

Application and Principle of Quadcopters

Gary Tan ^{1,*}, Yuhan Jin ^{2, a}

¹ Wellington College International Shanghai, Shanghai, China

² BANZ Global Campus, Shanghai, China

* Corresponding author: Gary Tan (Email: Vivienne_sheng@163.com), ^a 2123588647@qq.com

Abstract. In an era of rapid growth in unmanned aerial vehicles (UAVs) and robotic systems, understanding the fundamental hardware and software architecture of quadcopters is crucial for both academic research and practical applications. This paper provided a comprehensive overview of the various components of a quadcopter, in the spirit of answering such demand in this era where the usage of quadcopters, and robotic systems in general, is growing at an exponential rate. This paper summarizes information from numerous sources to create a top-level excursion into the subject. This review systematically introduces core modules, and elucidates their roles in achieving stable flight control. Furthermore, this paper delves into algorithms such as Kalman filtering and PID control to illustrate how data fusion and feedback mechanisms improve accuracy, response speed, and stability. This paper not only summarizes the structural and functional design of quadcopters but also provides a theoretical foundation for further optimization of UAV autonomy, sensor fusion, and intelligent flight control.

Keywords: Quadcopters; Robotic Systems; UAV Systems.

1. Introduction

A drone, or formally Unmanned Aerial Vehicle (UAV), is an aircraft with no human onboard and is remotely controlled or autonomous [1]. Their history can be traced as far back as when the Montgolfier Brothers hoisted livestock aloft in unmanned balloons in the late 18th century. The first record of drones being used in war dated to 1,849, when the Austrian army launched a fleet of explosive-loaded unmanned balloons over the city of Venice as a punishment for its citizens' revolt [2].

However, the electronic system that constitutes a drone nowadays is dramatically more advanced and sophisticated than the humble balloons centuries ago. Combining with advancements in battery, camera, and artificial intelligence technologies, a generation of powerful and versatile drones are born, proving to be useful in various fields, including aerial photography, agriculture, emergency response, delivery, 3D mapping, military, and many more [3-6].

There are multiple flavors of drones, categorized by their rotor configuration:

Multi-Rotor Drones: These are hoverable drones with a shorter flight time, including the quadcopters that the average hobbyist and photographer would have, as it has a relatively friendly pricing.

Fixed-Wing Drones: This kind of drone looks like an airplane, requiring a runway or catapult for takeoff. It is used for mapping, surveying, long-range monitoring, and military purposes due to its energy efficiency (hence long flight range). The swarm tactics drones seen in the Russia-Ukraine conflict belongs to this category.

Single-Rotor Helicopter Drones: These drones resemble a typical helicopter, with a single large rotor, enabling it to transport a heavy payload. Thus, it finds usage in delivery and military contexts.

Fixed-Wing Hybrid VTOL Drones: Combining the merits of multi-rotor and fixed-wing drones, these are capable of both vertical takeoff & landing and efficient forward flight, making them useful for surveying, mapping, and search and rescue [7].

2. System Design

The key components of a quadcopter include the frame, battery, flight controllers, sensors, and motors and propellers. Utilizing the principle of conservation of energy in fluid flow (Bernoulli's principle), the spinning propellers produce a lifting force that allows the drone to stay in air. The quadcopter is controlled by the flight control module, which consists of the Microcontroller Unit (MCU), sensors, power management, communication (including USART and I²C), and memory [8]. The following section is about various aspects of the flight control module that enable it to function.

2.1 STM32F103C8T6 (the MCU)

The MCU is the “brain” of the drone, performing data processing and controlling its other components. The STM32F103C8T6 is from the STM32F1 family produced by STMicroelectronics. It is based on the ARM Cortex-M3 core, which is a 32-bit processor running at up to 72 MHz. This gives it good performance while keeping power use low. It also has many input/output (I/O) options and built-in peripherals, which made it a good choice for the project [9].

The STM32F103C8T6 is packaged with 48 pins, with each pin able to work as a general I/O or for special functions. Internally, it has a 64 KB Flash memory, a 20 KB SRAM, a clock system (with both an internal RC oscillator and support for external crystals), and buses for connecting components (AHB for high-speed access and APB for peripherals). Its peripherals include GPIO (for digital input/output), as well as several communication interfaces like USART, I²C, SPI, and USB [10].

