitoring and user interaction, an ESP32 module enables real-time telemetry and communication with a mobile ground control application. Additionally, a depth-based vision approach is integrated to assist in obstacle detection and autonomous navigation. Practical testing confirms that the system delivers stable flight behavior, accurate sensor fusion, and effective obstacle avoidance, making it suitable for both academic research and real-world UAV applications.

***IndexTerms*-STM32 microcontroller, quadcopter flight controller, PID control algorithm, inertial measurement unit (IMU), sensor, ESP32communication, real-time monitoring, open-source drone design.**

I. INTRODUCTION

Unmanned Aerial Vehicles have moved far beyond hobby-level platforms and are now widely used in fields such as inspection, mapping, monitoring, and autonomous navigation. As UAV applications continue to expand, the demand for flight controllers that are both reliable and adaptable has increased. However, many existing flight controllers available in the market operate as closed systems, offering limited flexibility for customization, experimentation, and academic research.

The GARUDA project was initiated to address these limitations by developing a fully customizable flight controller that allows complete control over both hardware and software design. The primary goal of this work is to create a platform that can be easily modified, tested, and extended according to specific research or application needs. By building the controller from scratch, deeper insight into flight dynamics, sensor behavior, and control algorithms can be achieved.

At the core of the system is the STM32F411 microcontroller, selected for its processing capability, real-time performance, and low power consumption. The controller continuously processes data from multiple sensors to estimate the quadcopter's orientation, altitude, and position. Based on this information, motor speeds are adjusted in real time using a PID control algorithm to maintain stable flight. In addition to basic stabilization, the system integrates depth-based perception and wireless telemetry, allowing the quadcopter to react to obstacles and transmit live flight data to a mobile application. This combination of control, perception, and communication makes GARUDA a practical and research-friendly UAV platform.

II. LITERATURE SURVEY

[1] "Autonomous Vehicle" by R. Beard et al.
The authors, through this scientific work, introduce the fundamental aspects of Autonomous Vehicles (AVs), probably the general ideas, system structure, and problems of enabling vehicles which might be robots on the ground, drones, or even submarines to function without the need for human intervention. It is a detailed welcome to a new domain, a technological revolution in which machines are capable of sensing, perceiving, localizing, planning their paths, and controlling themselves. The paper, dated back in time, captures the state-of-the-art early in the decade and probably sets out a next series of hurdles and research topics for achieving full autonomy in the system, which makes it an essential source for grasping the general background of UAV autonomy.

[2] "Nonlinear Complementary Filters on the Special Orthogonal Group" by R. Mahony, T. Hamel, and J. M. Pflimlin.
The paper revolves around creating a robust and a mathematically sound method that is adaptable to state estimation changes, with the attitude estimation of dynamic systems such as UAVs as the primary example. They present a Nonlinear Complementary Filter

(NCF) that is geometrically formulated on the Special Orthogonal Group (SO(3)), which is a mathematical representation of 3D rotations. This filter, therefore, aims at a very close to reality, drift-free fusion of multi-sensor data (e.g., gyroscopes, accelerometers, magnetometers) for attitude estimates. The manuscript is a landmark, as it furnishes the theoretical core for the modern IMU-based attitude determination, which is the energy source of the stability and control of any autopilot system, i.e., quadrotors.

[3] “A Survey on Open-Source Flight Control Platforms of Unmanned Aerial Vehicle” by E. Ebeid, M. Skriver, and J. Jin.
The authors in this paper, go through a detailed analysis of different open-source flight control units (autopilots) for Unmanned Aerial Vehicles (UAVs). The paper benchmarks these platforms with the help of important parameters like the hardware architecture (microcontrollers, sensors), software stack (operating systems, programming languages), supported UAV configurations (fixed-wing, multirotor) and the enabled features (GPS navigation, mission planning). This is a great resource that helps the researchers and the community of hobbyists to find the most suitable platform for their UAV projects by pointing out the trade-offs between different open-source options in terms of cost, flexibility, and complexity.

[4] “A Project of an Embedded Control System for Autonomous Quadrotor UAVs” by S. P. Madruga, A. de Holanda B. M. Tavares, A. V. de Brito, and T. P. Nascimento.
In this article, the authors explain how they designed and built a control system that is an integral part of the autonomous quadrotor UAV. The authors' hardware and software work is implied to be very detailed and comprehensive, including the choice of a microprocessor, IMU, GPS, and software architecture for the flight. Some of the main features of the system are the creation of a real-time operating system or scheduling, the execution of flight control algorithms, and the use of autonomous features such as waypoint navigation. The paper is a hands-on, project-based guide to the realization of a quadrotor that works, with emphasis on the integration issues involved in coupling software control loops with the physical dynamics of the flying platform.

[5] “Open-Source Project (OSPs) Platform for Outdoor Quadcopter” by S. Sabikan and S. W. Nawawi.
This document highlights the authors' effort in using and evaluating a platform of an Open-Source Project (OSP) for a quadcopter in the outdoor. Their work, most probably, entails the choice of a widely used open-source flight stack (for example, PX4, ArduPilot) and its setting up for a build of a custom outdoor quadrotor which is the primary emphasis is on the flight in the outdoor. Hence, it incorporates GPS and a barometer for positional hold and altitude control, and also performance testing of the platform under atmospheric conditions. The paper becomes instrumental in providing local and practical experience on the use of OSPs as a means towards the development of aerial vehicles which are reliable, economical, and can perform the tasks like surveying or surveillance.

[6] “Survey of Autopilot for Multi-rotor Unmanned Aerial Vehicles” by Z. Yang, F. Lin, and B. M. Chen.
The authors have compiled a detailed survey in this paper, which is a study of the autopilots that are used in multi-rotor Unmanned Aerial Vehicles (UAVs) only. Such a review may illustrate and evaluate the different facets of the on-board control system of multi-rotor vehicles, including the hardware parts, the control methods (like PID, LQR, backstepping), the techniques for state estimation, and the abilities for mission planning. The document serves as an essential reference to grasp the progress and present situation of the control systems for VTOL (Vertical Take-Off and Landing) devices, thus offering a comparative study of the different control strategies and their application for the attainment of a robust and agile multi-rotor flight
.
[7] “Build Your Own Quadrotor: Open-source Projects on Unmanned Aerial Vehicles” by H. Lim, J. Park, D. Lee, and H. J. Kim.
The researchers here delineate in detail and broadly review the conception of a working drone by the aid of open-source projects. It is highly probable the article illustrates all the stages and requirements of the hardware development, like designing the mechanical structure of the drone and choosing the right motor/propeller to combining the flight control board and setting up the open-source firmware. This project is a transparent vehicle for the learners and the developers, thus breaking down the drone building process with the ready-made parts and using the community-driven development of the most popular open-source autopilot stacks for flying.

