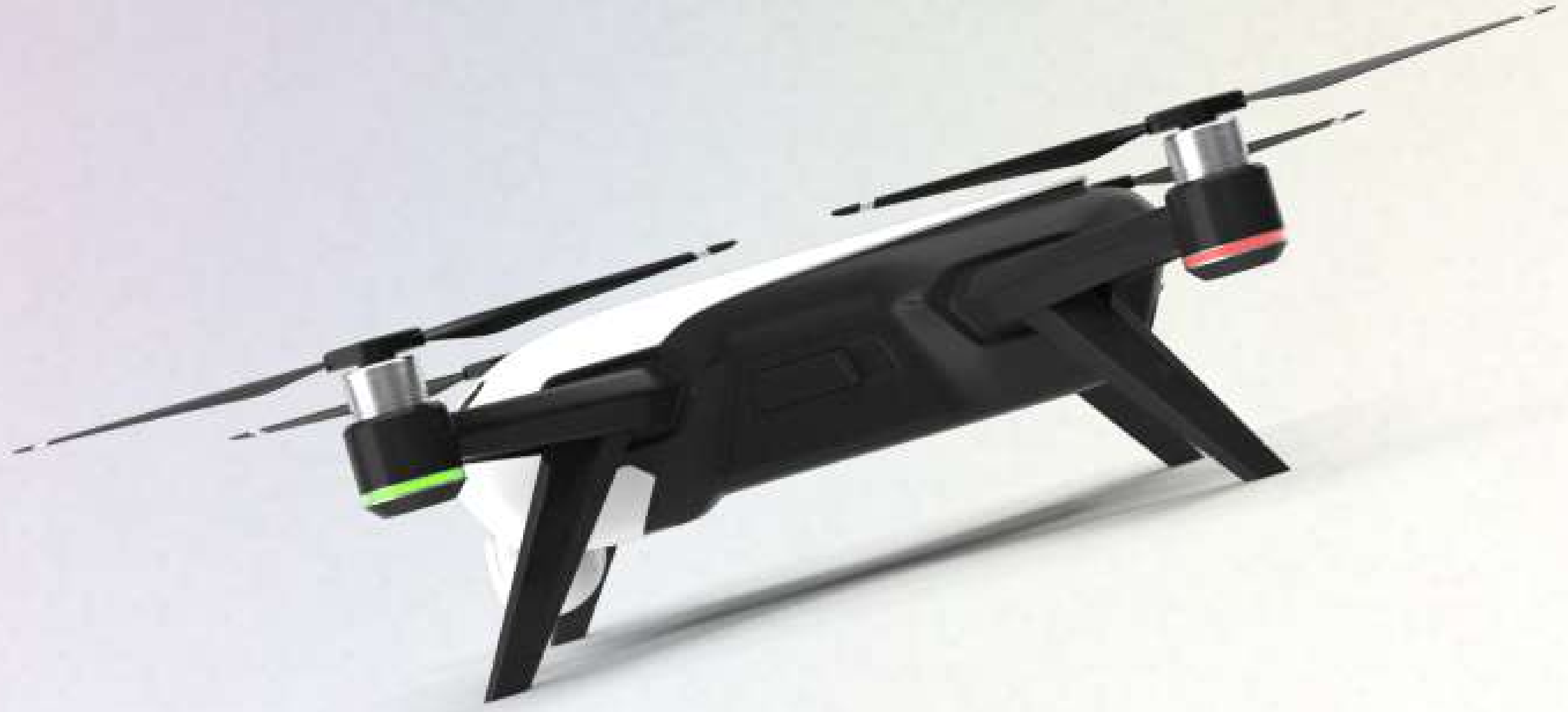
Team 45

DRONA AVIATION

PLUTO DRONE SWARM CHALLENGE



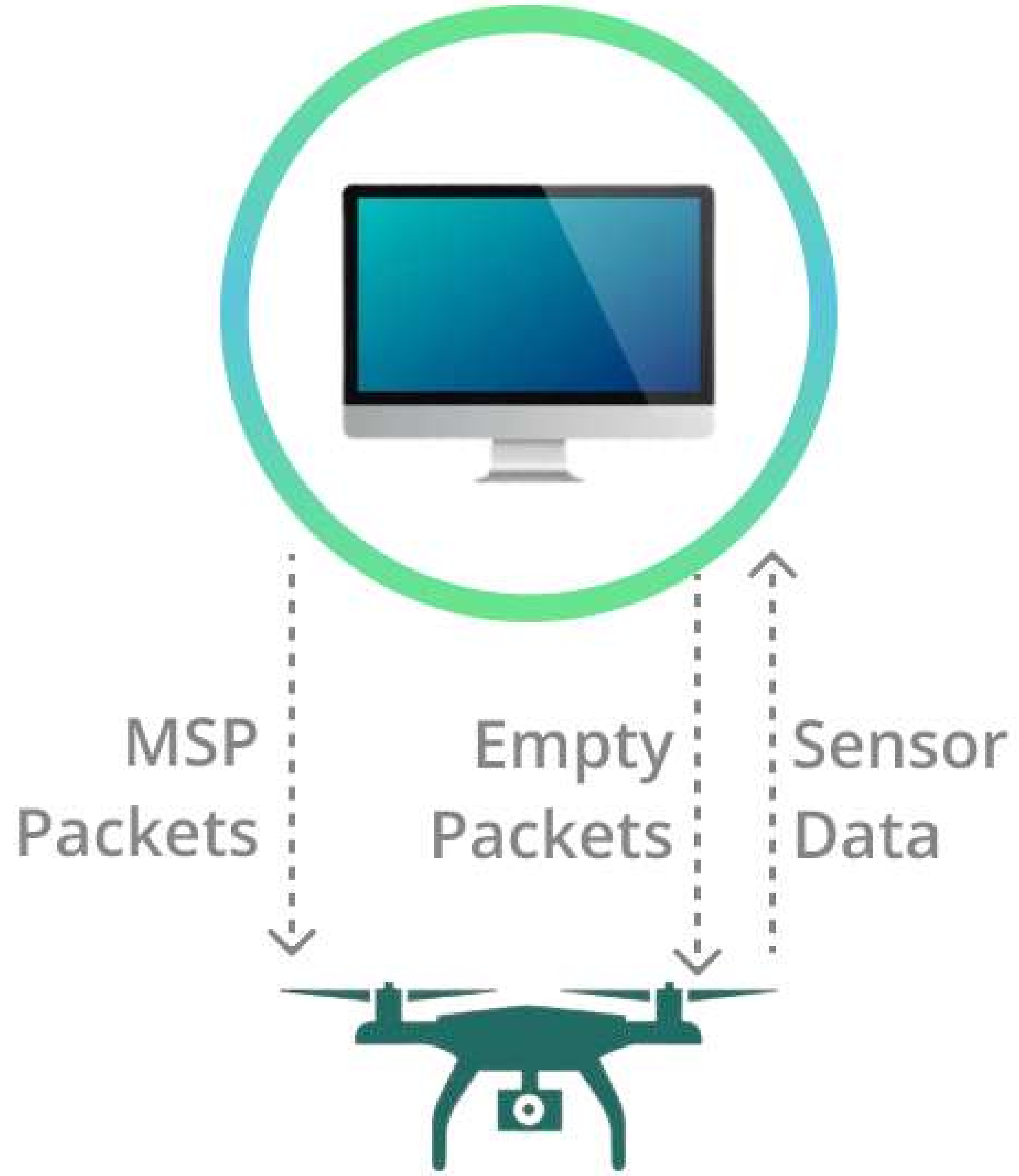
TASK 1

- Python wrapper based on telnet protocol
- Provides real-time 2 way communication based on TCP connection
- **Multiprocessing** and **Threading** were used to reduce temporal complexity
- Separate threads were utilised for ArUco and Drone control

