Performance Prediction using Exploratory Data Analysis with the Attitude Parameters of a Quadcopter

Darshit Dhake[1], Himanshu Rawat[2], Swapnil Mohanty[3], Shardul Mehetar[4]

[1-2]Students, Dept. of Computer Science, Lokmanya Tilak College of Engineering, Navi Mumbai, Maharashtra, India
[3]Students, Dept. of Computer Science, National Institute of Technology, Rourkela, Odisha, India
.[4]Students, Dept. of Computer Science, Terna Engineering College, Navi Mumbai, Maharashtra, India

---

Abstract- *Drones are widely used in the world for various applications ranging from a drone racing and until surveillance purposes. The performance of a drone could be predicted using exploratory data analysis thereby they could be maintained for safe and reliable field operations. Various techniques are being conducted in the industry to predict the performance of the drone in the industries. One such analysis would be the using the attitude parameters collected from the SD card of any flight controller used in quadcopters. Exploratory Data Analysis is applied to the obtained parameters and results are further compared for prediction of safe operation and unsafe operations for the next flight. We perform post flight analysis, where in the information is obtained from data flash log files when loaded in the mission planner software. The extracted data is sent into EDA model which predicts the reliable operation by using the mode function in the dataframe.*

***Keywords*-Drones, Exploratory Data Analysis, Performance prediction**

I. INTRODUCTION

The drones these days are integrated to several applications in the current scenario ranging from racing to surveillance. Incorporation of better performance is in high demand in these days when it comes to a fact of having full mission flights. In this paper, we talk about the attitude parameters that are out from a PD controller used in the quadcopter. The copter taken into the consideration is the micro drone customized based for the purpose of hobbyist flying. Here the intention of the pilot is to learn the drone piloting and hence it is important to know if the drone would fly the very next time. The drone has Ardupilot based Pixhawk cube flight controller and the parameters are accessed via the Mission Planner Software. Post flight analysis is done using the telemetry and data flash log files. Data flash logs are preferred as the communication between the air unit and ground unit remains established throughout the mission flight. The parameters from the dataflash log files cannot be viewed initially as they have .bin or binary format. In order to have a visualization of the parameters they are converted to .log format through the file convert options available in the mission planner software. Here as attitude analysis is performed, we are interested to look upon the data under ATT parameter in the binary file. The binary file is fed as input to the python code which contains the programming to extract data under sub parameters like des_roll, roll, des_pitch, pitch, des_yaw, yaw, time etc. We are more particular in understanding the patters of the actual and desired values of roll, pitch and yaw values. These attitude parameters are controlled using the quadcopter's motor speeds. By adjusting the speeds of the motors in various combinations, the quadcopter can achieve the desired roll, pitch, and yaw angles. To maintain stable flight, the attitude control system constantly monitors the quadcopter's orientation and makes adjustments as needed. The following process is followed under two different scenarios. One where there was a smooth flight during the mission and the second when there was a crash during the mission. The endurance of flight in both the cases is about 14.5 minutes. Exploratory Data Analysis is performed in both the cases and the performance of the flight is predicted further. This paper is divided into different sections. The drone was customized as described in [1]. The description of the paper gets initiated by considering the quadcopter dynamics as in [2]. This is described in section II. Section III talks about the software requirements for this project. Mission planner, python, UAV Log Viewer along with other packages need to be installed in the python environment in order to get the required results. Section IV, talks about the approach towards the project or the pseudo code of the execution of the python program. Section V talks about the discussions upon the results achieved. Further in section VI future works are discussed.

Quadcopters

Flying machines or air vehicles are generally classified based on the way the wings are mounted. Hence an aerial vehicle could be either fixed wing, e.g., airplanes, or rotating wing, e.g., helicopters. A multi-rotor helicopter with the ability to VTOL is known as a quadrotor or quadcopter. Four rotors lift them and move them forward. They are classified as rotorcraft among aircraft since they are lifted and propelled by a group of rotors. Two identical fixed pitch propeller pairs, two rotating in the clockwise (CW) and two in the counter-clockwise (CCW) directions, are used by quad copters. These regulate lift and torque by varying RPM. Computer/electronic systems can regulate the motion of the vehicle by altering the torque load and thrust/lift characteristics of one or more rotor discs, which alters their rotation rate.

Figure 1: Quadcopter Under Consideration

A simple quad rotor is constructed through mechanical components such as frame and fixtures, electric motors and electronic components such as electronic speed controllers, sensors, on board microprocessor, batteries and GPS. Some of the sensors include the Inertial Measurement Sensors called as Inertial Measurement Unit (IMU) which is the combination of gyroscope (to measure the direction) and accelerometer (to measure the acceleration). The altitude sensor provides the information about the height of the quadrotor from the ground and is achieved through SONAR. The Bluetooth or the telemetry or the Wi-Fi serves to communicate between the quadrotor and the ground station. In case of indoor navigation, the GPS cannot provide accurate data and so the LASER sensors play a vital role for enabling the navigation by generating a local map for the purpose of finding the position of the quad rotor. This process is actually called Simultaneous Localization and Mapping (SLAM). Some of the controllers used for the autonomous navigation are PIXHAWK, APM etc. The need for these controllers is to provide adequate interfaces for stabilizing the autonomous motion of the copters. The computing for the navigation process could be achieved either onboard or off board. The algorithm is implemented in the on-board processor in the former case or in the ground station in case of off board processing.