[8] “Autopilot Design for a Target Drone using Rate Gyros and GPS” by S. Cho, S. Park, and K. Choi.
The authors describe in detail in their paper the engineering of an autopilot system for a target drone, which is generally a drone that needs extremely accurate and repeatable flight paths. The system uses rate gyros to achieve fast attitude stabilization and GPS for outer-loop navigation and position control. The article probably elaborates the control architecture and the tuning of the control parameters in great detail, as these aspects have a decisive influence on the performance level of a target drone. It describes the processing of sensor data to be used in the control loop and the design of the robust control laws that ensure the correct execution of the flight path.

[9] “Flight PID Controller Design for a UAV Quadrotor” by A. L. Salih and M. Moghavvemi.
The main topic of this paper is the creation and adjustment of a Proportional-Integral-Derivative (PID) controller used to maintain flight stability and control of a UAV quadrotor. PID control is the standard method of multirotor stabilization, and the authors probably go through the modeling of the quadrotor dynamics, and then a systematic way of finding the best PID gains for the inner (attitude/rate) and outer (position/velocity) control loops. Such a project is essential in learning how to secure stable flight by properly handling disturbances and issuing the correct thrust to the four motors.

[10] “Development of a Small UAV with Real-time Video Survei by C. Nam and S. Danielson.

The authors in this paper, present the development of a small Unmanned Aerial Vehicle (UAV) whose primary objective is to provide video surveillance in real-time. The focus of the work is on the integration of a vision system (camera, image processing) with the flight control system.
Some of the prominent technical aspects that were elaborated upon are: the system architecture for transmitting high-bandwidth video data in real-time, payload optimization, and flight planning for a surveillance mission. The paper is about the linkage of the control

system to a real-world application, thus, emphasizing the difficulties in handling the payload, communication, and power for visual monitoring tasks.

III. Hardware Design and Selection

3.1. Flight controller board:

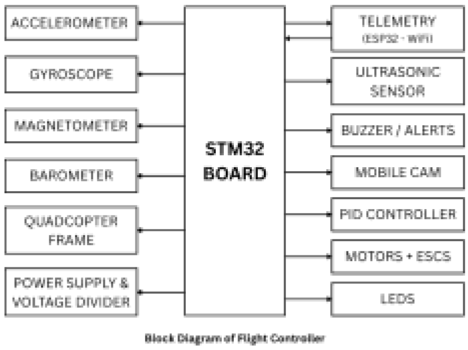

Fig 1. block diagram of flight controller

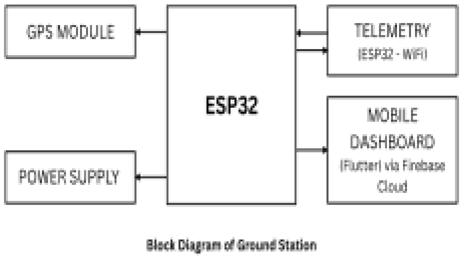

Fig 2.block diagram of ground station

3.1.1 STM32 microcontroller:

The STM32 microcontroller serves as the main processing unit of the GARUDA flight controller. It was selected due to its ability to handle real-time computations required for flight stabilization while maintaining low power consumption, which is critical for battery-operated UAVs. The microcontroller executes control algorithms, processes sensor data, and generates precise PWM signals for motor control.

Multiple communication interfaces such as UART, I²C, and SPI are utilized to connect sensors including the IMU, GPS module, barometer, and magnetometer. The STM32 processes this incoming data at high speed, enabling continuous updates of the quadcopter's attitude and position. Its internal timers and PWM peripherals allow accurate control of electronic speed controllers, ensuring smooth and responsive motor behavior. These features make the STM32 a suitable and reliable choice for implementing a custom flight control system.

3.1.2. Electronic speed controller:

The Electronic Speed Controller acts as the link between the flight controller and the brushless motors. Its role is to translate the low-power control signals generated by the STM32 into the high-current signals required to drive the motors. Since motor speed directly determines thrust, the overall stability and maneuverability of the quadcopter depend heavily on how accurately the ESC responds to control inputs.

Each motor in the case of a multirotor UAV is linked to its own ESC. This one-to-one relationship allows the estimation and control of each motor's speed to be done independently and accurately. A better example is a quadcopter whose four motors are regulated by four separate ESC hence the flight controller has the freedom to give different levels of thrust to the arms. The UAV is able to move in the different ways such as it can hover, soar, dive, yaw, turn, and pitch by simply changing the speed of the motors it chooses.

ESCs through their capability of having rapid response and flawless motor control also help to UAV stability concepts by providing an effective way of solving the instability generated by the wind or by sudden changes in the load. Nowadays, a lot of ESCs come with the features such as braking, current sensing, thermal protection as well as support for advanced communication protocols like PWM, Oneshot, and DShot thus ensuring maximum safety, efficiency, and ego performance. So, in short, these are the ESCs without which UAVs won't be available for flight with reliability and stability.

3.1.3 ESP32:

The ESP32 module is used as the communication backbone of the GARUDA system. While the STM32 focuses on real-time flight control, the ESP32 handles wireless data exchange and telemetry management. Its dual-core architecture allows it to process incoming sensor data, manage communication tasks, and transmit information without interrupting flight-critical operations.

Except for control, the ESP32 is also an excellent choice for communications roles because it comes with both Wi-Fi and Bluetooth hardware on board. Wi-Fi makes it a long-range, high-data-rate communication tool, thus, the ESP32 can be used for telemetry transmission, remote monitoring, and cloud-based data logging. On the other hand, Bluetooth (including BLE) is a short-range, low-power connection between mobile devices and ground control stations. Besides these wireless capabilities, the ESP32 can also be used for live data streaming and rudimentary video transmission purposes, thus allowing the system to be updated in real time and the operator's situational awareness to be improved..