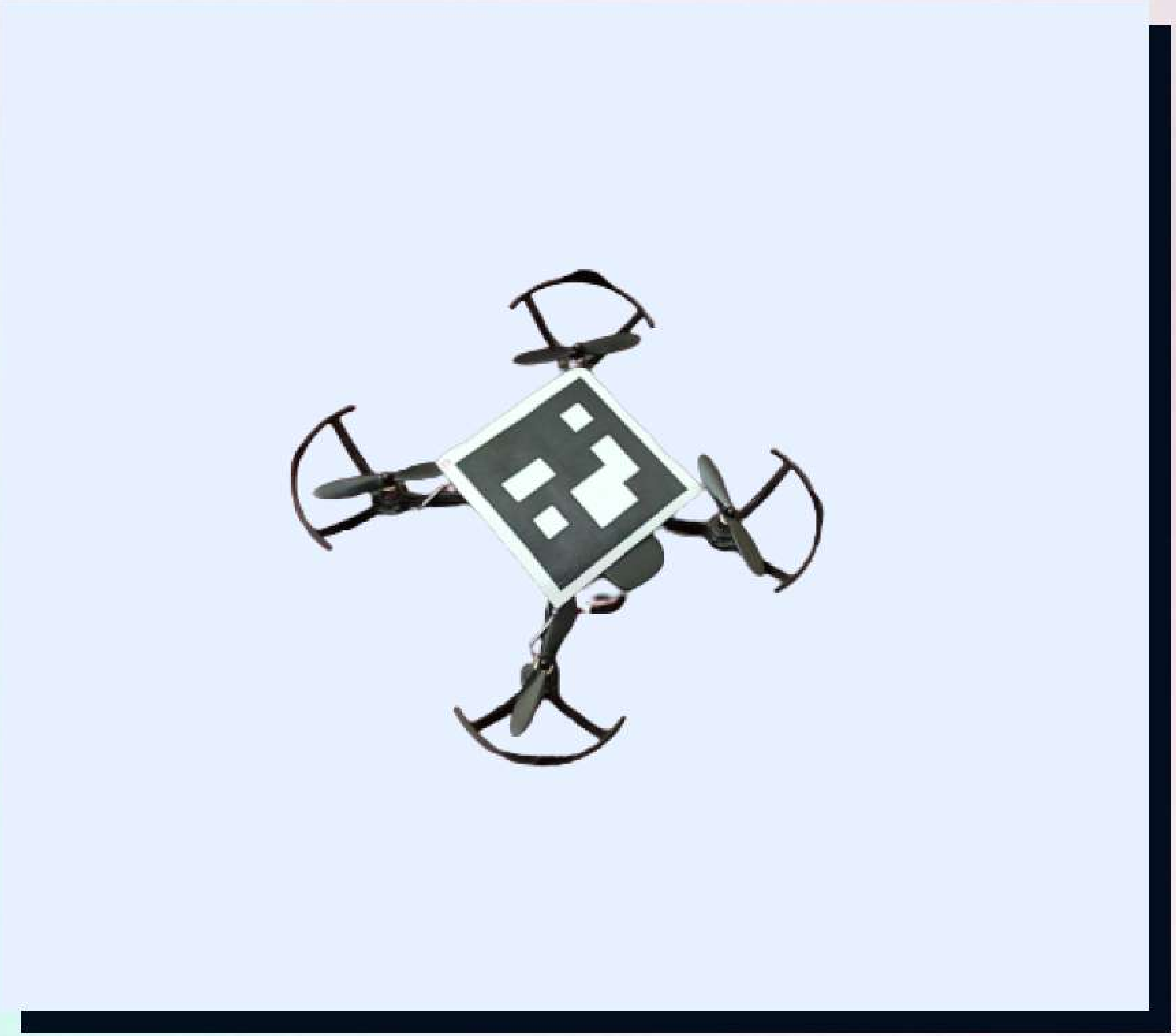
- Function to communicate with the drone using MSP packets were created

- Function parameters were changed and sent to the drone based on desired behaviour, like throttle, roll, yaw etc.

- Sensor data was also obtained by sending empty payload and then adding the data to “Log Files” for later analysis.



