

FPGA Enabled Flight Control Unit

Radhika Gopal Sharma^{1,*}, Janhvi Ratnakar Pande², Amish Ajay Naidu³, S.S. Lokhande⁴

Abstract

The FPGA flight controller revolutionizes drone technology with unmatched stability and precision. Integrated with an IMU and advanced sensors like the MPU 6050, it ensures accurate positioning critical for reliable UAV operation. Its efficient real-time data processing and seamless sensor communication enhance performance and reliability across various conditions. This advancement sets a new standard in UAV technology, suitable for applications ranging from surveillance to disaster relief. An aircraft's flight control system (FCS) is a crucial part. It enables the pilot to keep an eye on the aircraft's attitude, direction, and flying forces while in flight. A flight control system controls the inputs required to move the control surfaces so that the pilot can keep an eye on the aircraft. They consist of the controls for the aircraft engine, the cockpit, the connecting wires, and the essential operational mechanisms. Because there are no additional backup systems for a standard mechanical flight control system, a system failure can have disastrous consequences. As a result, designing a system with a significant degree of redundancy and fault tolerance is essential. Therefore, the primary goal of this research is to use a field programmable gate array (FPGA) to develop and implement a flight control system. The FPGA-based flight control system (FCS) has various beneficial applications and outperforms both the DSP chip-based UAV flight management system and the traditional microcontroller in terms of functionality. Four modules make up the overall flight control system: the Avalon bus control component sensor driver component, flight control module, and navigational instruction management module.

Keywords: FPGA, Basys 3, UAV, flight control unit, IMU, MPU 6050, sensor data processing

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become integral across industries for their versatility in tasks like surveillance, photography, and delivery. However, traditional microcontroller-based flight control systems often struggle in complex environments, driving the need for more sophisticated solutions. Field Programmable Gate Arrays (FPGAs) offer a promising alternative due to their reconfigurability and parallel processing capabilities, which can significantly enhance UAV performance.

*Author for Correspondence

Radhika Gopal Sharma
E-mail: radhika1980s@gmail.com

¹⁻³Student, Department of Electronics & Telecommunication, Sinhgad College of Engineering, Savitribai Phule University, Pune, Maharashtra, India

⁴Professor, Department of Electronics & Telecommunication, Sinhgad College of Engineering, Savitribai Phule University, Pune, Maharashtra, India.

Receiving Date: May 02, 2024

Accepted Date: June 04, 2024

Published Date: June 10, 2024

Citation: Radhika Gopal Sharma, Janhvi Ratnakar Pande, Amish Ajay Naidu, S.S. Lokhande. FPGA Enabled Flight Control Unit. Journal of Aerospace Engineering & Technology. 2024; 14(1): 19–25p.

To address these challenges, we propose an FPGA-enabled flight control unit for UAVs. FPGAs provide advantages such as parallel processing, reconfigurability, and high-speed data processing. This system can efficiently process sensor data from the Inertial Measurement Unit (IMU) in real-time, ensuring precise UAV positioning and orientation. Moreover, the flexibility of FPGAs allows for the implementation of complex control algorithms, enhancing stability, precision, and adaptability.

This project aims to develop a robust and efficient flight control unit using FPGA technology to improve control stability and precision for UAVs. The system is designed to be adaptable for a wide

range of UAV applications, from aerial photography and surveillance to environmental monitoring and disaster response. Comparative analysis with traditional microcontroller-based systems highlights the superior performance of the FPGA-based approach, demonstrating its potential to significantly enhance UAV performance and reliability for more sophisticated and versatile drone applications.

LITERATURE SURVEY

A comprehensive literature survey was conducted to explore existing solutions and advancements in the field of UAV flight control systems, with a specific focus on FPGA-based implementations and control methodologies. FPGA-based flight control systems have garnered attention due to their parallel processing capabilities and reconfigurability. Monterrosa et al. [1] presented a UAV flight controller design based on a state machine approach, highlighting the advantages of using FPGAs for real-time control and flexibility in system design.