3.1.4 LiPo battery:

A lithium polymer battery is used as the primary power source for the quadcopter due to its lightweight structure and ability to deliver high current when required. These characteristics are especially important during takeoff, rapid acceleration, and sudden maneuvering, where motors demand immediate bursts of power.

The selected LiPo battery provides stable voltage throughout the flight duration, ensuring reliable operation of motors, ESCs, sensors, and onboard electronics. Its compact form factor allows easy integration into the airframe without significantly increasing overall weight. Proper battery selection and power delivery play a key role in maintaining consistent performance and preventing unexpected voltage drops during demanding flight conditions.

LiPo batteries are very light and their production does not pose any limitations in terms of shape and capacity, which means that one can use them as building blocks of very small UAVs. They also help the entire setup in terms of voltage stability since they are delivering a constant voltage during the whole duration of the work cycle which in turn leads to the continuous and reliable performance of the motors, ESCs, and other electronics that are onboard the drone. Further to that, LiPo batteries have been designed to support high C-ratings, a term that refers to their capability to provide vast currents without significant voltage drop. These are the features that make LiPo batteries a must if one aims at clean, steady, and responsive drone flights that are not limited to maneuvers in the air but also go as far as the control signal level.

3.1.5 LEDs and buzzer:

LEDs are commonly used in UAV systems to provide visual indications of the flight controller's status. They display important updates such as system initialization, flight modes, GPS lock status, and battery charge levels. Different colors and blinking patterns help users quickly identify normal operation, warnings, or fault conditions. In some applications, LEDs are also integrated with sensors to assist in obstacle detection or to visually signal the presence of the UAV, improving situational awareness and safety.

A buzzer is included to generate audible alerts during critical situations such as low battery levels, system faults, or emergency conditions. Additionally, the buzzer serves as a simple deterrent against birds by producing sound when obstacles are detected nearby. This dual-purpose alert mechanism enhances operational safety and helps protect the UAV during outdoor flights.

3.2 Sensor Integration:

Integrating sensors like gyroscope, accelerometer, magnetometer, barometer and GPS module to monitor flight controller location and position

3.2.1 MPU6050:

The MPU6050 sensor functions as the primary inertial measurement unit of the quadcopter. It combines a three-axis accelerometer and a three-axis gyroscope to continuously measure linear acceleration and angular motion. This sensor provides the raw data needed to determine the orientation and movement of the UAV during flight.

Data from the MPU6050 is transmitted to the STM32 using the I²C interface and processed in real time. Since raw sensor readings contain noise and drift, filtering techniques are applied before the data is used by the control algorithm. Accurate IMU data allows the flight controller to maintain balance, execute smooth maneuvers, and respond effectively to disturbances.

3.2.2 Magnetometer:

The magnetometer is a device that detects the intensity and direction of the Earth's magnetic field. It is through this sensor that the flight controller gets to know the drone's compass heading and orientation with respect to the magnetic north. The magnetometer is the device that eventually enables the UAV to have a stable and accurate directional reference by giving precise yaw information, a feature that is extremely indispensable during navigation and autonomous flight.

During navigation relying on GPS, the magnetometer is the main player who supports the positional data given by the GPS module. So, while GPS is the provider of location, speed, and altitude, the magnetometer is the one that gives the directional heading. Hence, the flight controller is the one that has the exact alignment of the drone to meet the waypoints and follow the pre-set paths. With the help of sensor fusion algorithms, which combine data from the accelerometer, gyroscope, and magnetometer, the latter device becomes responsible for the improved accuracy of orientation, less drift, and better overall navigation reliability

.

3.2.3 Barometer:

The barometric pressure sensor (barometer) is a device that measures the atmospheric air pressure used to figure out the drone's altitude above the sea level. Since the air pressure gets lower as the height increases, the flight controller can determine the changes in altitude very accurately by looking at the pressure variations recorded by the barometer.

The barometer is a source of very accurate altitude information, especially for small changes in the vertical direction, which is a requirement of a stable flight. The device is a must for such operations as altitude hold, a smooth takeoff, and landing as well as hover stability. Using sensor fusion techniques where GPS altitude data and inertial sensor inputs are integrated, the barometer becomes even more precise for altitude measurement and thus facilitates the drone's vertical control which is not only stable but also reliable.

3.2.4 GPS module:

The GPS module provides real-time location information including latitude, longitude, altitude, and ground speed. This data enables the flight controller to track the UAV's position during flight and supports navigation features such as position hold and waypoint tracking.

GPS data is received through a serial communication interface and integrated with inertial and barometric sensor readings. This combined approach improves positional accuracy and allows the quadcopter to perform autonomous tasks with greater reliability. The GPS module is especially useful during outdoor flights where precise navigation and return-to-home functionality are required.

IV. CONTROL DEVELOPMENT ALGORITHM

Flight stability in the quadcopter is achieved using a PID control approach applied independently to the roll, pitch, and yaw axes. The controller constantly compares the desired orientation with the actual orientation measured by onboard sensors and calculates corrective motor commands based on this error.

The proportional component helps the system respond immediately to deviations, while the integral component compensates for long-term offsets that may arise due to sensor drift or external disturbances. The derivative component improves overall stability by reducing oscillations and preventing sudden overshoots. Considerable effort was spent tuning these parameters through testing to achieve a balance between responsiveness and smoothness. Proper tuning resulted in stable flight behavior under different operating conditions, including minor disturbances and changes in load.

V. TESTING

5.1 Ground Control Station (GCS):

The Ground Control Station (GCS) is the main user interface where the drone can be tracked and controlled during its mission. It shows the live telemetry data like location, altitude, speed, battery status, sensor readings, and flight modes. Operators can also change the flight parameters, PID values, waypoints, and the mode of the flight from manual, assisted, or autonomous through the GCS. Usage of a GCS elevates the accuracy, safety, and adaptability of the operation as it enables uninterrupted monitoring of the drone's performance and a prompt reaction to any sudden events in the air.