2.2 Pulse Width Modulation (PWM)

The lift of the quadcopter depends on the rotation of the propellers [11]. To control the position of the quadcopter, namely the rate of rotation of the four propellers, a technology named Pulse Width Modulation (PWM) is utilized. It controls how much power is delivered to a device by turning the input signal to that device on and off very quickly. Instead of trying to make adjustments to the magnitude of the voltage of the signal, PWM alternates between fully ON (high voltage) and fully OFF (low voltage). By changing the proportion of the amount of time the signal stays ON against the time it stays OFF, the average power is adjusted [12].

The Electronic Speed Controller (ESC) creates the PWM signal for motor control. It consists of two components: the Time-Base Unit and the Output Compare Unit [13].

The Time-Base Unit consists of three parts: the Prescaler (PSC), Counter (CNT), and Auto-Reload Register (ARR). The prescaler divides the system clock frequency (72MHz in the project), lowering it to f_1 :

$$f_1 = \frac{f_{system\ clock}}{PSC + 1} \Big|_{f_{system\ clock}=72MHz}$$

For example, if $PSC = 71$, $f_1 = 1MHz$.

Note that for all equations in this section, a component's name denotes its value.

This prepares the signal for further processing. The prescaler is necessary because an unreasonably huge ARR value will be needed otherwise. The new signal with frequency f_1 goes into the CNT, which counts up from 0 until it reaches the ARR value (i.e. the upper limit). Then, CNT resets to 0 and starts counting again [14]. This creates a cycle, effectively dividing the signal again to produce a new signal with $ARR+1$ times the original period:

$$f_{final} = \frac{f_1}{ARR + 1}$$

For example, if $f_1 = 1MHz$, and the target f_{final} is 1kHz, then the ARR value should be adjusted to 999.

The Output Compare Unit is necessary to produce a PWM signal with the desired duty cycle, which is the proportion of the pulse width (length of ON signal) to the length of one period:

$$T_{period} = T_{ON} + T_{OFF}$$
$$duty\ cycle = \frac{T_{ON}}{T_{period}} = \frac{CCR}{ARR + 1}$$

It contains the Capture/Compare Register (CCR), which holds a value between 0 and ARR [15]. For the normal configuration, when CNT is less than CCR, the output signal is high and otherwise is low. This switching creates the PWN signal, enabling the adjustment of the duty cycle via the CCR value, and hence the output power [16]. The CCR value can be determined by:

$$CCR = (ARR + 1) \times \text{duty cycle}$$

For example, if ARR = 999 and the desired duty cycle is 30%, CCR = 300. If there comes the need to double the output power, this can be achieved by doubling the duty cycle to 60% by changing CCR to 600.

Figure 1 shows the output signal being produced as CNT changes using values from the previous example.

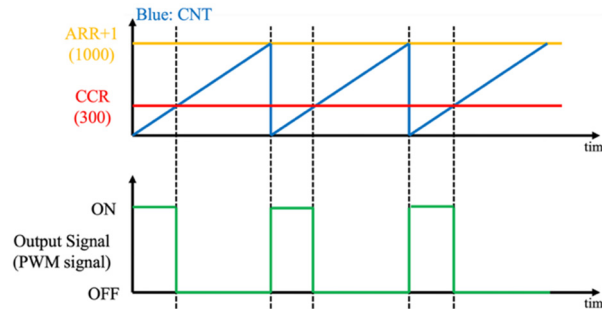


Figure 1. Output signal generated by the change of CNT

There are 3 main advantages to the usage of PWM:

Efficiency: PWM adjusts the power output via the duty cycle. Relatively less energy is wasted in this process in comparison to adjusting the DC voltage with variable resistors, where a considerable amount of the input is dissipated as thermal energy [17].

Stability: High-frequency switching makes the output smooth and stable.

Responsiveness: Changing duty cycle adjusts motor speed faster than changing DC voltage directly [18].

2.3 Universal Synchronous/Asynchronous Receiver/Transmitter (USART)

USART is a transceiver in STM32 for external communication [21]. It is an asynchronous, serial communication interface widely found in embedded systems, industrial equipment, and for connecting microcontrollers to computers for tasks like debugging or data logging [22].

USART supports full-duplex data transmission, consequently it requires two data transmission pathways, as shown on Figure 2. For each device, the TX pin is for data transmission, and RX is for data reception [23]. The ground wire (GND) is connected between the two devices to let them share a common reference voltage. The Voltage at Common Collector (VCC) is the positive power supply that must be supplied to the devices in order for them to function. This structure dictates that USART is a one-to-one transmission method [24].