TASK 2

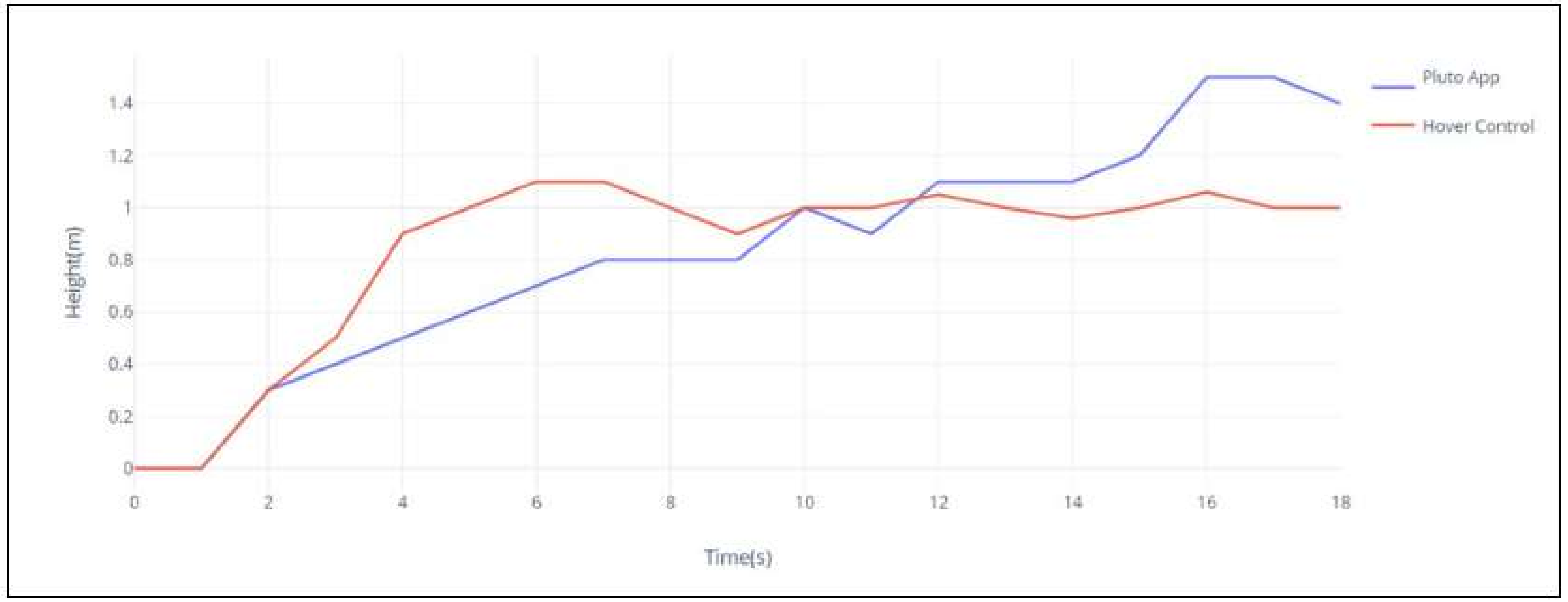


1. **Hovering**

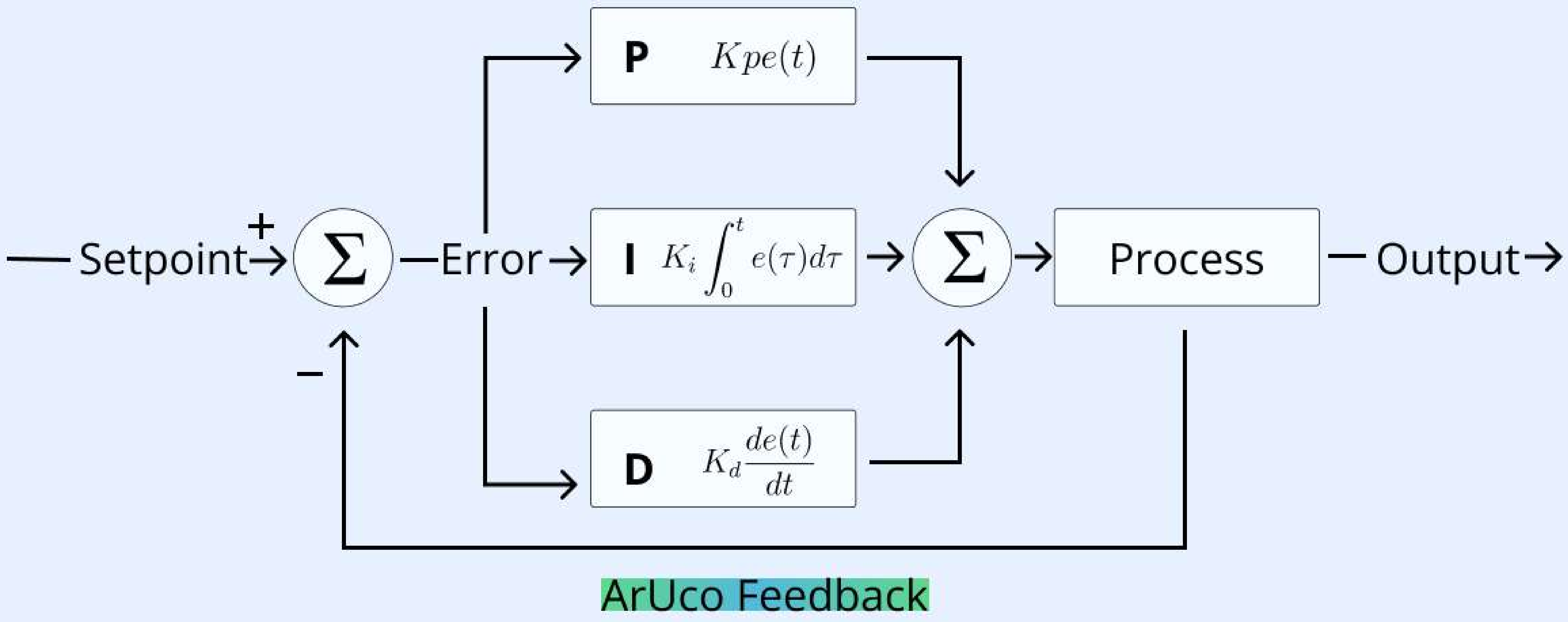
- One of the best controllers that are currently used is Fractional Order PID.
- We settled at the PID controller following the non Fractional dynamics of our system and relatively fewer data for determining the fractional dynamics.
- To determine the exact index we need excessive data from related sensors which was not available.

