IMPLEMENTATION OF UNMANNED AERIAL VEHICLE

SONU PRASAD, VINO WILSON, RONIT SINGH, YASH BODKE

**Department of Computer Engineering, Sinhgad Institute of Technology.
Lonavala, Maharashtra, India.**

sonuprasad.official@gmail.com,wilson.vino79@gmail.com,ronit.singh93@gmail.com

ABSTRACT—*Today, robotics and multirotor are huge industries emerging in the market. The increasing complexity and versatility of the drone overall increases the sophistication of the drone. In this paper we have tried to illustrate the making of the quadcopter drone by assembling transmitter and receiver on both the ends and controlling the quad using only the ground station module i.e. without use of any Remote control device. Transmission of data is via small packets via infrastructure network which is converted into a command for quadcopter that operates on a remote site.*

I. Introduction

The Quadcopter is proving to be a versatile tool that appears likely to support a number of markets and missions. Military missions, of course, are beyond the scope of our current project so we evaluated potential missions in the commercial and industrial sector that would be appropriate for the Quadcopter that we are building. For example, we were thinking that the local police here on campus that have to spend many hours looking at parking permits on cars could send the helicopter out down the car line and allow an operator to watch the onboard camera to see the permits. This would cut down on man hours as well as not wasting gas by driving around to inspect cars. Additionally, the Quadcopter could be used during a campus incident to assess a dangerous situation without putting officers or first responders in harm's way.

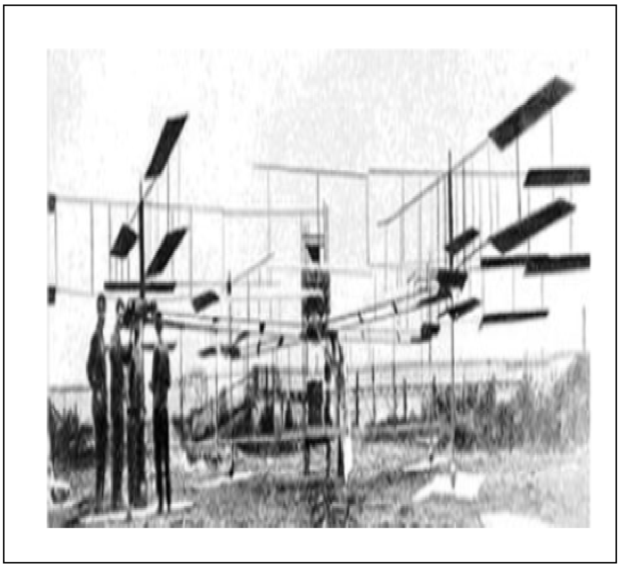
Other ideas include inspecting a pipeline. Our Quadcopter would easily go down a pipeline and allow an operator to watch the onboard camera and check for problems in the pipeline. Pipelines often pass through rugged terrain with very poor roads and using a Quadcopter would mean they wouldn't have to drive the entire route. They should be able to go down the pipeline a lot faster and also they would be able to maneuver the Quadcopter when they needed to inspect a certain area in closer detail or even from other angles. It might also be possible to install an infrared imager on the Quadcopter to look for areas of different temperature which might indicate a spill or a weakness in the pipeline.

II. Problem Definition

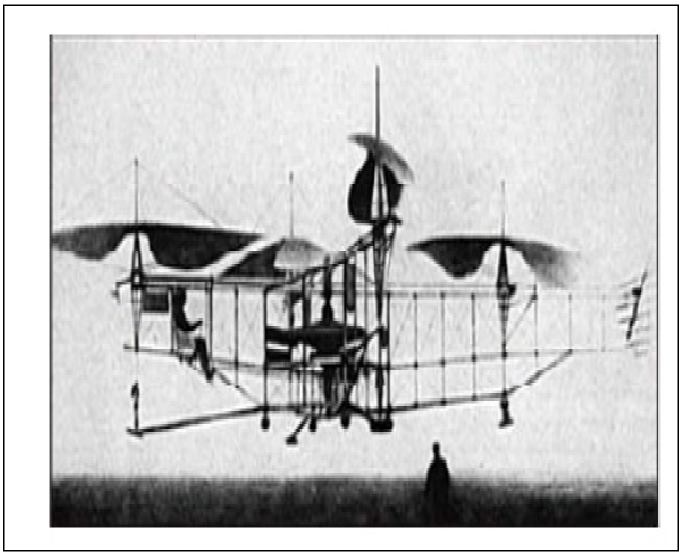
The military use of unmanned aerial vehicles (UAVs) has grown because of their ability to operate in dangerous locations while keeping their human operators at a safe distance. The larger UAVs also provide a reliable long duration, cost effective, platform for reconnaissance as well as weapons. They have grown to become an indispensable tool for the military. The question we posed for our project was whether small UAVs also had utility in military and commercial/industrial applications. We postulated that smaller UAVs can serve more tactical operations such as searching a village or a building for enemy positions. Smaller UAVs, on the order of a couple feet to a meter in size, should be able to handle military tactical operations as well as the emerging commercial and industrial applications and our project is attempting to validate this assumption.