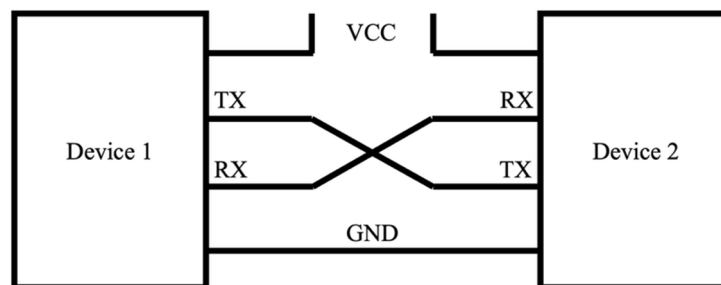


Figure 2. USART full-duplex data transmission path

As its name suggests, USART supports both synchronous and asynchronous transmission modes, though the latter has a significantly wider application in practical situations. To support synchronous

mode, an additional clock line is required, with one device acting as the master to provide clock signal to the other device (slave). In the case of communication between the MCU and a peripheral, the master device will be the former [25]. However, this mode of transmission was never used in the project.

For asynchronous transmission, the devices will agree upon a baud rate, which is the number of symbol units transferred per second. In the case of USART, one symbol is one bit, so the baud rate can be seen as the number of bits transferred per second [26].

In USART, data is transmitted as frames. A single frame consists of, in order, a start bit, data bits (5-9 bits, usually 8 for easier processing), an optional parity bit, and a stop bit (the stop bit can be 1-2 bits) [27]. The start bit is a low signal, signifying the start of the frame. The end bit is a high, bringing the signal back up so that the start bit of the next frame can turn it low again. In the data bits, there are highs and lows representing 1s and 0s. The least significant bit (LSB) is transmitted first, and the most significant bit (MSB) is transmitted last. Hence, the order that the data bits are sent is actually in reverse with the order which the byte is commonly written [28].

The optional parity bit enables the performance of a parity check. This is to detect any distortion that might occur to the data during the transmission process. The parity check can either be odd or even, as agreed beforehand by the communicating devices. The transmitting device will add the parity bit that ensures the total number of 1s in the data (parity bit and data bits) is odd or even, depending on the type of parity used. The receiving device will check the parity upon receiving, and if the parity is mismatched it knows that an error has occurred [29].

In the quadcopter built, USART communication has two main usages.

Serial debug output: it enables communication with the host computer during the development phase. This is useful for logging information, such as variable values and error messages, on the screen, making it more convenient for the programmer to develop and debug. Note that a USB to USART converter is needed due to the computer's inability to parse USART signals directly [30].

Wireless communication module connection: it is used to establish connection with ad-on wireless communication modules, such as the HC-05 Bluetooth module used on the quadcopter in the project. This enables wireless communication with the controller and/or computer.

2.4 Inter-Integrated Circuit (I²C) and MPU6050 (Inertial Measurement Unit (IMU))

Inter-Integrated Circuit (I²C, also written as IIC and I2C), is a small-distance serial communication tool invented and enhanced by Philips at around the 1800s, designed to allow low-speed data exchange between integrated circuits (ICs) within electronic systems to be possible. Its broad adoption stems are simple, only using two bidirectional wires: SDA (Serial Data Line) for transmitting data and SCL (Serial Clock Line) for synchronizing timing, making it perfect for codes with a limited amount of pin availability, such as embedded systems, microcontrollers, and compact devices. Running on a master-slave structure allows one or more master devices to start communication, while multiple slave devices respond to commands through a unique 7-bit or 10-bit addresses, enabling the coexistence of multiple peripherals on a single bus. Critical hardware components include the master and slave ICs, along with pull-up resistors (typically 2.2k Ω to 10k Ω) on SDA and SCL lines, which maintain high logic levels when no device is driving the lines, preventing signal floating and ensuring reliability; these resistors are crucial for stable communication, as IIC signals rely on open-drain or open-collector outputs that can only pull lines low, requiring the resistors to pull them high. This combination of minimal wiring, flexible addressing, and simple hardware has made IIC a staple in embedded systems, consumer electronics, and industrial control, where it efficiently connects microcontrollers to peripherals with low power consumption and straightforward implementation.

The MPU6050 is the inertial measure unit (IMU) used in the quadcopter project. It is a highly combined 6-axis device that tracks motion, broadly used in embedded systems, robotics, and consumer electronics. Developed by InvenSense, it integrates a 3-axis accelerometer and a 3-axis

gyroscope in a single compact package, making it a go-to choice for applications requiring precise motion sensing.