- The camera positioned on the top was used to calculate the current height of the drone, which acted as the feedback for the control loop.

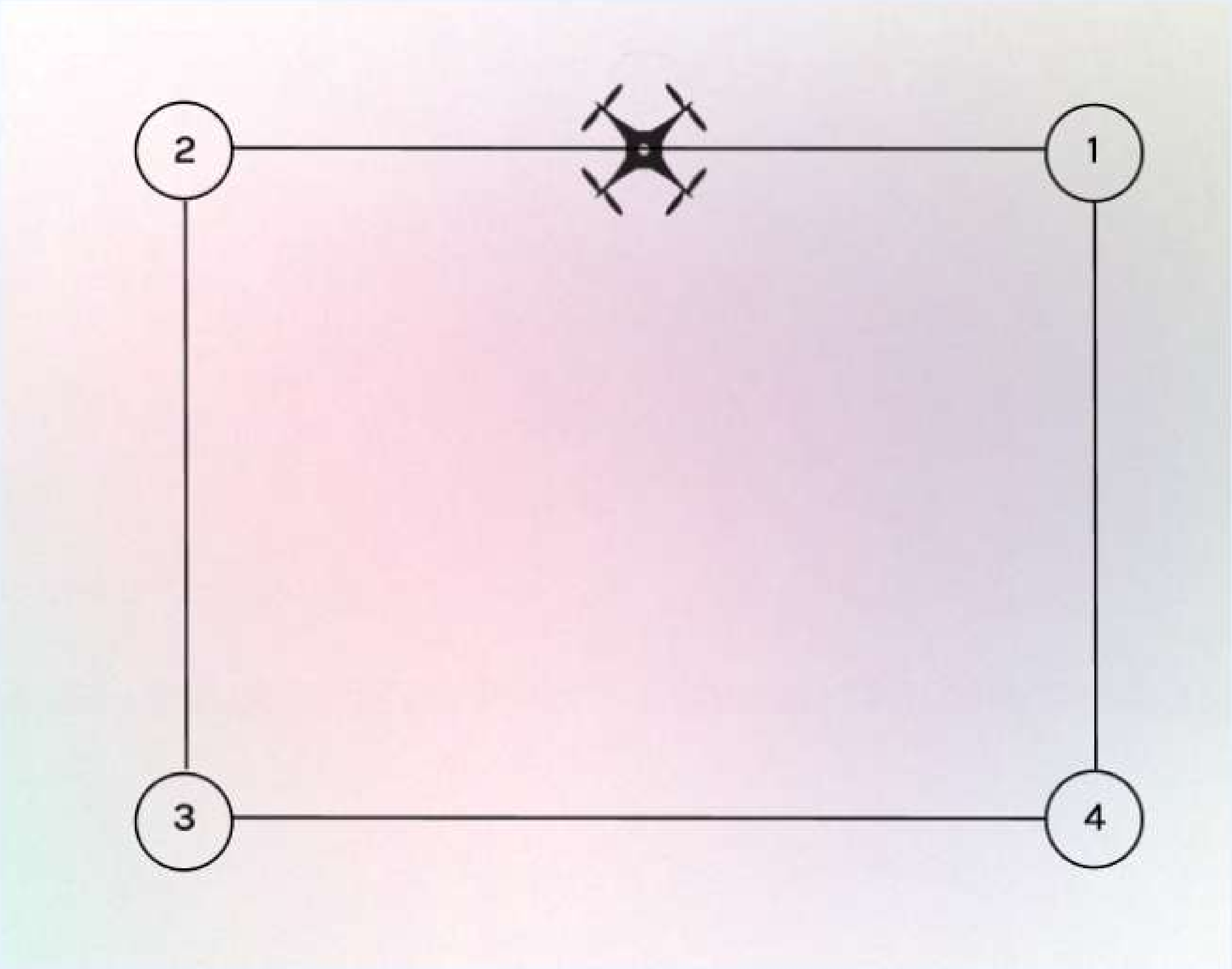
- We started with the conventional way by increasing kp until the system began oscillating and further determining ki and kd through an observational approach reaching to a relatively stable system.