5.2 Firmware Development:

Firmware development is about writing very basic low-level embedded code that is instrumental in handling the flight controller's flight dynamics and must be capable of performing real-time sensor data processing. The firmware in question facilitates communication with the GPS module, barometer, magnetometer, accelerometer, and gyroscope while fetching data for each device through sensor fusion algorithms. The data is then used to calculate the control signals for ESCs and motors. Proper firmware architecture goes a long way in achieving the required timing, fast response, and system stability. A highly efficient firmware will provide the drone with correct navigation, stabilization of the flight, and smooth hardware to software communication all the way up to the GCS level.

VI. IMPLEMENTATION

The quadcopter flight controller system along with the hardware is very well thought out and the STM32 microcontroller has been taken as the main processor which will handle the main tasks like processing data from the sensors and stabilization of the flight. The STM32 was decided as a suitable candidate because of its speed, low power consumption, and different communication ports making it a perfect choice for a UAV of a complicated nature.

The entire set of the most important sensors has been brought together with one another into the STM32 microcontroller. They keep on reporting live data about orientation, altitude, heading, and position and also the distance to an obstacle or an object. This re-communication is done by standard serial protocols like UART. Here, the fusion algorithm is used for the sensor data, where information from different sensors is taken to work out the best and the most noise-free estimations of the attitude and position of the quadcopter.

A PID control algorithm has been employed to handle a quadcopter's roll, pitch, and yaw movements. The PID controller, to the control it is given, finds the difference between the desired setpoints and the current values and then it calls for the ESCs and the motors to be given the necessary control signals. The code for this control automation has been written and checked in the Arduino IDE. This is fine enough a place for development and testing and also supports flexible firmware development. The PID's well-tuned parameters bring about the desired effects of the system as its response to commands becomes smooth, with a minimum of oscillations and at any flight conditions, the stability is kept.

An ultrasonic sensor is the one that has been selected to serve for both the purposes of obstacle detection and avoidance by measuring the distance between the sensor and objects around. I.e., when an obstacle comes within the pre-given safety threshold, the system reacts immediately by activating the user alert via LED indicators and a buzzer to be able to visually and audibly provide warnings. At this very moment, the flight controller processes the obstacle data and sends the quadcopter new flight instructions to carry out the safe route already stored or to go find another safe route. This serves as the autonomous response which is handy in the prevention of accidents, the protection of the flight controller, and finally, the safe and efficient delivery of the quadcopter to its destination.

VII. RESULTS

The proposed quadcopter flight controller using the STM32 microcontroller and integrating the Depth Analysis Model for obstacle detection. The results include hardware testing, sensor validation, controller performance analysis, communication verification, and overall system behaviour under different flight conditions.

7.1 HARDWARE TESTING AND VALIDATION

Each sensor was individually validated to ensure accurate data acquisition:

7.1.**1 Sensor Module Testing:**

1. MPU6050/ ICM Sensor Testing:

The first step in the sensor fusion work revolved around rigorous testing and validation of the MPU6050/ICM sensor ensemble, which is the central unit of the quadcopter's Inertial Measurement Unit (IMU). This phase was necessary to ensure that the IMU could accurately detect motion since the IMU directly affects the stability of the flight and the performance of the control. Testing was initiated by accessing and observing the raw accelerometer and gyroscope outputs through the serial monitor.

By capturing these raw numbers, baseline sensor behavior, bias, drift, and noise characteristics were established and tested both in the static and dynamic scenarios.

They found out that the raw data from sensors were very noisy, especially from the accelerometer, and the gyroscope data drifted over time. Such unprocessed data cannot be used directly in a flight control system because it leads to unstable attitude estimation and thus controlled flight becomes impossible. To solve this problem, a Complementary Filter was put into operation.

The complementary filtering method selectively uses the advantages of both sensors: the gyroscope which gives precise short-term angular rate information with quick response and the accelerometer which gives a dependable long-term orientation reference based on gravity but is sensitive to noise and vibrations. After weighting and fusing these two data sources, the filter not only compensates for gyroscope drift but also suppresses accelerator noise.

Consequently, the complementary filter yielded continuous and trustworthy estimates of pitch, roll, and yaw, to a great extent, correct the attitude estimation. This upgrade was necessary to implement stable flight control as it gave the PID controller clean and consistent orientation data making the resulting stability, responsiveness, and overall flight performance better.

Table 7.1.1.1: Raw accelerometer and gyroscope values observed in serial monitor

Roll	Pitch	Yaw	Distance
6.54	1.87	-129.63	216.80
6.61	1.86	-129.77	241.45
6.68	1.85	-129.48	217.26
6.72	1.84	-130.21	182.57
6.78	1.84	-129.59	241.75

The figure 4.1.1 illustrates the device output after the filter and control loop have been successfully implemented. It displays Motor Speeds (M1 to M4) along with the Telemetry data, which are the filtered roll, pitch, yaw, and distance readings.

As an instance, the telemetry always indicates pitch around 1.850 and roll around 6.70, thereby showing the stability of the attitude measurement. The changes in motor speed that go along with it depict the PID controller which is actively holding these values while it is also responding to the changes in distance (for example, 216.80 to 241.45). This was a confirmation that the sensor fusion was the source of the reliable attitude reference which was needed for the functioning PID stabilization.

2. Barometer/Altimeter Testing

- Measured altitude variance at static position.
- Maximum drift observed: ~3–5 cm under stable indoor conditions.

Fig 7.1.1:sensor integration on beard board

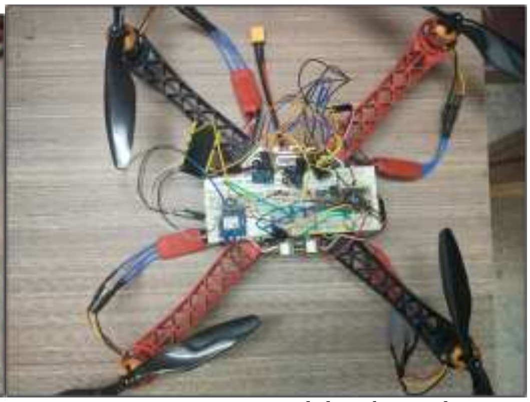

Fig 7.1.2: control algorithm implementation

3. Depth Analysis Model

The real-time output of the Depth Analysis Model is shown in the image. This model is a major part of the drone's autonomous navigation system. It gets depth frames and turns them into a 3x3 matrix, where each component is the approximate distance in meters of that space from the drone.