Control and navigation methodologies are crucial for ensuring stable and precise flight in UAVs. Hongxi Yang and Qingbo Geng [2] discussed the hardware architecture and control navigation methodology of a flight control system for small UAVs with static stability, based on ARM and FPGA. Their approach utilized multiple sensors, including MEMS gyros, accelerometers, GPS, and air pressure sensors, to enhance measurement precision and ensure altitude hold and desired route navigation.

Sensor fusion techniques are essential for accurately estimating the UAV's attitude. Yu et al. [3] proposed a low-cost MEMS-based navigation and control system for micro air vehicles, employing sensor fusion algorithms to combine data from inertial sensors and GPS for improved accuracy in position and posture estimation. These studies provide valuable insights into the design and implementation of FPGA-based flight control systems for UAVs, emphasizing the importance of sensor fusion, control strategies, and hardware architecture in achieving stable and reliable flight performance.

METHODOLOGY

The development of the FPGA-enabled flight control unit involves the following steps:

1. Design and development of the FPGA (Basys 3) architecture.
2. Integration of the IMU sensor (MPU-6050) for real-time positioning and orientation data.
3. Implementation of efficient algorithms for sensor data processing and flight control.
4. Testing and validation of the system in various UAV applications.

The development of the FPGA-enabled flight control unit for UAVs involves a systematic approach that encompasses hardware design, sensor integration, algorithm implementation, and system testing. The methodology is divided into several key steps:

Hardware Design and Setup

The first step involves designing the hardware architecture of the flight control unit. An FPGA board is selected based on its processing capabilities, I/O ports, and compatibility with other components. The FPGA board is then interfaced with an Inertial Measurement Unit (IMU) such as the MPU-6050, which provides accelerometer and gyroscope data for determining the UAV's orientation and movement [4–5].

1. *FPGA Board Selection:* The FPGA board is the central component of the flight control unit. It is responsible for processing sensor data and executing control algorithms in real-time. When selecting an FPGA board, several factors must be considered: - Processing Capability: The FPGA should have sufficient processing power to handle real-time data processing and complex control algorithms. - I/O Ports: The board should have adequate input/output ports for interfacing with sensors, communication modules, and other peripherals. - Compatibility: The FPGA should be compatible with the development tools and software used for programming and simulation.
2. *IMU Sensor Integration:* The IMU sensor, such as the MPU-6050, is a critical component for measuring the UAV's motion and orientation. It typically includes a 3-axis accelerometer and a

- 3-axis gyroscope. The integration of the IMU sensor with the FPGA board involves the following steps: - **Interfacing:** The IMU is connected to the FPGA board via I2C or SPI communication protocols. The physical connection is made using appropriate cables and connectors. - **Configuration:** The IMU sensor is configured to operate at the desired settings, such as sampling rate and sensitivity. This is usually done through software by writing configuration values to the sensor's registers. - **Calibration:** Before use, the IMU sensor must be calibrated to ensure accurate measurements. This involves compensating for sensor biases and scaling factors.
3. **Power Supply and Connectivity:** The FPGA board and IMU sensor require a stable power supply to function correctly. A power management system is designed to provide the necessary voltage levels and current to the components [6–8]. Additionally, connectivity options such as USB, Ethernet, or wireless modules can be integrated into the hardware setup to enable communication with a ground station or other devices.
 4. **Hardware Testing and Validation:** Once the hardware components are assembled, testing and validation are conducted to ensure that the setup functions as expected. This includes verifying the communication between the FPGA board and the IMU sensor, checking the accuracy of sensor readings, and ensuring that the FPGA can process data and execute algorithms without errors [9].

By carefully designing and setting up the hardware, the foundation is laid for the development of a robust and efficient FPGA-enabled flight control unit for UAVs.