III. Historical Development of UAVs

- Breguet-Richet Gyroplane (1907) a four rotor helicopter de- signed by Louis Breguet. This was the first rotary wing aircraft to lift itself of the ground, although only in tethered right at an altitude of a few feet. In 1908 it was reported as having own 'several times', although details are sparse.



- Oehmichen No.2 (1920) Etienne Oehmichen experimented with rotorcraft designs in the 1920s. Among the six designs he tried, his helicopter No.2 had four rotors and eight propellers, all driven by a single engine. The Oehmichen No.2 used a steel-tube frame, with two-bladed rotors at the ends of the four arms. The angle of these blades could be varied by warping. Five of the propellers, spinning in the horizontal plane, stabilized the machine laterally. Another propeller was mounted at the nose for steering. The remaining pair of propellers were for forward propulsion.



IV. Motivation of the Project

Motivation of the Project Latest Technology has surpassed the historic technologies. If use of technology is increased in Autonomous drones then more profit can be gained with the optimal use of resources. It will be also helpful in avoiding future risks of accidents as well as resource management in short amount of time. As the technology is increasing and the daily users are not able to cope or serialize up with different types of sensors, it motivated us to build a economically proven, efficient and resource driven device which makes optimal use of resource at run time without using too much of energy for redundant purposes.

V. *Hardware*

To make this quadcopter autonomous we used ardupilot APM 2.6 to program and applied PID algorithm to calculate output values of motor commands by using input values from transmitter and sensors. We used an inertial measurement unit (IMU) sensor which give values regarding angles and angular velocities of quad copter frame. The hardware requirement for the drone-

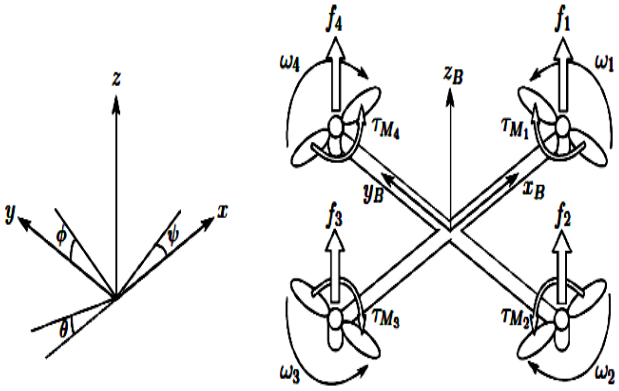
1. Chipset is Ardupilot apm 2.6
2. 4 ESC’s (Electronic speed controller)
3. Motors, propellers, batteries, xt60 connectors

4. Telemetry
5. GPS and compass
6. Gimbal
7. Few other sensors
8. A Microsoft or Linux based pc.

VI. Mechanism

Each rotor produces thrust and a torque about its centre of rotation and this forces are used to fly and move quad copter.

Two rotors mounted on opposite arms of quad copter are set into clockwise and into anticlockwise direction. This orientation of motors and your direction of rotation cancels all the torque generated given the speed of motors are same.



VII. Software Interface Description

In this project, we will use Python Libraries for UI. UI will have simple buttons for basic tasks (Take O_/Land, Start/Pause/Stop Recording and Enable/Disable Tracking Mode) And Will use key-

Board inputs and game controller for manual control over AR. Drone. Operating system used is Linux Ubuntu or Microsoft Windows whichever is preferable. Initially Mission planner is used for configuring the drone as the chipset is Ardupilot APM 2.6 which is officially supported.