This matrix is what the Decision Logic uses. The logic scans the distance data—most notably the shortest distances in the bottom row (e.g., 0.85 m in the first frame)—to be compared with a safety threshold set.

It is through the analysis that the camera system decides the necessary evasive action. In the first four frames, the minimum distance is on the right side of the drone's path, so the logic is producing the output manoeuvre required: Direction to LEFT, all the time. Likewise, in the two frames following the shortest distance is on the both sides of the drone's path, so the logic is producing the output manoeuvre required: Direction STRAIGHT, all the time. Moreover, the last two frames indicate that the shortest distance is on the left side of the drone's path, so the logic is consistently directing the required manoeuvre: Direction to RIGHT.

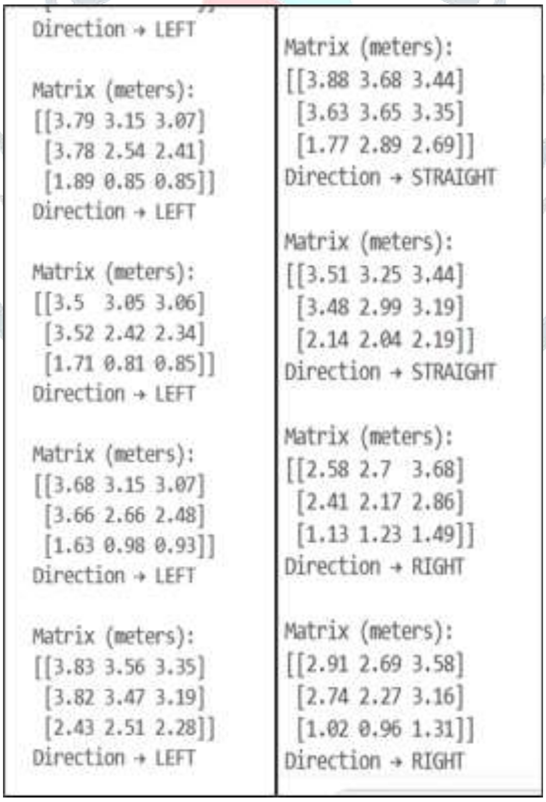

Fig.7.1.1.1. Estimated object distance derived from depth analysis model

7.2 MOTOR AND ESC RESPONSE ANALYSIS

7.2.1 PWM Output Verification

STM32 timers have been set up to produce 50 Hz PWM signals, which is the normal frequency for the control of most UAV ESCs. The duty cycles of these PWM signals were translated to the range of 1000–2000 µs, which represents the minimum and maximum throttle commands for accurate motor speed control.

In order to check the correctness and the stability of the PWM signals thus generated, the outputs were visually inspected via an oscilloscope. The readings showed that the PWM pulses were stable and did not exhibit any jitter, thus allowing the motors to be controlled in a reliable and consistent manner. This stable PWM generation is necessary, among other things, for smooth flight, accurate maneuvering, and the correct behavior of the PID-controlled flight system.

7.2.2. ESC and Brushless Motor Testing

The essential components responsible for generating thrust—the Electronic Speed Controllers (ESCs) and Brushless Motors—underwent thorough individual testing to ensure they met the performance and reliability requirements of the quadrotor system.

Testing Procedure and Results

As illustrated in figure 7.2.1. provided hardware diagram, the motors are powered via the ESCs, which in turn receive regulated power from the Li-Po Battery through the Power Distribution Board (PDB).

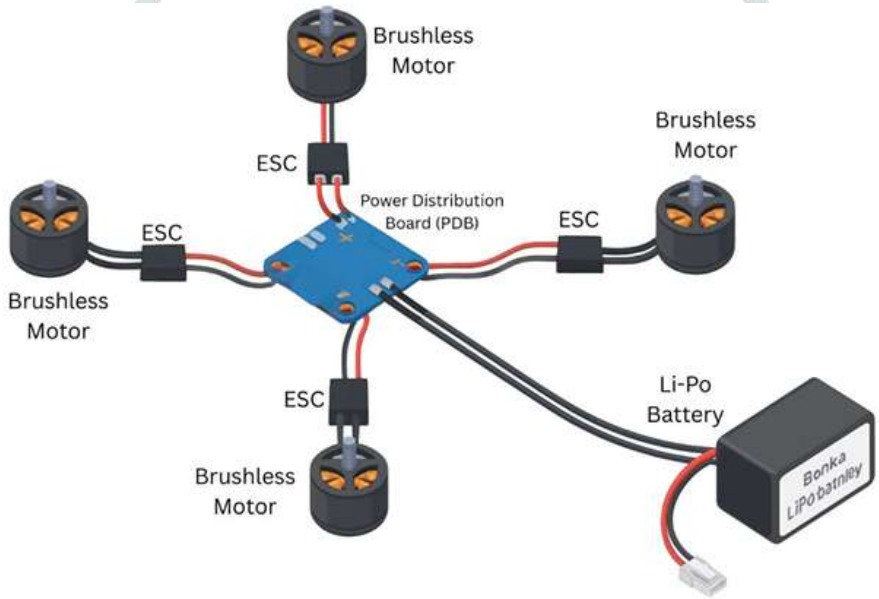

Fig.7.2.1. Power Train (ESC and Brushless Motor) Configuration

Individual Motor Testing and Powertrain Validation (Elaborated):

Prior to the full assembly of the quadcopter, each of the four motors underwent a detailed individual testing routine in a systematic manner. This method made it possible to detect and correct any performance issues, defects, or irregularities that might have appeared in the individual motors or ESCs at an early stage of the process. The separate testing of motors allowed the verification of the reliability of each propelling unit, which is very important, as the stability and safety of the whole UAV may be jeopardized by just one malfunctioning motor.

Response Time Measurement:

The response time of the motors, i.e., the time the motor takes to go from idle to full throttle, is a very important parameter of flight dynamics for UAV. The response time that was recorded during the testing was always between 120 ms and 150 ms, thus showing that the motors are capable of very fast thrust changes. Fast response times, such as these, are a must when high-frequency control loops are used, for example, the PID controller, which is how the flight controller gets to do the quick corrections of roll, pitch, and yaw. Stable flight is thus guaranteed even if the UAV executes a sudden maneuver or experiences a disturbance like a wind gust. Otherwise, delayed reactions, instability, or oscillations could be exhibited by the UAV.