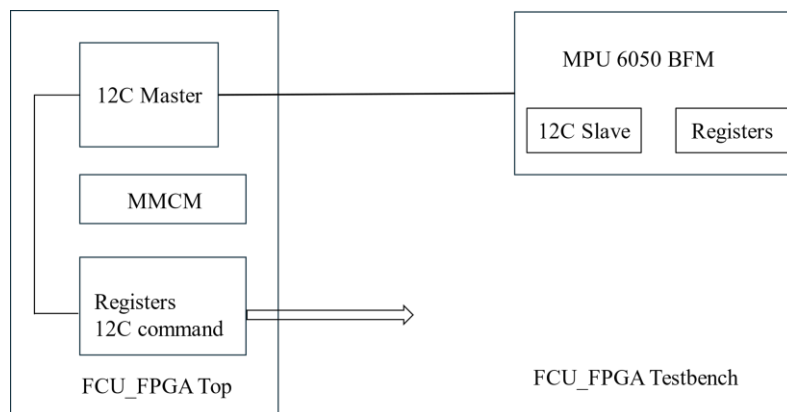


Figure 1. Block diagram of the control unit.

Sensor Integration and Calibration

The IMU sensor is integrated with the FPGA board, and calibration procedures are performed to ensure accurate readings. The calibration process involves adjusting sensor offsets and scaling factors to minimize errors in the measurements. The IMU data is then processed to obtain the UAV's roll, pitch, and yaw angles using the following formulas:

$$\text{Roll} = \arctan\left(\frac{\text{Accel}_y}{\sqrt{\text{Accel}_x^2 + \text{Accel}_z^2}}\right)$$

$$\text{Pitch} = \arctan\left(\frac{-\text{Accel}_x}{\sqrt{\text{Accel}_y^2 + \text{Accel}_z^2}}\right)$$

$$\text{Yaw} = \arctan\left(\frac{\text{Gyro}_z}{\text{Gyro}_x}\right)$$

where Accel_x , Accel_y , and Accel_z are the accelerometer readings, and Gyro_x and Gyro_z are the gyroscope readings along the respective axes.

Algorithm Implementation

Efficient algorithms are implemented on the FPGA for real-time processing of sensor data and flight control. These algorithms include sensor fusion techniques such as the Kalman filter or complementary filter to combine accelerometer and gyroscope data for accurate estimation of the UAV's orientation. The control algorithms are designed to maintain the desired flight path and respond to external disturbances. The implementation of algorithms on the FPGA is a crucial step in developing the flight control unit [10]. These algorithms are responsible for processing sensor data, estimating the UAV's orientation, and executing control commands to maintain stable flight. The algorithm implementation process involves several key components:

Sensor Fusion Algorithm

The sensor fusion algorithm combines data from the accelerometer and gyroscope to provide a reliable estimate of the UAV's orientation. Two common approaches are the Kalman filter and the complementary filter:

Kalman Filter: The Kalman filter is a recursive algorithm that provides an optimal estimate of the UAV's state by minimizing the mean square error. It consists of two steps:

The prediction step, which projects the current state estimate forward in time, and the update step, which incorporates new measurements to refine the estimate. The Kalman filter equations are as follows:

$$\begin{aligned}\hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_{k-1} \\ P_{k|k-1} &= AP_{k-1|k-1}A^T + Q \\ K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_kH)P_{k|k-1}\end{aligned}$$

where \hat{x} is the state estimate, P is the estimate covariance, K is the Kalman gain, z is the measurement, and A , B , H , Q , and R are matrices defining the system dynamics and noise characteristics.

Complementary Filter: The complementary filter is a simpler alternative to the Kalman filter that combines the accelerometer and gyroscope data by taking a weighted sum of their respective estimates. The filter equations are as follows:

$$\text{Angle}_{\text{filtered}} = \alpha(\text{Angle}_{\text{filtered}} + \text{Gyro}_{\text{rate}} \times \Delta t) + (1 - \alpha) \times \text{Angle}_{\text{accel}}$$

where α is the filter constant, $\text{Gyro}_{\text{rate}}$ is the angular rate from the gyroscope, Δt is the time interval, and $\text{Angle}_{\text{accel}}$ is the angle calculated from the accelerometer data.