At its core, the accelerometer measures linear acceleration along the X, Y, and Z axes, detecting forces like gravity and movement, while the gyroscope tracks angular velocity around the same axes, capturing rotational motion such as tilting, spinning, or turning. This dual functionality allows the MPU6050 to provide comprehensive data about an object's orientation, velocity, and acceleration in 3D space.

A key feature of the MPU6050 is its built-in Digital Motion Processor (DMP), which offloads complex motion processing tasks—like fusing accelerometer and gyroscope data to compute orientation angles (Euler angles or quaternions)—from the host microcontroller. This reduces the computational burden on the main system, enabling real-time motion tracking with minimal latency.

The device communicates via the I²C protocol, using two wires (SDA for data, SCL for clock) to interface with microcontrollers such as Arduino, Raspberry Pi, or STM32. It operates on a low voltage (typically 3.3V) and consumes minimal power, making it suitable for battery-powered devices. Additionally, it includes a user-programmable interrupt pin, allowing it to trigger actions in the host system based on predefined motion events, such as detecting a sudden movement or a specific orientation.

Common applications include drones (for stability control), gaming controllers (for motion input), smartphones (for screen rotation), and robotics (for balance and navigation), highlighting its versatility in bridging the physical and digital worlds through precise motion sensing.

2.5 Kalman Filter

A Kalman Filter is an algorithm that takes a series of measurements collected over time, with statistical noise and other inaccuracies, and incorporates them into a prediction model, producing estimates that incline to be more precise comparing to the raw measurements alone. It is iterative and recursive, continuously producing estimates of the state of a dynamic system from a series of noisy measurements [19].

In the quadcopter project, it is used to filter IMU readings to produce more reliable results, and hence a more stable flight. It also blends the IMU and GPS readings to obtain more precise information about the position of the drone. The following section explains the principle of the Kalman algorithm by considering the example of getting the acceleration from one of the three accelerometers in the IMU (each for a distinct axis).

The accelerometer provides a measurement subject to noise, which can be modeled as a Gaussian distribution. This measurement is denoted as z_n :

$$z_n \sim N(\mu_n, r_n)$$

where μ_n represents the actual accelerometer reading, and r_n denotes the variance, as specified by the manufacturer for the accelerometer. The subscript n denotes a specific point in time, n , that the value is taken. Note that r_n is used due to convention, though it is a constant in this particular case.

There is also a prediction model that estimates the value of the acceleration at time n , $\hat{x}_{n,n-1}$:

$$\hat{x}_{n,n-1} = \hat{x}_{n-1,n-1} + \Delta a$$

where $\hat{x}_{\alpha,\beta}$ denotes the prediction of the acceleration \hat{x} at time α made at time β . Note that $\hat{x}_{n,n-1}$ and $\hat{x}_{n-1,n-1}$ are also modeled as Gaussian distributions, and $\hat{x}_{n,n-1}$ has variance $P_{n,n-1}$, which is a predetermined value by test-and-trial. More specifically, $\hat{x}_{n-1,n-1}$ is the final estimation of the Kalman algorithm at the last point in time, $n - 1$ [20]. Δa denotes the expected change in acceleration over the time interval from $n - 1$ to n , derived by a mathematical model which detail will not be discussed in this paper.

Neither of the two models, z_n and $\hat{x}_{n,n-1}$, are of sufficient precision for a reliable output. The Kalman algorithm combines them to produce a more precise result, $\hat{x}_{n,n}$:

$$\hat{x}_{n,n} = \omega_1 z_n + \omega_2 \hat{x}_{n,n-1}$$

The weights ω_1 and ω_2 sum to 1, hence:

$$\omega_2 = 1 - \omega_1$$

The goal of the algorithm is to optimize the variance of $\hat{x}_{n,n}$, and thus make the mean of $\hat{x}_{n,n}$ closer to the real value (this is the value that will be used for flight control calculations):

$$\begin{aligned} \text{Var}(\hat{x}_{n,n}) &= \text{Var}(\omega_1 z_n + \omega_2 \hat{x}_{n,n-1}) \\ &= \omega_1^2 r_n + (1 - \omega_1)^2 P_{n,n-1} = f(\omega_1) \end{aligned}$$

to find the value of ω_1 that yields the minimum of $f(\omega_1)$, it is necessary to calculate the root of the derivative of $f(\omega_1)$:

$$\begin{aligned} f'(\omega_1) &= 2\omega_1 r_n - 2(1 - \omega_1)P_{n,n-1} = 0 \\ \omega_1 &= \frac{P_{n,n-1}}{P_{n,n-1} + r_n} \end{aligned}$$

substituting this into the formula for $\hat{x}_{n,n}$ gives:

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + \frac{P_{n,n-1}}{P_{n,n-1} + r_n} (z_n - \hat{x}_{n,n-1})$$

which is the formula to calculate the estimate of $\hat{x}_{n,n}$ at time n . This estimation is performed continuously at arbitrary intervals (which is usually consistent with the sensor output frequency) in a recursive manner, continuously producing a more-precise reading that is critical for accurate flight control.