Continuous Operation Reliability:

Long runtime tests were performed to assess the ability of the motors and ESCs to endure continuous operation. The tests revealed that neither the motors nor the ESCs got overheated, thus allowing the assumption that the parts can take the electrical load and the mechanical stress of the operation during normal flight conditions. In addition, the PWM signals and the motor responses were fully in tune at all times as the perfect synchronization was maintained without any synchronization or timing drift. This was a confirmation of the propulsion system's durability and it showed that it is capable of ensuring the steady performance under extended operation.

7.3. PID CONTROLLER TUNING RESULTS

During the first flight test, the quadcopter was showing oscillations in the roll, pitch, and yaw axes due to an incorrectly tuned PID. The P and D gains of the PID controller were thus adjusted carefully with the help of the proportional (P) and derivative (D) response of the PID controller.

The proportional gain was increased as a result of which the system's responsiveness to error was improved, whereas the derivative gain tuning helped to oscillations to be dampened and overshoot to be reduced. Thus the initial instability, which was the root of the problem, was reduced to a large extent, resulting in the smoother and more controlled flight behavior.

Furthermore, the integral (I) term of the PID controller was susceptible to windup, where the accumulated error could cause overshooting or a slow response after a long period of disturbance. Therefore, an anti-windup mechanism was installed to stop this by limiting the integral accumulation when the actuator (motor output) is saturated. This made sure that the integral component remained effective in correcting steady-state errors without causing instability. The PID controller with these adjustments was capable of stable and accurate control, thus the quadcopter could keep its balance, respond correctly to pilot inputs or autonomous commands, and be able to perform steadily even when the flight conditions were different.

7.4. DEPTH-BASED OBSTACLE DETECTION RESULTS
7.4.1. Distance Measurement Accuracy

The obstacle detection unit employed an ultrasonic sensor that was off by an average of 2-4% when measuring distance. So, this device achieved a high-level proximity sensing. This level of precision is adequate for safe navigation and collision avoidance in normal flight scenarios.
A system like this is perfect for detecting static objects; as well as slow-moving obstacles. The flight controller thus receiving the command to change the UAV's path or to issue alerts. Still, very fast-moving objects may need some additional sensing strategies or faster update rates to detect them reliably. In general, the obstacle detection system was a source of dependable and timely data, which it used to ensure the quadcopter operated safely and autonomously.

7.4.2. Real-Time Obstacle Avoidance Test

FIG 7.4.2.1. obstracle avoidance test

Figure 4.4.1 shows the Depth Analysis Model working live with the Grid-Based Classification method for the removal of obstacles that was the main part of the system checked during the Real-Time Obstacle Avoidance Test. This display demonstrates how the operation reads the input from the instruments and makes the navigation decisions within the shortest time period.

The left side of the figure gives the camera feed separated into a 3x3 spatial grid (or some other grid with a similar structure), where every cell matches a portion of the camera's field of view. The system calculates the depth distance in meters for each grid cell and prints the numbers. For instance, a hand that is right in front of the drone may only be 0.66 m away, while the background objects can be at a distance of 3.82 m. This spatial partitioning enables the system to locate obstacles and determine their relative distances in various parts of the camera frame.

The right panel displays a depth map using color coding, where the colors stand for the closeness of the objects—red or yellow for the nearest objects, and blue or green for the farthest ones. This representation reveals the surroundings at a glance to the operator as well as the flight controller, making it easier for them to understand the spatial relationships.

If the system detects a depth that is lower than the safety threshold set beforehand, it will, for instance, the 0.66 m measurement for the hand, very quickly send a signal for the avoidance. The microcontroller cuts the forward motion and starts the corrective maneuvers upon receiving the avoidance signal. The correction is carried out in a manner that is aware of the grid, i.e. the system determines the direction having the clearest path (e.g. turning left or right) with the help of depth values in the neighboring grid cells.

It is such a method that reveals how the system interprets depth data in the present moment and, thus, functions as a safety shield for obstacle-free and, generally, human intervention-free navigation. By uniting depth analysis with grid-based classification, the quadcopter is not only accurate in detecting but also quick in correcting its path which is very important in autonomous and obstacle-loaded environments.

7.5. MOBILE APPLICATION VALIDATION: GARUDA_PHASE

The mobile app GARUDA_PHASE which is based on Flutter was put through a series of tests to prove that it can be used as the main Ground Control and Monitoring Station (GCS) for the quadcopter system. The validation ensured telemetry in real-time from the device to the monitoring station, as well as sensor data figures and waypoint recording made robust and accurate by the logging of the data stream from the flight controller. Such a thorough testing procedure was necessary to confirm that the software would be able to facilitate the monitoring and command of the flight as if in real operational conditions.

The application is divided into five parts, each referring to one of the application's functions:

- Dashboard: Gives an overview of the UAV's state that includes attributes such as remaining battery, system health, as well as flight mode.
- Sensors: Offers both original and filtered sensor readings from modules such as the IMU, GPS, barometer, and magnetometer, thus providing on the spot monitoring of the quadcopter’s orientation, altitude, and position.
- Camera: Enables the transmission of images or videos along with the depth information captured through the use of cameras located on the UAV, hence, helping the operator in obstacle avoidance and the surveillance process.
- Graphs: Manifesting of the historical data acquired from the UAV like the PID controller and sensor inputs, is the system's tool for analysis, tuning, and performance optimization.
- GPS: A feature that supports navigation and also allows the setting of autonomous flying paths via waypoint management.

Upon validation, the rendering of all modules was successful, and they were also able to synchronize different data streams, thus proving that the application is capable of accommodating multiple inputs from real-time sources simultaneously without any delay or loss of data. Such a confirmation guarantees the solidity of the interface thus the operators are assured of getting usable and timely feedback which is of great help in debugging, decision-making, and flight management.

GARUDA_PHASE is now validated and this accomplishment paves the way for the subsequent phase of the system test and its eventual deployment when it will become the central platform for UAV operations in autonomous or semi-autonomous mode being monitored, controlled, and managed.