- The drone's motion was then controlled using a PID control loop, where the control parameters (Kp, Kd, Ki) were manually tuned based on the drone's responses.



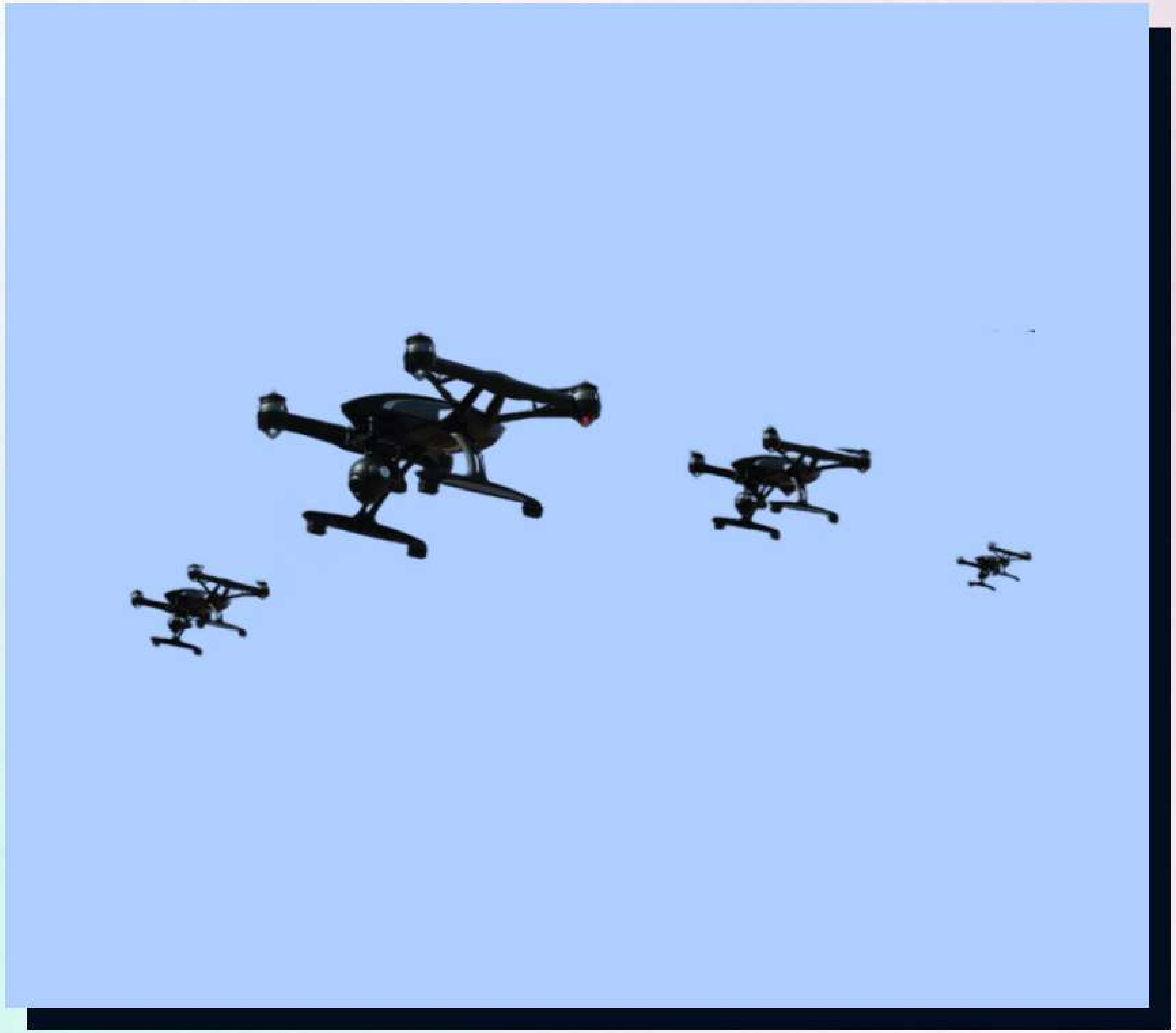
TASK 2



2. Rectangular Path

- 4 setpoints were created that act as the vertices of a rectangle
- A modulus error was calculated and when less than the threshold value, the setpoint was switched
- The next setpoint became the centre and drone moved to gaining aid from PID controller