2.6 PID Control

PID control is the most commonly used feedback mechanism in industrial and engineering settings [31]. This is because of the simplicity, efficiency, and effectiveness nature of the algorithm in comparison to its alternatives. PID technology is capable of adjusting the state of the device in real time via actuators such as motors. The basic principle is that it continuously monitors the error between the desired setpoint and the measured value and trying to eliminate the error in an optimized manner through proportional, integral, and derivative (PID) calculations [32].

Devices uses closed loop systems, such as PID control, to automatically rectify its status rather than open loop systems, which requires manual handling. In the case of the quadcopter, the error to be rectified during flight is air turbulence. The aim of PID control (and any other closed loop systems) is the minimize the error between the actual status and the desired status [33]. Figure 3 shows how a closed loop system is used in the quadcopter.

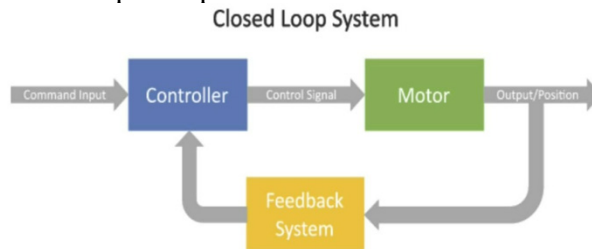


Figure 3. Closed-loop system used in quadcopter

The PID controller is favored over the simplistic On/Off control as it is capable of producing a smoother and more precise control, while the latter results in fluctuations that is not tolerable for the purpose of the project, as illustrated in Figure 4 [34].

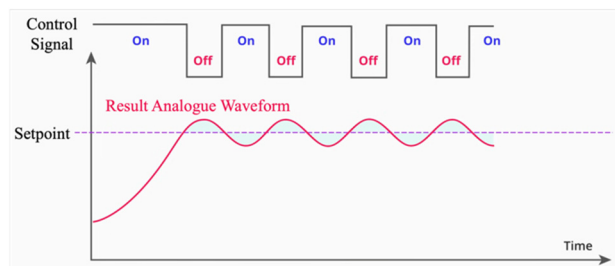


Figure 4. The issue of simplistic On/Off control

The letters in PID stands for proportional, integral, and derivative, which are the core components in the PID algorithm. These components are combined to produce the control output, $u(t)$, which is then used to control the actuators to adjust the status:

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

where $e(t)$ is the error function, that is, the function that outputs the error between the reference (desired status), $r(t)$, and the output (actual status), $y(t)$, with respect to time t :

$$e(t) = r(t) - y(t)$$

K_p , K_i , and K_d are the proportional gain, derivative gain, and integral gain respectively, in other words, constant coefficients set manually during configuration. These determine the significant of the three components, P, I, and D:

The proportional component (P) is simply the error function, $e(t)$. This component responds rapidly to error, enabling a quick response even for a small error [35]. However, if used excessively, this component will also increase the overshoot, causing the drone to be unstable and sway in the air in an overresponse to the error. This is because the drone's inertia is not considered in the calculation.

The derivative component (D) is $de(t)/dt$, the derivative of $e(t)$ with respect to t . This component adjusts the control output in accordance with the rate of change of error, which opposes the proportional component, reducing the instability it causes [36].

The integral component (I) is $\int_0^t e(\tau) d\tau$, which is not significant for the quadcopter project. This is more useful when there is a continuous effect, as the integral will build up as the effect continues in a direction, such as a large constant wind. However, it is not particularly useful for the random gusts under normal circumstances encountered during flight [37].

3. Discussion

To evaluate the performance of the designed quadcopter system and validate the effectiveness of the implemented algorithms, several experiments were conducted focusing on signal processing, communication, and motion sensing. The analysis primarily investigates how Kalman filtering, PID control, and communication modules contribute to the overall system stability and precision.

Figure 5 shows the output of ACC_X without Kalman filtering, and Figure 6 shows it with Kalman filtering (not recorded simultaneously). It is noticeable that the signal with filtering is much smoother than such without filtering.

