**a)** Swarm drones, also known as a fleet of Unmanned Aerial Vehicles (UAVs), consist of multiple aerial robots working in concert to achieve specific goals. Each drone in a swarm is typically propelled by rotors, allowing for vertical takeoff and landing (VTOL). These drones can be controlled either manually through remote operations or autonomously via onboard processors. While their military applications are well-established, civilian uses are rapidly expanding due to the low cost and versatility of drone swarms.

**Technical Components and Characteristics**

* **Control and Autonomy:** Swarm drones can operate under different levels of autonomy. Manual control involves human operators using remote controls to direct drone activities, while autonomous operation relies on processors onboard the drones. These processors enable drones to perform tasks without human intervention, making decisions based on local information and interactions with neighboring drones. This decentralized approach enhances coordination and adaptability, allowing the swarm to function efficiently even if individual drones fail.
* **Classification of Swarms:** Drone swarms can be classified based on their level of autonomy and organizational structure. Fully autonomous swarms operate without human intervention, executing pre-programmed missions. Semi-autonomous swarms might require occasional human input. Structurally, swarms can be single-layered, where each drone acts independently, or multi-layered, where leadership is hierarchical. In multi-layered swarms, dedicated leader drones manage subsets of worker drones, with a ground-based server station at the top layer overseeing the entire swarm.
* **Communication and Coordination:** Effective communication is crucial for swarm drone operations. Ad hoc networks, such as Mobile Ad Hoc Networks (MANETs), allow drones to communicate dynamically without fixed infrastructure. These networks support real-time data exchange and coordination. Data fusion techniques aggregate information from multiple drones, creating comprehensive situational awareness. To ensure mission success, communication must be reliable, facilitated by motion-driven packet forwarding algorithms and network expansion techniques like the Takahashi Self Deployment (TSD) algorithm.
* **Energy Management:** Long-duration missions necessitate efficient energy management. Autonomous battery swapping and recharging systems are critical. For example, systems have been developed to perform hot-swapping of batteries, where external power maintains the drone’s operations while the battery is exchanged. These systems can be enhanced to handle multiple drones simultaneously, prioritizing those with urgent needs. Advanced ground recharge stations further optimize energy management, using better electrical contacts and balancing circuits to reduce charging times.
* **Robustness and Collision Avoidance:** Swarm drones must be robust against collisions to ensure mission continuity. Drones are often designed with lightweight, protective materials such as carbon fiber cages to withstand impacts. Collision avoidance algorithms, such as those based on Model Predictive Control (MPC), predict trajectories and execute evasive maneuvers. Additionally, ultrasound localization methods enable precise navigation in environments where GPS signals are unavailable, further enhancing collision avoidance capabilities.
* **Surveillance Systems:** Quadcopter-based surveillance systems leverage the maneuverability and payload capacity of drones for effective monitoring. These systems use advanced algorithms for target detection and tracking, often modeling the swarm as a multi-agent system to optimize cooperation and ensure safety. Path planning algorithms guide drones through dense environments, ensuring thorough coverage and avoiding obstacles. Techniques such as scalable dynamic grids allow for efficient area surveillance, adapting drone altitude and positioning as needed.
* **Swarm Design and Optimization:** Designing and managing drone swarms involves principles from Organic Computing (OC). This approach facilitates the self-optimization of individual drones and the collective optimization of the swarm’s efforts. Distributed localization frameworks enable drones to autonomously determine their positions relative to others, supporting fast propagation of this information throughout the swarm. These frameworks, often based on Internet of Things (IoT) platforms, improve the efficiency of 3D swarming applications and enhance interaction with human operators.
* **Hovering Performance:** Maintaining stability and synchronization during hovering is crucial for successful mission execution. Two-stage controllers generate control feedback to ensure the stability of individual drones and synchronization within the swarm. This approach is effective even in the case of individual drone failures, maintaining overall swarm performance.