Control Algorithms

The control algorithms are responsible for generating commands to maintain the desired flight path and respond to disturbances. Common control strategies include Proportional-Integral-Derivative (PID) control and Model Predictive Control (MPC). PID Control: The PID controller adjusts the control inputs based on the error between the desired and measured values. The control law is given by $u(t) = K_p e(t) + K_i \int e(t) dt + K_d \dot{e}(t)$, where $e(t)$ is the error, and K_p , K_i , and K_d are the proportional, integral, and derivative gains, respectively. Model Predictive Control (MPC): MPC is an advanced control strategy that uses a model of UAV dynamics to predict its future states and optimize the control inputs over a finite horizon. The optimization problem is formulated as:

$$\min_u \sum_{k=0}^{N-1} (x_k - x_{\text{ref}})^T Q (x_k - x_{\text{ref}}) + (u_k - u_{\text{ref}})^T R (u_k - u_{\text{ref}})$$

Subject to the system dynamics and constraints, where X_k and U_k are the state and control input at time step k , X_{ref} and U_{ref} are the reference state and input, and Q and R are weighting matrices.

FPGA Implementation

The algorithms are implemented on the FPGA using hardware description languages such as VHDL or Verilog. The implementation involves designing digital circuits that perform the necessary computations efficiently. The FPGA's parallel processing capabilities allow for realtime execution of the algorithms, ensuring fast and accurate control of the UAV.

By implementing these algorithms on the FPGA, the flight control unit can process sensor data and execute control commands in real time, enabling stable and responsive flight for the UAV.

System Integration and Testing

The final step involves integrating the flight control unit with the UAV's propulsion system and testing the system's performance in various flight scenarios. The testing phase includes evaluating the stability, precision, and responsiveness of the UAV under different conditions. Adjustments are made to the control algorithms based on the test results to optimize the system's performance.

The final phase of developing the FPGA-enabled flight control unit involves integrating the hardware and software components into a cohesive system and conducting comprehensive testing to ensure its performance and reliability.

Integration of Components

System integration involves connecting the FPGA board, IMU sensor, and other peripherals to the UAV's propulsion system. This step includes:

- Establishing communication between the FPGA board and the IMU sensor to ensure accurate data transmission.
- Interfacing the FPGA board with the UAV's motor controllers to enable control signal transmission.
- Integrating any additional sensors or communication modules required for specific applications.

The integration process requires careful attention to the electrical connections and data protocols to ensure seamless interaction between the components.

Software and Firmware Configuration

The FPGA board is programmed with the sensor fusion and control algorithms developed in the previous phase. The firmware is optimized for real-time performance and efficient resource utilization. Additionally, any necessary software for ground station communication or data logging is configured at this stage.

Functional Testing

Once the system is integrated, functional testing is conducted to verify that each component operates as expected. This includes:

- Testing the communication between the FPGA board and the IMU sensor to ensure accurate data transfer.
- Verifying the motor controllers' response to control signals from the FPGA board.
- Ensuring that any additional sensors or modules function correctly within the system.

[4] Flight Testing:

The most critical phase of testing involves evaluating the flight control unit's performance in real-world flight scenarios. Flight testing is conducted in a controlled environment and typically follows a structured approach:

Pre-flight Checks: Before each flight, a thorough inspection of the UAV and flight control system is performed to ensure that all components are functioning correctly. - Hover Test: The initial flight test involves hovering the UAV in a fixed position to assess the stability and responsiveness of the flight control system. - Manoeuvring Test: Subsequent tests involve executing various manoeuvres, such as turns, ascents, descents, and transitions between flight modes, to evaluate the system's control accuracy and adaptability. - Disturbance Rejection Test: The UAV is subjected to external disturbances, such as wind gusts, to assess the flight control system's ability to maintain stability and recover from perturbations.

Data Analysis and Optimization

After each flight test, the data collected from the sensors and flight control unit is analysed to assess the system's performance. Key metrics such as stability, control accuracy, and response time are evaluated. Based on the analysis, adjustments may be made to the control algorithms or hardware configuration to optimize the system's performance.