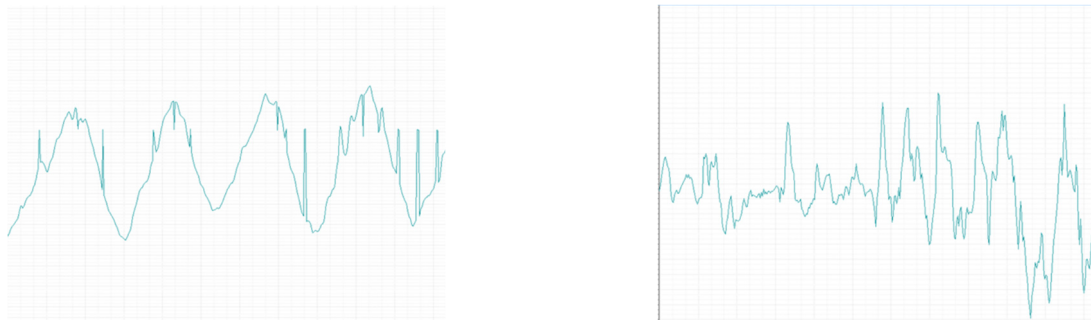


Figure 5. The output of ACC_X without Kalman filtering; **Figure 6.** The output of ACC_X with Kalman filtering

Figure 7 shows the transmission of the string “Hello World!\r\n” via USART, and the reception of it using an open-source flight control terminal. Figure 8 shows output from MPU6050 about the rotation along the x axis, also transmitted to the computer via USART.

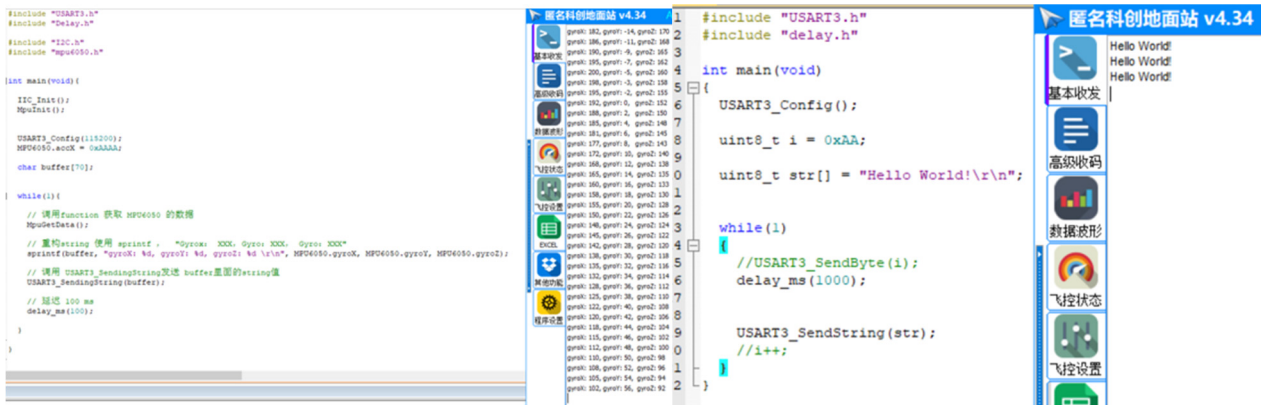


Figure 7. Figure 8 Related code diagram

Overall, the experimental results validated previous expectations.

4. Conclusion

This paper provides a comprehensive and top-level overview of the various components in the quadcopter. This review has presented a detailed and systematic examination of the fundamental components and control principles underlying a quadcopter system. By integrating electronic modules, the study illustrates how modern quadcopters achieve precise maneuverability and stability. Furthermore, the implementation of Kalman filtering and PID control demonstrates the critical role of signal processing and feedback regulation in ensuring reliable performance under dynamic flight conditions.

Future work may focus on expanding the system's autonomy through adaptive control, real-time path planning, and the integration of AI-driven flight strategies. Additionally, enhancing communication reliability, developing energy-efficient architectures, and introducing swarm coordination could further broaden the applications of quadcopters in areas such as environmental monitoring, precision agriculture, and search-and-rescue operations. More research can be conducted with further depth to explore these fields in greater technical detail, forming an overview that is suitable for professionals in the industry. Overall, the research lays a solid foundation for understanding and advancing the interdisciplinary design of modern UAV platforms.