**Dynamics and Flying Mechanisms**

An autonomous and adaptive quadcopter is an underactuated system featuring four input engines and propellers, allowing for control over roll, pitch, yaw, and thrust, while maintaining six degrees of freedom for movement and maneuvering. The propellers, also known as rotors, have fixed-pitch, mechanically moveable blades, meaning the rotor pitch does not vary during rotation. The quadcopter's movement direction, along with its roll, pitch, and yaw, depends on the differential throttle provided by the four motors operating the propellers. Stable flight necessitates high-precision motor control, with algorithms continuously taking feedback and adjusting the motor throttle to control these movements.

**Roll, Pitch, and Yaw Control**

* Yaw Movement: Controlled to ensure stability of rotation on the vertical or Z-axis.
* Pitch Movement: Managed to maintain stability on the Y-axis (lateral or transverse axis), determining the degree of side-to-side tilting.
* Roll Movement: Involves controlling how high the nose or front of the drone is lifted, ensuring stability on the X-axis (longitudinal axis).

**Rotor Configurations**

There are two primary configurations for quadcopter rotors: the ‘+’ and ‘x’ patterns. In both configurations, two rotors on opposite ends always rotate in the same direction, while the other two rotate in the opposite direction. The 'x' design is more stable and can produce more rotational acceleration than the ‘+’ organization. The ‘+’ configuration, however, is well-suited for sports flying due to its ease of maneuvering, while the ‘x’ configuration is preferred for aerial photography as it keeps the propellers out of the screen.

**Swarm Hierarchy and Control**

In a hierarchical swarm of quadcopters, a leader-subordinate flying model allows a single user to control the entire swarm through the leader drone using an intuitive remote control interface. This self-organizing structure behaves like a multi-agent control system, with communication between the operator and the swarm facilitated via a wireless system. Each drone can communicate with its peers at the same level and with its immediate leader drone(s). Leader drones at the highest hierarchy level communicate with the ground-based server, sharing data collected by the swarm and distributing mission objectives.

**Mechanical Design**

The initial step in constructing a balanced quadcopter involves building an adaptive computing platform capable of handling disturbances and uncertainties. A dynamic model of the drone is created, consisting of mathematical equations that deal with all acting forces at any given time interval. A basic battery-powered quadcopter includes a mainframe, Electronic Speed Control (ESC), Inertial Measurement Unit (IMU), programmable microcontrollers, electric motors, radio transmitters and receivers, batteries, sensors, and four rotors.

**Sensor Integration**

Key sensors essential for stable flight include ESCs and IMUs. ESCs, with brushless DC motors, consist of a three-phase inverter and rotor position feedback, converting Pulse Width Modulation (PWM) signals from radio receivers or flight controllers to run the motor. An IMU, comprising an accelerometer and a gyroscope, determines the angular position and altitude of the quadcopter. Additional modules like barometers, magnetometers, and altitude sensors provide real-time measurements, while ultrasonic sensors (or sonar/radar) detect and avoid obstacles. Infrared rangefinders and LiDAR are used for mapping and detecting flat objects. Cameras mounted on quadcopters provide visual feedback, image recognition, 3D obstacle identification, and avoidance, while GPS modules aid in path planning, tracking, and localization during outdoor operations.

**Control Approaches**

Various control techniques are employed to manage UAVs, including Proportional Integral Derivative (PID) control, adaptive control, Boltzmann-Hamel equations of motion, localization and mapping methods, marker recognition algorithms, vision-based schemes, wireless communication methods, memory-based controllers, brain emotional learning-based intelligent controllers, and learning-based control methods like fuzzy logic, artificial neural networks, iterative and reinforcement learning.