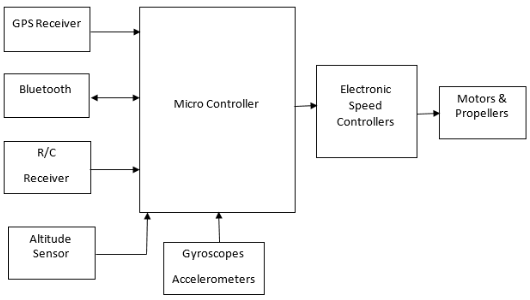

Figure 2: Block Diagram of a quadcopter

CONTROL THEORY & MODEL EQUATIONS-NEED

The propellers and motors are the main components that enable the flying operations of the quad rotors. Although they are chosen identical, the manufacturing defects associated make them behave in a different way reflecting in the rotation rates of each of the rotor. Signals from microcontroller are the PWM signals applied to each of the motors do generate different forces causing irregular rotation rates. Figure 3 shows the driving scheme employed in quadrotors. Therefore, the system requires the introduction of a control feedback scheme in order to decrease the error between the desired and measured value of the rotor speeds.

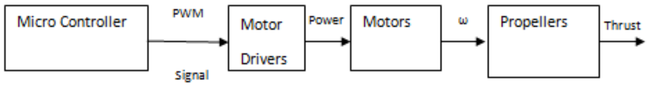

Figure 3: Motor/Propeller Driving Scheme

QUADCOPTER DYNAMICS

The block diagram depicted in figure 1 can serve as a representation of the drone under consideration's overall control architecture. The nested feedback loop is made up of the inner attitude and outside position control loops.

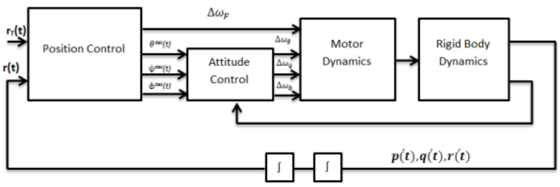

Figure 4: Position and attitude control via nested feedback loops

Dynamic Modelling and Analysis

Figure 5 depicts the exact free body and systems of drone under consideration. It comprises of the fixed world frame (W), to which the drone's center of mass attaches the body frame. To simulate the rotating movement of the quadrotor in the W frame, the Z-X-Y Euler angle convention is utilized. We rotate via Zxy by yaw, then about the intermediate x-axis by roll, and lastly about the Y_b by pitch to get at the B frame. As a result, the rotation matrix R from the W to B frame is be represented as follows:

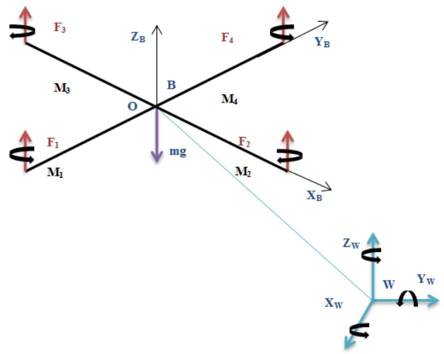

Figure 5: Figure of the Overall Free Body

$$R = \begin{bmatrix} c\psi\, c\theta - s\phi\, s\psi\, s\theta & -c\phi\, s\psi & c\psi\, s\theta + c\theta\, s\phi\, s\psi \\ c\theta\, s\psi + c\psi\, s\phi\, s\theta & c\phi\, c\psi & s\psi\, s\theta - c\theta\, c\psi\, s\phi \\ -c\phi\, s\theta & s\phi & c\phi\, c\theta \end{bmatrix} \quad (1)$$

where sine and cosine are represented by the letters s and c, respectively. If r is the centre of mass' position vector in the W, then equation describes the center of mass acceleration.