References

- [1] Wikipedia contributors. (2025, July 22). Unmanned aerial vehicle - Wikipedia. https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle.
- [2] The complete history of drones for police and law enforcement. (n.d.). Axon. Retrieved July 23, 2025, from <https://www.axon.com/resources/history-of-drones>.
- [3] Oxbow. (2024, December 26). Top 5 applications of drone technology across modern industries. Oxbow Services. <https://oxbowintellect.com/blog/top-5-applications-of-drone-technology-across-modern-industries/> #: ~: text= In% 20 recent% 20 years% 2C% 20drone% 20technology, that% 20are% 20revolutionizing % 20modern% 20industries.
- [4] Kenney, M. (2024, October 21). Drone technology changing modern service industries. MacroFab, Inc. <https://www.macrofab.com/blog/how-drone-technology-is-changing-modern-service-industries/#:~: text=From % 20showcasing% 20real% 20estate% 20properties,bring% 20is% 20evident% 20across% 20industries>.
- [5] Gillis, A. S., Lutkevich, B., & Earls, A. R. (2024, July 26). What is a drone (UAV)? Search IoT. <https://www.techtarget.com/iotagenda/definition/drone#:~: text=Drones% 20can% 20be% 20equipped% 20with,drone% 20is% 20to% 20the% 20ground>.

- [6] AZoRobotics. (2024, October 9). Aerial Robotics: Exploring the technology behind unmanned flying systems. <https://www.azorobotics.com/Article.aspx? ArticleID=719#:~: text=In% 20agriculture% 2C% 20drones%20monitor%20crop%20health%2C%20soil,to%20improve%20yields%20and%20reduce%20 resource%20consumption>.
- [7] Dukowitz, Z. (2025, April 22). Drone Types and Uses: A Comprehensive Guide for drone Pilots [New for 2025]. Drone Pilot Ground School. <https://www.dronepilotgroundschool.com/drone-types/#:~: text= Fixed%20Wing%20Drone-,Single%2DRotor%20Helicopter%20Drones,II%20Surveying%20and%20M apping%20Drone>.
- [8] Pachpute, S. (2024). Working principle and components of drone. CFD Flow Engineering. <https://cfdflowengineering.com/working-principle-and-components-of-drone/#:~: text=It% 20ca n% 20be %20guided%20autonomously,using%20onboard%20computers%20and%20robots.&text=The%20 basic%20components%20of%20a,%2C%20flight%20controller%2C%20and%20sensors>.
- [9] STM32F103C8 - STMicroElectronics. (n.d.). STMicroelectronics. <https://www.st.com/en/ microcontrollers- microprocessors/stm32f103c8.html>.
- [10] STMicroelectronics. (2023). STM32F103x8, STM32F103xB datasheet: Arm®-based 32-bit MCU. Retrieved July 29, 2025, from <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>.
- [11] Experimental and numerical investigation of rotor aerodynamics for a quadrotor UAV in hover. (2017). *Journal of the American Helicopter Society, 62*(1), 1–9. <https://doi.org/10.4050/JAHS.62.012005>.
- [12] GeeksforGeeks. (2025, July 23). Pulse Width Modulation (PWM). GeeksforGeeks. <https://www. geeksforgeeks. org/electronics-engineering/pulse-width-modulation-pwm/>.
- [13] Holmes, D. G., & Lipo, T. A. (1992). Pulsewidth modulation—A survey. *IEEE Transactions on Industrial Electronics*, 39(5), 410–420. <https://ieeexplore.ieee.org/document/192110>.
- [14] Holmes, D. G., & Lipo, T. A. (2003). Pulse width modulation for power converters: Principles and practice. IEEE Press. <https://books.google.com/books?id=JQ1BgAAQBAJ&pg=PA97>.
- [15] Völp, M., et al. (2018). A configurable output compare unit for real-time embedded systems. *IEEE Transactions on Industrial Electronics*, 65(4), 1–12. <https://doi.org/10.1109/TIE.2017.2756584>.
- [16] van der Broeck, H. W. (1988). Space vector PWM techniques for three-phase inverters. In *Proceedings of the IEEE Industry Applications Society Annual Meeting* (pp. 1–8). IEEE. <https://doi.org/10.1109/ IAS.1988.25194>.
- [17] Zhang, Y., et al. (2021). Neural network-based adaptive PWM control for motor drives. *IEEE Transactions on Industrial Informatics*, 17(3), 1–10. <https://doi.org/10.1109/TII.2020.3015432>.
- [18] Reusch, D., et al. (2015). Efficiency optimization of 10 MHz PWM GaN converters. In *Proceedings of the IEEE Applied Power Electronics Conference (APEC)* (pp. 1–7). IEEE. <https://doi.org/10.1109/ APEC. 2015. 7104401>.
- [19] DeepAI. (2020, June 25). Kalman filter. DeepAI. <https://deepai.org/machine-learning-glossary-and- terms/kalman-filter#:~: text=A%20Kalman% 20Filter%20is%20a,on% 20a%20single% 20 measurement% 20alone>.
- [20] Alex Becker (www.kalmanfilter.net). (n.d.). Online kalman filter tutorial. <https://www. kalmanfilter. net/ alphabeta. Html>.
- [21] Lee, M., & Zhang, S. (2020). Efficient data communication using USART in IoT devices. *IEEE Transactions on Industrial Informatics*. <https://doi.org/10.1109/TII.2020.3023456>.
- [22] Nguyen, T., & Wang, L. (2020). Optimizing USART communication for low-power microcontrollers. *ACM Symposium on Embedded Systems for Real-Time Multimedia*. <https://doi.org/10.1145/ 3423211. 3425678>.
- [23] Kumar, S., et al. (2019). Real-time bidirectional communication using full-duplex USART in embedded systems. *ACM International Conference on Embedded Systems (EMSOFT)*. <https://doi.org/10.1145/ 3358228.3358241>.
- [24] Garcia, E., et al. (2017). Comparative analysis of half-duplex vs. full-duplex USART in wireless sensor networks. *IEEE International Symposium on Circuits and Systems (ISCAS)*. <https://doi.org/10. 1109/ ISCAS. 2017.8050867>.