* PID Control: Studies have shown that while P and PI controllers may not provide sufficiently good responses, PD and PID controllers perform adequately in steady-state stabilization.
* Gravity Compensated PID: Effective for altitude control and attitude stabilization in both indoor and outdoor flight tests.
* Nonlinear Control: Techniques like backstepping and Sliding Mode Control (SMC) have been used for position and attitude control, with backstepping proving more efficient in high perturbation scenarios.
* Model Predictive Control (MPC): Employed for generating specified trajectories, ensuring robustness compared to geometric tracking algorithms.
* Kalman Filters: Used for fusing sensor data to improve altitude measurements and control the altitude of quadrotors.
* Robust Control Methods: Approaches like pole placement, Lyapunov functions, and the Linear Quadratic Regulator (LQR) have been implemented for stabilization and trajectory tracking.
* Adaptive Control Schemes: Such as model reference adaptive control, have been proposed to maintain performance before and after payload changes.

**Applications in Military**

In military contexts, swarm drones are used for reconnaissance and surveillance, covering large areas efficiently and providing real-time data to command centers. They can execute coordinated offensive operations, overwhelming enemy defenses through simultaneous multi-vector approaches. Additionally, swarm drones are valuable in search and rescue operations, particularly in hazardous environments, locating and assisting individuals in need.

**b)** Using an ESP8266 to communicate with a controller (such as an Arduino, Raspberry Pi, or any other microcontroller) involves setting up a communication interface between the two devices. Here’s a step-by-step guide on how to achieve this:

**Materials Needed**

1. ESP8266 module (e.g., ESP-01, NodeMCU)

2. Microcontroller (e.g., Arduino Uno, Raspberry Pi)

3. USB to Serial Adapter (if programming ESP8266 separately)

4. Breadboard and jumper wires

5. Power supply (ensure the correct voltage for both ESP8266 and the microcontroller)

**Step-by-Step Guide**

**1. Setting Up the ESP8266**

* Powering the ESP8266: Ensure you provide the correct voltage to the ESP8266. Typically, it operates at 3.3V.
* Connecting to a Computer: Use a USB to Serial adapter if you need to program the ESP8266 separately. Connect TX, RX, GND, and 3.3V from the adapter to the ESP8266.

**2. Flashing Firmware (if necessary)**

If your ESP8266 doesn’t have the desired firmware, you might need to flash it with firmware like NodeMCU (Lua) or use the Arduino IDE to upload a custom sketch.

Using Arduino IDE: Install the ESP8266 board package and write your sketches in C/C++.

**3. Connecting ESP8266 to the Controller**

* Power Connections: Ensure that the ESP8266 and the microcontroller share a common ground (GND).
* Serial Communication: Connect the RX and TX pins of the ESP8266 to the TX and RX pins of the microcontroller respectively (note that some setups may require you to cross the connections: RX to TX and TX to RX).
* Voltage Level Shifting: If the microcontroller operates at a higher voltage (e.g., 5V for Arduino Uno), use a level shifter or a voltage divider to protect the ESP8266.

This setup allows the drone (via ESP8266) to communicate with a remote server over Wi-Fi. The code provided just shows a sketch of how data can be sent and how to run any specific commands in a loop. But no specific data has been sent or received in the attached code.

**c)** The Bidirectional Multitier Cognitive Swarm Drone Network (BMCSDN) represents a cutting-edge solution to the challenges faced by 5G networks, focusing on enhancing spectrum efficiency and optimizing downlink transmission parameters. Central to this network are cognitive swarm drones (CSDs), which operate on a master-slave principle. In this setup, the master drone guides the path and resource allocation for slave drones, ensuring that the network's performance is maximized through coordinated and cooperative efforts.

One of the key innovations of the BMCSDN is its approach to resource allocation. Using wireless cyclic prefix orthogonal frequency division multiple access (CP-OFDMA), the drones can dynamically allocate resources in response to real-time network conditions. This dynamic allocation not only optimizes the usage of the available spectrum but also reduces the likelihood of interference, which is a critical factor in maintaining high-quality communication links.

Dynamic spectrum sharing is another critical feature of the BMCSDN. The network employs long short-term memory (LSTM) protocols to facilitate adaptive communication strategies. These protocols enable the drones to learn from and adapt to current network conditions, ensuring that resource sharing is both efficient and responsive to the ever-changing demands of 5G applications.