The APM board runs the Busy Box based GNU/ Linux distribution with 2.6.47 kernel. The software of the apm not only provides communication but also takes care of stabilization and assisted maneuvers.

Over the command channel the user controls the drone, i.e. requests it to take-off, land, calibrate sensors, PWM or motors modulation etc. However, the most used commands are roll, pitch, yaw, throttle.

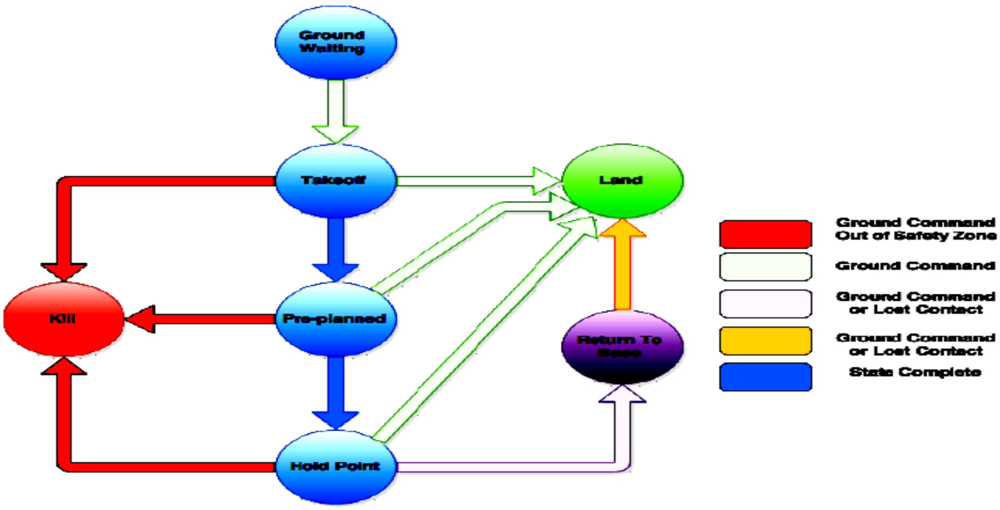
The navdata channel provides the drone status and preprocessed sensor data. The status indicated whether the drone is flying or is being calibrated, which algorithms are initialized, the current speed of altitude controllers etc. The sensor data contains latest roll, pitch, yaw, altitude, battery state.

The Ground station is programmed in Python language on Windows platform. Dronekit-Python is used to program Drone application on companion computer.

DroneKit-Python allows developers to create apps that run on an onboard companion computer and communicate with the ArduPilot flight controller using a low-latency link. Onboard apps can significantly enhance the autopilot, adding greater intelligence to vehicle behaviour, and performing tasks that are computationally intensive or time-sensitive (for example, computer vision, path planning, or 3D modelling). DroneKit-Python can also be used for ground station apps, communicating with vehicles over a higher latency RF-link. The API communicates with vehicles over MAVLink. It provides programmatic access to a connected vehicle's telemetry, state and parameter information, and enables both mission management and direct control over vehicle movement and operations.

VIII. State Diagram:

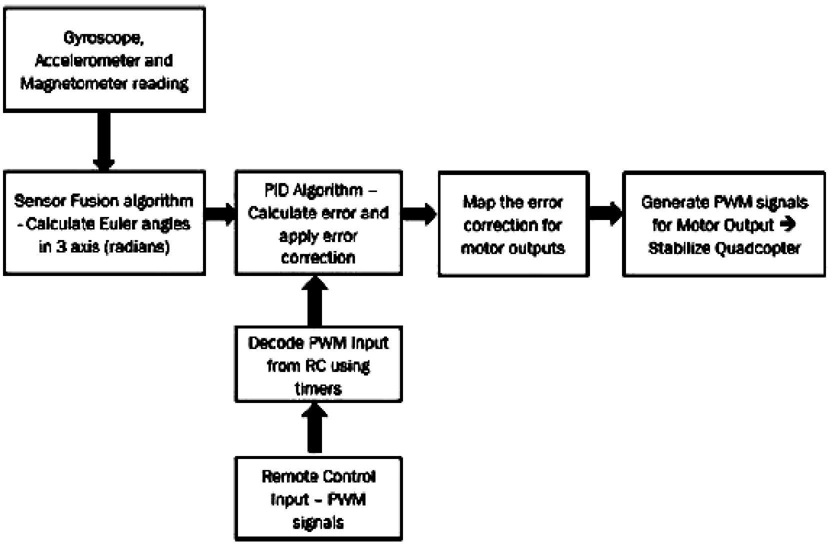
The states are represented in ovals and state of system gets changed when certain events occur. The transitions from one state to the other are represented by arrows. The Figure shows important states and events that occur while creating project.