- [25] Sharma, A., & Patel, R. (2020). Comparative analysis of UART and USART communication protocols in embedded systems. *International Journal of Engineering Research & Technology (IJERT)*. <https://doi.org/10.17577/IJERTV9IS050452>.
- [26] Johnson, M., & Smith, P. (2019). Analysis of baud rate tolerance in asynchronous serial communication. *IEEE Transactions on Instrumentation and Measurement*. <https://doi.org/10.1109/TIM.2019.2944568>.
- [27] Rodriguez, M., et al. (2021). Error detection and correction in USART communication for industrial applications. *IEEE International Symposium on Industrial Electronics (ISIE)*. <https://doi.org/10.1109/ISIE45552.2021.9576231>.
- [28] Müller, J., et al. (2020). Inverted USART signal handling in low-voltage embedded systems. *IEEE Transactions on Circuits and Systems II*. <https://doi.org/10.1109/TCSII.2020.3042996>.
- [29] Watson, D., & Williams, H. (2019). *Cambridge International AS & A level Computer Science*. Hodder Education.
- [30] Deshpande, S., et al. (2022). Baud rate adaptive USART debugging for heterogeneous embedded networks. *Journal of Hardware and Systems Security*. <https://doi.org/10.1007/s41635-022-00128-6>.
- [31] García Carrillo, L. R., López-Estrada, F. R., Lozano, R., & Dzul, A. E. (2020). PID control of quadrotor UAVs: A systematic review. *Drones*, 4(3), 37. <https://doi.org/10.3390/drones4030037>.
- [32] Veronesi, M., & Visioli, A. (2013). Optimal tuning of PID controllers using iterative learning approach. *Control Engineering Practice*, 21(1), 74–84. <https://doi.org/10.1016/j.conengprac.2012.10.007>.
- [33] Franklin, G. F., Powell, J. D., & Emami-Naeini, A. (2019). *Feedback control of dynamic systems* (8th ed.). Pearson Education. <https://doi.org/10.1002/9781119058218>.
- [34] Visioli, A., & Zhong, Q.-C. (2011). Comparative study of PID and on/off control in temperature regulation systems. *IEEE Transactions on Control Systems Technology*, 19(5), 1276–1281. <https://doi.org/10.1109/TCST.2010.2043732>.
- [35] Skogestad, S. (2003). Tuning rules for optimal proportional gain selection in PID controllers. *Journal of Process Control*, 13(4), 291–309. [https://doi.org/10.1016/S0959-1524\(02\)00062-8](https://doi.org/10.1016/S0959-1524(02)00062-8).
- [36] Åström, K. J., & Hägglund, T. (2002). The effect of derivative action in PID controllers. *Automatica*, 38(5), 703–709. [https://doi.org/10.1016/S0005-1098\(02\)00021-4](https://doi.org/10.1016/S0005-1098(02)00021-4).
- [37] Visioli, A. (2001). Optimal tuning of integral gain for industrial PID controllers. *Control Engineering Practice*, 9(10), 1195–1201. [https://doi.org/10.1016/S0967-0661\(01\)00040-X](https://doi.org/10.1016/S0967-0661(01)00040-X).