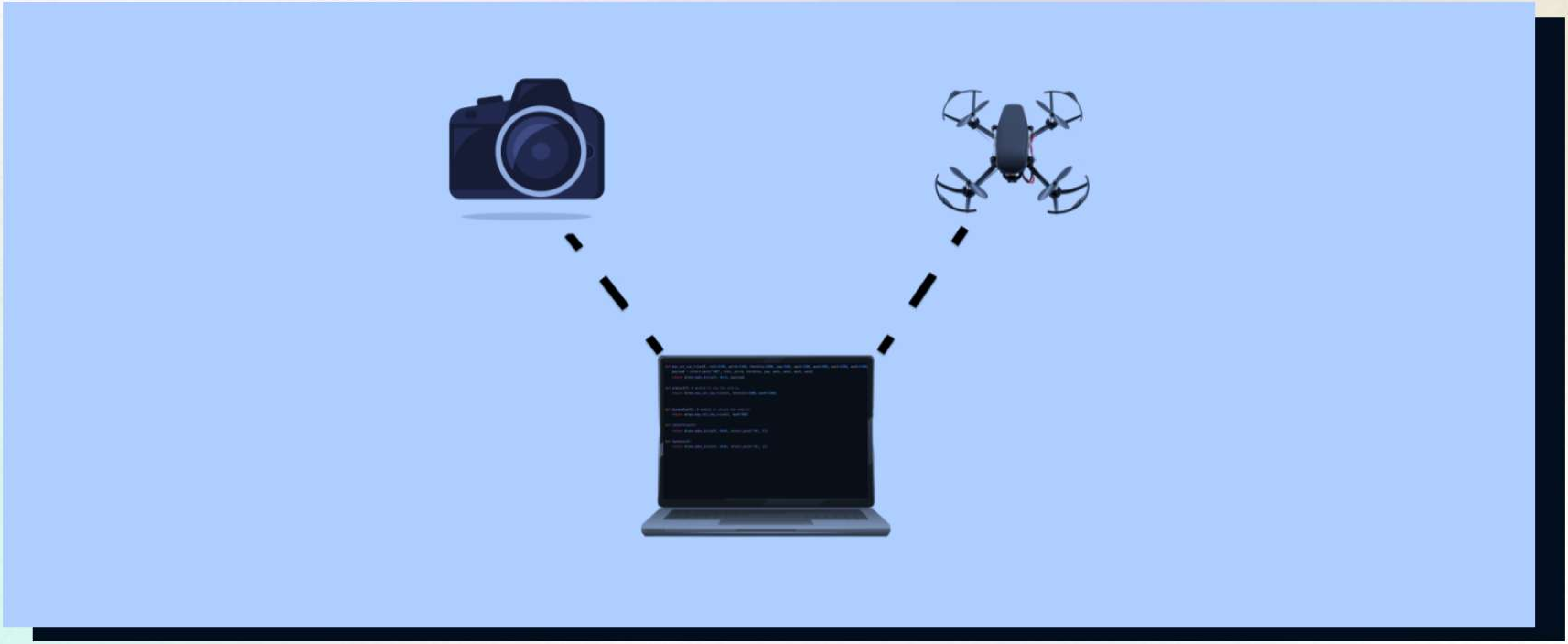
TASK 3

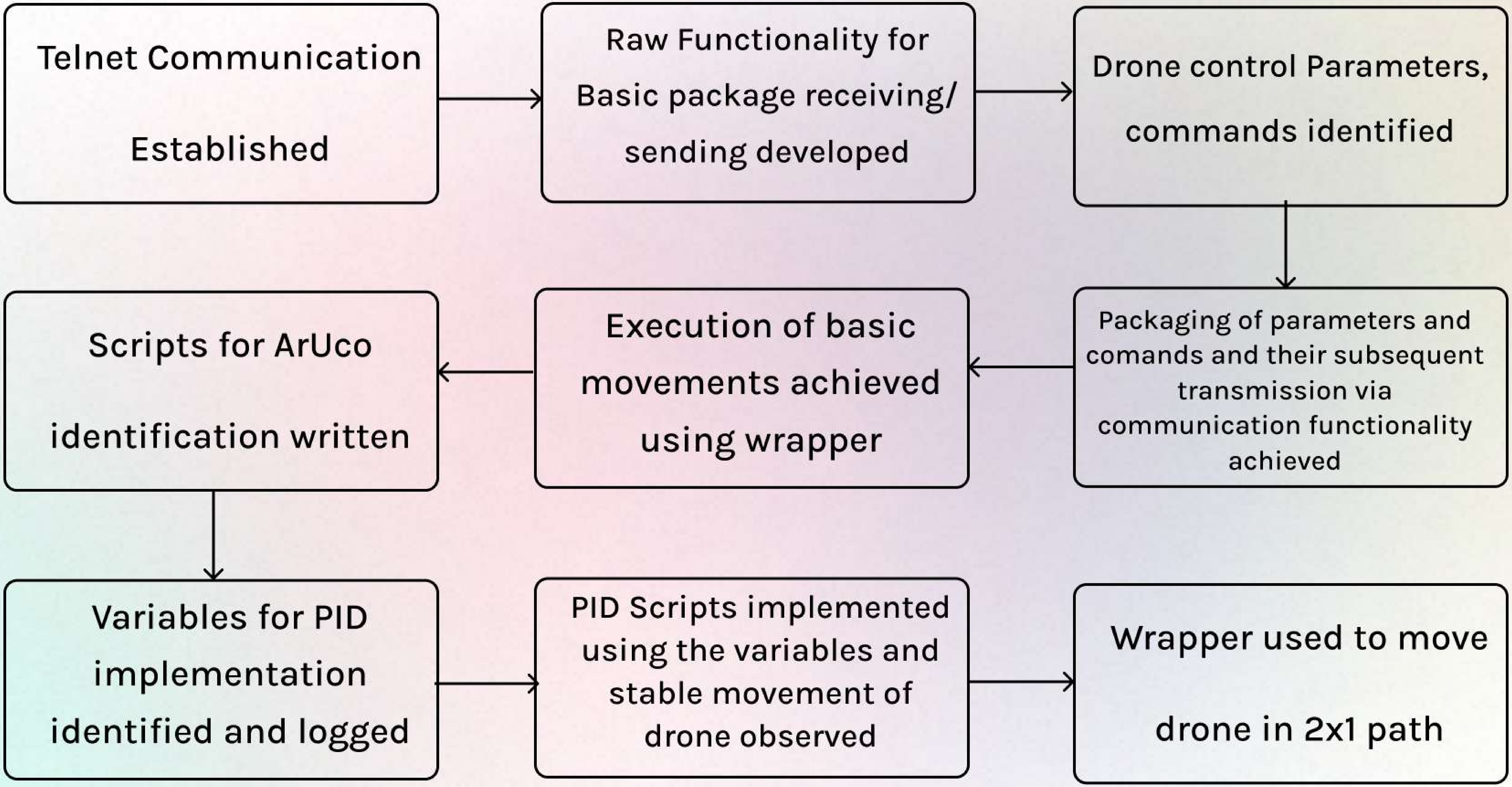


- The drones are set in both Station and Access Point Mode and connected to the device hotspot.

- The two drones are then controlled using two separate threads, each with its own ArUco associated with a unique ID. The objective is to have the second drone maintain a specific distance from the first one and use the first drone's 3D pose as a setpoint goal if the distance grows beyond a certain range.

- The euclidean distance between the two drones is calculated and stored, and the first drone's pose is used as the new setpoint goal for the second drone if necessary.




Telnet Communication Established
Raw Functionality for Basic package receiving/ sending developed
Drone control Parameters, commands identified
Packaging of parameters and comands and their subsequent transmission via communication functionality achieved
Execution of basic movements achieved using wrapper
Scripts for ArUco identification written
Variables for PID implementation identified and logged
PID Scripts implemented using the variables and stable movement of drone observed
Wrapper used to move drone in 2x1 path