7.5.1. Dashboard Module

The user interface shows a complete picture of the system and threat diagnostics, which is very helpful in real-time monitoring the UAV performance and its interaction with the environment. The display has been divided into several blocks, each of them showing the essential information:

- Critical-Threat Panel: This is a feature that gets activated when an object is detected within around 4 cm thereby indicating the source of the immediate possible danger. In this moment the taser and buzzer icons are highlighted, thus, the operator is informed of his/her presence, and the laser indicator remains at the standby. Hence, it is possible to recognize the situation quickly and take a short time response action in order to alleviate the threat.
- Flight Status Panel: Shows the drone’s current flight condition whereby “Cruising” is displayed along with moderate stability thus making a rapid check of overall flight performance and responsiveness possible.
- Orientation Block: Displays pitch, roll, and yaw live with different colors of the respective indicators if there are deviations or sudden movements, which help the operators to visually check UAV attitude and assure that it is flying steadily.
- Threat-History Block: Keeps the record of recent happenings whereby it mentions the number of critical, high, and moderate threat situations, thus providing a basis for the analysis of dangers that reoccur and the UAV interaction with the environment over time.
- Flight-Intelligence Block: Indicates the state of UAV movement, the distance to the nearest object, and the average detection range. This briefed data serves to the operator as both the immediate and the overall environmental context hence helping to make right decisions whether during autonomous or manual flight.

There is no doubt that these elements together provide a real-time, safekeeping, situational-awareness-enhancing, and intelligence-boosting interpretive viewpoint of the UAV in operational scenarios and of environmental threats at the same time.

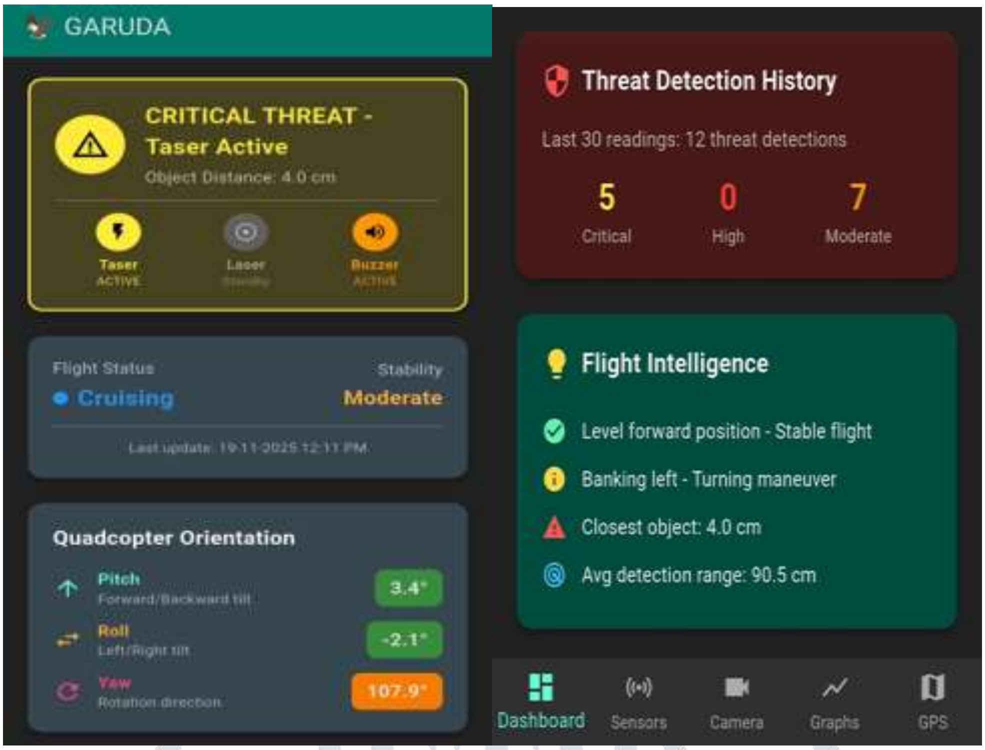

FIG.7.5.1. DASHBOARD MODULE

7.5.2. SENSORS MODULE

The system provides a time-stamped sensor log that records the latest 30 readings, offering a detailed history of UAV sensor activity and threat interactions. Each entry in the log contains critical flight and environmental data, including ultrasonic distance measurements and the UAV's pitch, roll, and yaw values, allowing operators to analyze flight dynamics and obstacle proximity over time.

- **Critical-Threat Entries**: These log entries occur when an object is detected at approximately 4 cm, indicating an immediate hazard. In these cases, the Taser is active, signaling a high-priority threat and documenting the UAV's response.
- **Moderate-Threat Entries**: Objects detected at distances between ~40–50 cm are recorded as moderate threats, highlighting potential obstacles that require attention but do not demand immediate evasive action.
- **Non-Threat Entries**: Readings showing distances greater than 50 cm are logged as non-threat events, representing safe flight conditions with no immediate hazard.

The log viewer supports per-entry expansion, allowing operators to inspect detailed values for each reading, including exact pitch, roll, yaw, and distance measurements. This feature enables thorough analysis of the UAV's performance, sensor accuracy, and threat response behavior.
Overall, the sensor log provides a comprehensive, historical record of flight and environmental data, which is invaluable for debugging, system optimization, and post-flight analysis, ensuring better understanding and improvement of UAV operations.