Interference management is adeptly handled through the use of orthogonal regenerative decode and forward (ORDF) and orthogonal amplify and forward (OAF) strategies. These methods allow the network to maintain reliable communication links by effectively managing and mitigating interference between the drones. This ensures that the communication remains robust even in densely populated spectrum environments.

Lastly, the BMCSDN optimizes transmission efficiency by dynamically adjusting both transmission power and drone altitudes. This adaptability enhances the network's coverage and capacity, allowing it to meet the diverse and demanding requirements of various 5G applications. By integrating these advanced drone technologies and cognitive resource management techniques, the BMCSDN framework significantly enhances the reliability, efficiency, and flexibility of 5G networks.

**d)** **Hardware Setup**

To set up our UAV drone system, we begin by mounting the Pixhawk Cube flight controller onto each drone. This controller is chosen for its reliability and support for various peripherals. Each drone is equipped with essential components such as GPS modules for navigation, telemetry radios for communication with our ground control station (GCS), and additional sensors like LIDAR for obstacle detection and color sensors for target identification.

### **Software Installation**

We install necessary software components on both the drones and our ground control station (GCS). We start by installing MAVLink, a lightweight messaging protocol for communicating with unmanned systems. Next, we install DroneKit, a powerful Python library that allows us to communicate with Pixhawk-based drones using MAVLink. These libraries enable us to send commands to the drones and receive telemetry data back from them during operation.

### **Configuring Pixhawk Cube**

Configuration of the Pixhawk Cube is crucial to ensure proper functionality and communication between our drones and the GCS. Using tools such as Mission Planner or QGroundControl, we configure each Pixhawk Cube unit. We set unique system IDs for each drone to distinguish them when communicating over the network. Additionally, we calibrate sensors like GPS and compass to ensure accurate navigation and orientation data during flight.

### **Network Setup**

Establishing a reliable network setup is essential for seamless communication between our GCS and the drones. We configure telemetry radios on each drone to establish communication links with the GCS. Each telemetry radio operates on a unique frequency to prevent interference between drones. We test the communication range and reliability of the telemetry radios to ensure they can maintain a stable connection throughout our flight mission.

### **Pre-Flight Checks**

Before conducting any flight operations, we perform thorough pre-flight checks to verify the functionality and readiness of each drone. We check all physical connections between the flight controller, sensors, and peripherals to ensure they are securely attached and functioning correctly. We perform firmware updates on the Pixhawk Cube to ensure it has the latest features and bug fixes. Additionally, we conduct a pre-flight calibration of sensors and verify GPS lock to ensure accurate positioning data during flight.

### **Flight Test Preparation**

We prepare for the flight test by planning a mission route using our GCS. We define waypoints and set parameters for each drone to follow during autonomous flight. Our mission plan includes objectives such as obstacle detection and target identification using onboard sensors like LIDAR and color sensors. We ensure that our remote or GCS is configured to initiate and monitor the flight test, including logging telemetry data and receiving real-time updates from the drones.

### **Initiate Test Flight**

Once all preparations are complete, we use our GCS to initiate the test flight. We arm and take off each drone manually or using autonomous flight modes supported by the Pixhawk Cube and DroneKit. We monitor the drones closely during flight to ensure they follow the planned mission route and perform tasks such as obstacle avoidance and target detection as programmed. We use telemetry data and onboard sensors to gather information about the drones' performance and adjust flight parameters as needed.

### **Code Integration**

We integrate sensor data processing and swarm communication logic into our control software. Using the capabilities provided by DroneKit, we handle communication with each drone and implement swarm technology for collaborative tasks such as sharing target detection information among drones. We develop custom scripts or use existing libraries to optimize the efficiency and coordination of our drone swarm during flight operations.

**References:**

1. <https://www.sciencedirect.com/science/article/pii/S2452414X18300086>
2. <https://www.instructables.com/ESP8266-Communication-With-Server-and-ESP8266/>
3. <https://ieeexplore.ieee.org/document/9162676>