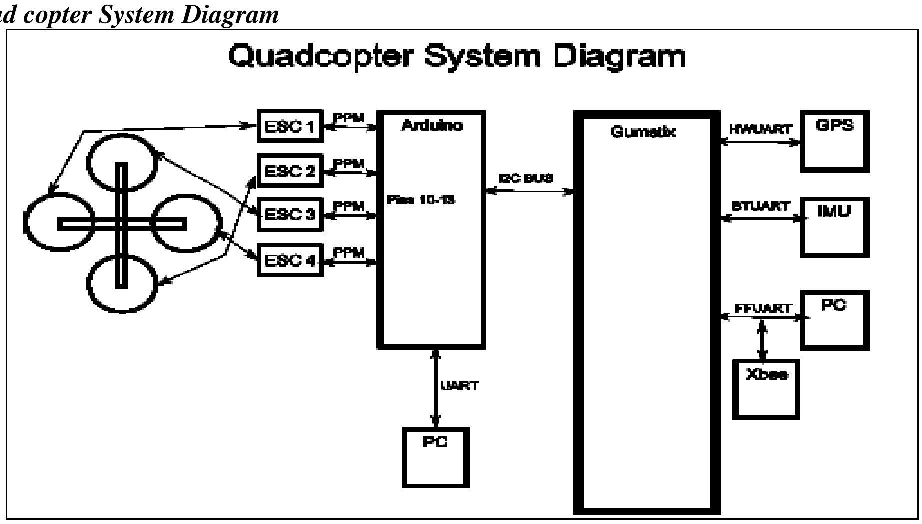
IX. Detailed Design Document

1. Architectural Design

A description of the program architecture is presented.



2. Quad copter System Diagram



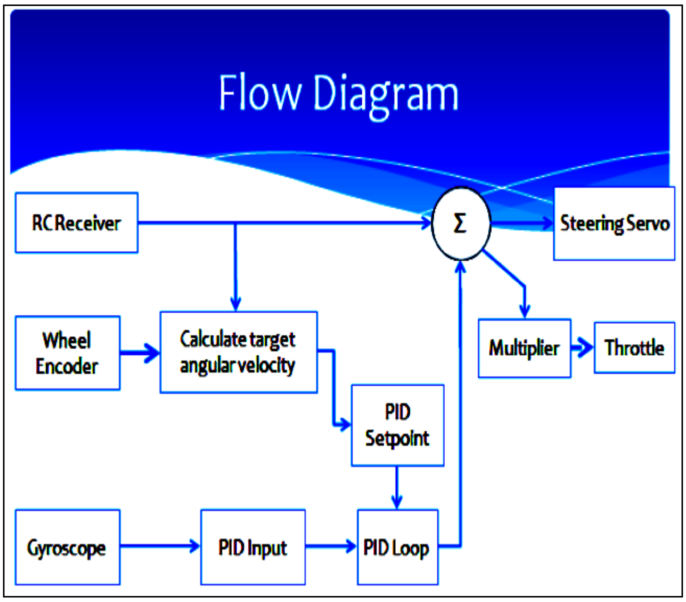
-UART-Universal Asynchronous receiver transmitter
-ESC- Electronic speed controller
-HWUART-hardware UART
-BTUART-Bluetooth UART
-IMU-Inertial measurement unit

3. Functional Model and Description

A description of each major software function, along with data flow (structured analysis) or class hierarchy (Analysis Class diagram with class description for object oriented system) is presented.

a) Data Flow Diagram

This section shows the relevant data flow from various sources.