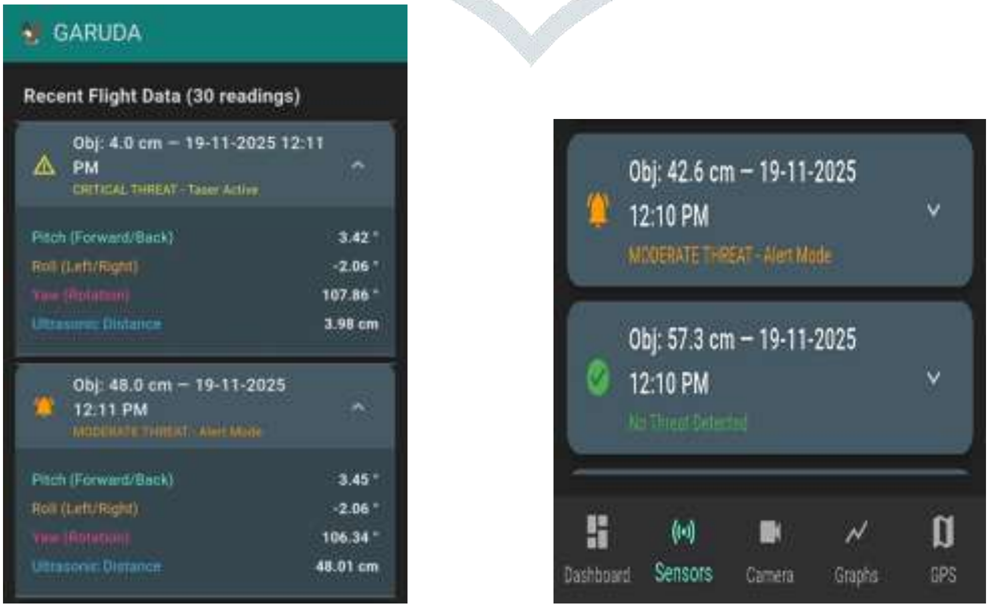

Fig.7.5.2. Sensors Module

7.5.3. Camera Module

Provides the live stream feed of the quadcopter camera in the app as well as an option to open the camera in Browser.

FIG.7 .5.3. CAMERA MODULE

7.5.4. GRAPH MODULE

The system is equipped with a graphical component which records pitch, roll, yaw, and ultrasound distance against time, in order to provide a UAV motion dynamic as well as obstacle proximity visualization. This multiple colored line graph lets the operators quickly spot the trends, changes, and also the irregularities in the sensor data.

Graph Interpretation: The yaw and ultrasonic distance lines demonstrate very quick changes, indicating fast rotational movements and immediate changes in the proximity of the obstacles. On the other hand, the pitch and roll lines change slowly and correspond to UAV attitude and stabilization adjustments. This visual distinction enables the operators to judge the responsiveness and stability of the flight control system.

Tooltip Panel: A tooltip panel located underneath the graph shows the exact sensor values for any selected timestamp, thus enabling a precise UAV state analysis at the most critical moments. This functionality is especially valuable for PID tuning, post-flight diagnostics, and obstacle avoidance confirmation.

In fact, the graph component is an easy, real-time way to see the changes in flight dynamics and environmental interactions which, in turn, helps the operator to performance monitoring, anomaly detection, and system behavior optimization.

Fig.7.5.4. Graph Module

7.5.5. GPS Module

The mobile application offers a geospatial movement and altitude logging interface, which tracks the UAV's position and elevation over time. This unit allows flight paths to be recorded, altitude changes to be monitored, and navigation performance to be analyzed with very accurate results.

Waypoint Recording: In fact, each waypoint insertion is also a timestamp and it registers the latitude and longitude, thus giving a very detailed path of the UAV's trajectory. The altitude values in the listed entries vary from about 890 m to 895 m, showing that the flight altitude has changed slightly.

Distance Measurement: The tool records the distance between the next waypoints, that is, the distance is usually from 0 m to 1 m in the entries shown, thus the precision of the flight and the consistency of movement can be verified.

Summary Panel: A separate panel gathers the essential pieces of information concerning the flight such as the total distance covered (10 m), the highest altitude (895 m), the mean altitude (810.4 m), and the total number of points (12). This brief gives to the operators the performance of the UAV and the amount of the area covered at a glance.

The module's operation was a success, which means that the telemetry stack, the user interface logic, and the data binding between the STM32 flight controller outputs and the Flutter-based GARUDA_PHASE interface are working smoothly. The data is sent and displayed without being dropped, jittered, or having display errors, which is a confirmation of the system being ready for real-time flight monitoring, navigation analysis, and operational decision-making.

Fig.7.5.5. GPS Module

VIII. References

[1] H.Y. Chao,Y. C. Cao, and Y. Q. Chen, “Autopilots for Small Unmanned Aerial Vehicles: A Survey,” International Journal of Control, Automation, and Systems, vol. 8, no. 1, pp. 36–44, 2010.

[2] E. Ebeid, M. Skriver, and J. Jin, “A Survey on Open-Source Flight Control Platforms of Unmanned Aerial Vehicle,” in Proc. Euromicro Conference on Digital System Design, 2017.

[3] S. P. Madruga, A. de Holanda B. M. Tavares, A. V. de Brito, and T. P. Nascimento, “A Project of an Embedded Control System for Autonomous Quadrotor UAVs,” in Brazilian Symposium on Robotics, 2018.

[4] S. Sabikan and S. W. Nawawi, “Open-Source Project (OSPs) Platform for Outdoor Quadcopter,” Journal of Advanced Research Design, vol. 24, no. 1, pp. 13–27, 2016.

[5] Z. Yang, F. Lin, and B. M. Chen, “Survey of Autopilot for Multi-rotor Unmanned Aerial Vehicles,” in Proc. IECON 2016 – 42nd Annual Conference of the IEEE Industrial Electronics Society, pp. 6122–6127, 2016.

[6] H. Lim, J. Park, D. Lee, and H. J. Kim, “Build Your Own Quadrotor: Open-source Projects on Unmanned Aerial Vehicles,” IEEE Robotics & Automation Magazine, vol. 19, no.3, pp.33–45, 2012.

[7] S. Cho, S. Park, and K. Choi, “Autopilot Design for a Target Drone using Rate Gyros and GPS,” International Journal of Aeronautical and Space Sciences, vol. 13, no. 4, pp. 468–473, 2012.

[8] A. L. Salih and M. Moghavvemi, “Flight PID Controller Design for a UAV Quadrotor,” Scientific Research and Essays, vol. 5, no. 23, pp. 3660–3667, 2010.

[9] C. Nam and S. Danielson, “Development of a Small UAV with Real-time Video Surveillance,” American Society for Engineering Education, 2011.

[10] H. Wu, D. Sun, and Z. Zhou, “Micro Air Vehicle: Configuration, Analysis, Fabrication, and Test,” IEEE/ASME Transactions on Mechatronics, vol. 9, no. 1, pp. 108–117, 2004.