Safety Considerations

Throughout the integration and testing process, safety is a paramount concern. Adequate safety measures, such as emergency stop mechanisms, fail-safes, and adherence to regulatory guidelines, are implemented to ensure the safety of the operators, the UAV, and the surrounding environment.

By systematically integrating and testing the system, the FPGA-enabled flight control unit can be validated for its performance and reliability, paving the way for its deployment in various UAV applications.

Tech Stack Process Description

The tech stack for this project includes the following components: - FPGA Board: Used for realtime data processing and control. - IMU Sensor (MPU-6050): Provides accelerometer and gyroscope data for determining the UAV's orientation. - Sensor Fusion Algorithm (Kalman Filter/Complementary Filter): Combines sensor data for accurate orientation estimation. - Control Algorithms: Designed to maintain stability and respond to disturbances.

The development process involves programming the FPGA with VHDL or Verilog, integrating the IMU sensor, implementing the algorithms, and testing the system on a UAV platform.

CONCLUSION

The developed FCS is more efficient than the FCS that was utilised in the past, such as the mechanical and microprocessor-based ones. The flight control systems that rely on mechanics were discovered to be weighty. They also needed the cables to be routed carefully. The FCS became heavier because redundant backup was required because the routing had to be done carefully. Less onboard computing power is available for autonomous real-time image processing for course planning and avoidance in microcontroller-based flight control systems.

Acknowledgement

The authors would like to thank the Department of Electronics and Telecommunication Engineering at Sinhgad College of Engineering, Pune, for their support and guidance throughout this project. Special thanks to our external guide Mr. Abhilash M. T. and internal guide Dr. (Mrs) S. S. Lokhande for their invaluable insights and expertise.

REFERENCES

1. Monterrosa, L., Cano, S., & Martínez, M. (2016). Design, Development and Implementation of a UAV Flight Controller Based on a State Machine Approach. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 5(10), 83238332.

2. Yang, H., & Geng, Q. (2011). The Design of Flight Control System for Small UAV with Static Stability. In 2011 IEEE International Conference on Mechatronics and Automation (pp. 799803). Beijing, China.
3. Yu, G., Wang, J., & Li, S. (2008). A Low-cost MEMS-based Navigation and Control System Design for Micro Air Vehicles. *Journal of Navigation*, 61(4), 701-718.
4. DeSantis, Christopher, & Hussein, Ahmed Refaey. (2022). AI SoC-Based Accelerator for Speech Classification. *IEEE Canadian Journal of Electrical and Computer Engineering*, 45(3), 222-231.
5. Kamal, Ashraf M., Bayoumy, A.M., & Elshabka, A.M. (2016). Modeling and flight simulation of unmanned aerial vehicle enhanced with fine tuning. *Elsevier Journal of Aerospace Science and Technology*, 51, 106-117.
6. Henriques, L. C. Florencio, & de Souza, C. Protásio. (2021). Development of an Embedded Longitudinal Flight Control Based on X-Plane Flight Simulator. *IEEE Latin America Transactions*, 19(10), 1684-1691.
7. Khachumov, Mikhail, Talalaev, Alexander, & Pankratov, Alexander. (2022). Onboard Computer Function Implementation Based on Cordic Algorithm. In 2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) (pp. 795800).
8. Li, Yilan, Li, Hongjia, Li, Zhe, Fang, Haowen, Sanyal, Amit K., Wang, Yanzhi, & Qiu, Qinru. (2019). Fast and Accurate Trajectory Tracking for Unmanned Aerial Vehicles based on Deep Reinforcement Learning. In 2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (pp. 1-9).
9. Telli, Khaled, Kraa, Okba, Himeur, Yassine, Ouamane, Abdelmalik, Boumehraz, Mohamed, Atalla, Shadi, & Mansoor, Wathiq. (2023). A Comprehensive Review of Recent Research Trends on Unmanned Aerial Vehicles (UAVs). *Systems*, 11(8), 400.
10. Ahmed, Faisal, & Jenihhin, Maksim. (2022). A Survey on UAV Computing Platforms: A Hardware Reliability Perspective. *Sensors*, 22(16), 6286.