X. ***Non Functional Requirements***:

- **Interface Requirements**

We will use Python Graphics Libraries for UI and android application also. UI will have simple buttons for basic tasks (Take Off/Land, Start/Pause/Stop Recording and Enable/Disable Tracking Mode) And Will use keyboard inputs for manual control over Drone.

- **Performance Requirements**

For The performance issue, target tracking is the most important functionality of the project to be considered. In Order not to lose the targets location while following it, detection of the target must be finished as fast as possible. In The project, it is considered to be over within 200 ms . Other Than that, bandwidth of the communication between the quadcopter and the controlling computer should be large enough to transmit 720p@30fps Video with a proper encoding.

- **Software quality attributes**

Availability - As the application for running the quadcopter i.e. Mission planner is based on open source and is compiled in java, C and visual Basic it is easily available and reliable,
Modifiability: The code can easily be ported to run in different operating system as it is compiled in Java.
Reusability: The same section of the code can be reused to reduce code size and redundancy of data and code as well.
Performance: The performance of the code as well as the quadcopter can be optimized by fine tuning the PID (Proportional Integral Derivative).
Security: The application as well as the data sent from the application to the quadcopter is secured but can be exploited by exploiting and running a backdoor in the processor of the drone.
User adaptability: The application is easily adaptable to the user because of the intuitive graphical user interface in the application.

XI. UAV application's

• Modern Warfare / Military purposes-Modular UAV's can be put into military use as they have the capacity to carry sufficient amount of payload. Weapons can easily be attached to the UAV, s which can be then triggered by a remote machine communicating with the drone. The drones can identify enemies and take the required action without human intervention.

- Monitoring
- Civil engineering sites
- Waterways and shipping
- Oil and gas pipeline
- Forestry
- Fishery Protection
- The countryside
- Pollution Control and Air Sampling
- Crop Performance
- Litter on beaches and in parks

Further there are additional applications of drone which can be put into use such as:
1. Videography / photography
2. Disaster response
3. Education
4. Aviation, Mapping
5. Maritime
6. Civil engineering

XII. *Future scope*

As of today's industry, the use of drones is very limited to high profile industries or in modern warfare. This should be expanded and the use of drones can be put into use of daily life. The drones equipped with various sensors in an embedded chip will help to reduce the worst conditions such as draught or heavy rain. Can be used as a fire extinguisher by installing proper fire sensors. Smart UAV drones can be used for carrying heavy payloads.

XIII. *Conclusion*

Our ultimate goal for this project was to design a quad copter UAV with live video surveillance. Very early in the project we discovered that quadcopter designs and parts were readily available and open source. In summary, the fundamentals of flight dynamics of the quad rotor aircraft were investigated and a full mathematical non-linear model is implemented using classical mechanics. In this document we provided details of architecture and initial design of our project which is Quad copters and we also provide details of the implementation of our project with respect to modules, viewpoints and data design and software design. Although a lot of work remains, we continue to be optimistic that we will complete the project on schedule. When the basic flight control systems are complete, the Quadcopter will be ready for experimental missions, at that point the project could go in a variety of directions since the platform seems to be as flexible as we initially intended. As a team, we can completely change what function it performs and we are able to integrate any technology that would prove to be useful. For future development, we will try to track multiple targets consisting of humans and vehicles. The device could also be programmed to perform a systematic sweep over a large area for search and rescue purposes. Overall a computer controlled system will eliminate human errors and allow for safe, consistent flying.

X. REFERENCES

1) Websource : Ardupilot.com for reference regarding main chip and some commands.
2) Websource : Qgroundcontrol for mavlink protocols.
3) Hardik Modh ”Quadrotor An Unmanned Aerial Vehicle” in journal IJEDR Volume 2, Issue 1 pp. 1299-1303 in 2014
4) P. Pounds, R. Mahony, and P. Corke, Modelling and Control of a Large Quadrotor Robot, in Control Engineering Practice, vol. 18, pp. 691 699, 2010.
5) P. Pounds and R. Mahony, Design principles of large quadrotors for practical applications, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 3265 3270, May 2009