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ \sum_{k=1}^{4} F_k \end{bmatrix} \quad (2)$$

Where $\ddot{r} = \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{Z} \end{bmatrix}$,

The quadrotor has a mass of m., while g is gravity causing acceleration. r,q and p stand for the angular velocity of robot in B. The following diagram illustrates the relationship between the pitch,yaw angles and roll in relation to these derivatives.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \cos\phi\sin\theta \\ 0 & 1 & \sin\phi \\ \sin\theta & 0 & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (3)$$

Motor Control

As a result of rotation by the rotor, a vertical force Fk is generated and is given by,

$$F_k = k_F \omega_k^2 \quad (4)$$

Each rotor's M_k moment is parallel to rotating plane of the blades. Rotors 3 and 1 generate moments in the – Z_b direction. Moments are produced in the direction of Z_b by rotors 2 and 4. With M3 and M1 acting in the Z_b direction and M4 and M2 acting in the different directions, then quadrotor creates a moment that acts in the opposite direction to the direction the blades rotate. L denotes the separation between the rotor's rotation axis and Centre of the quadrotor. The XB-YB-ZB axis, the MI(moment of inertia) matrix, I, is linked by center of mass, is produced by weighing the constituent parts of the quad rotor. In terms of angular acceleration,

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_1 - F_3) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (5)$$

The angular velocity components for the drone in the B are p, q, and r. The rotor moments caused by the rotors' angular velocity are given by

$$M = k_M \omega_k^2 \quad (6)$$

When the nominal thrusts are equivalent to the force of gravity, the hovering process takes place and is provided by

$$F_k = \frac{mg}{4} \quad (7)$$

The motor speed then would be:

$$\omega_{k0} = \omega_h = \sqrt{\frac{mg}{4k_F}} \quad (8)$$

Attitude Control

Attitude parameters are essential aspects of a quadcopter's control system that determine its orientation and stability in three-dimensional space. A quadcopter has six degrees of freedom: three rotational (roll, pitch, and yaw) and three translational (surge, sway, and heave). The attitude parameters specifically refer to the rotational aspects, which are roll, pitch, and yaw.

Roll: Roll refers to the rotation of the quadcopter around its longitudinal axis, which is an imaginary line running from the front to the back of the vehicle. When the quadcopter rolls, one side of the vehicle moves upward while the other side moves downward.

Pitch: Pitch is the rotation of the quadcopter around its lateral axis, which is an imaginary line running from one side of the vehicle to the other. When the quadcopter pitches, the front of the vehicle moves either up or down relative to the back.

Yaw: Yaw is the rotation of the quadcopter around its vertical axis, which is an imaginary line running from top to bottom. Yawing changes the direction the front of the quadcopter is facing without affecting its position.

Four distinct propeller speeds are input to the controller. The quadcopter's dynamic model is used to design the PD controller. Given that the associated vectors may be considered to be linearly combined, the desired rotor speeds are given by replacing (4) and (6) in (5), as illustrated below:

$$\begin{bmatrix}\omega_1^{des}\\ \omega_2^{des}\\ \omega_3^{des}\\ \omega_4^{des}\end{bmatrix} = \begin{bmatrix}1 & 0 & -1 & 1\\ 1 & 1 & 0 & -1\\ 1 & 0 & 1 & 1\\ 1 & -1 & 0 & -1\end{bmatrix} + \begin{bmatrix}\omega_H + \Delta\omega_F\\ \Delta\omega_\phi\\ \Delta\omega_\theta\\ \Delta\omega_\psi\end{bmatrix} \quad (9)$$

When the desired angular velocities for each rotor ω_k^{des} where(k=1,2,3,4), and hovering speed is ω_H. The proportional-derivative control rules govern the deviations that occur in pitch, yaw, roll, and net force along the z-axis, which are computed as.

$$\Delta\omega_\phi = k_{p,\phi}(\phi^{des} - \phi) + k_{d,\phi}(p^{des} - p)\}$$
$$\Delta\omega_\theta = k_{p,\theta}(\theta^{des} - \theta) + k_{d,\theta}(q^{des} - q)$$
$$\Delta\omega_\psi = k_{p,\psi}(\psi^{des} - \psi) + k_{d,\psi}(r^{des} - r)$$
$$\Delta\omega_F = \frac{m}{8k_F\omega_H}\ddot{Z}^{des} \quad (10)$$

Command of position

To enable the drone to follow the desired course $r_{i,T}$, the acceleration in command $\ddot{r}_{i,T}{}^{des}$ is based on the position inaccuracy and derived from the PD controller.

$$(\ddot{r}_{i,T} - \ddot{r}_i^{des}) + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i^{des}) + k_{p,i}(r_{i,T} - r_i) = 0 \quad (11)$$

Where $r_{i,T}$, r_i (i=1,2,3) are, respectively, drone's 3-D position and the planned trajectory. You should be aware that $\dot{r}_{i,T} = \dot{r}_i = 0$ for hover. The vehicle's orientation must be set close to zero throughout the duration of the flight. The motion equation that describes the common hover states can be linearized to produce this effect.

The value of nominal hover state ($r=r_0$, $\phi=\theta=0$, $\psi=\psi_T$, $\dot{r}_i = 0$ and $\dot{\theta}_i = \dot{\psi}_i = \dot{\phi} = 0$). Pitch and roll angles should not significantly alter while the aircraft is in flight. The requisite roll angles and pitch to produce motion is determined from equation (11) about these notional hovering states as indicated by the following equation:

$$\phi^{des} = \frac{1}{g}(\ddot{X}^{des}\sin\psi_T - \ddot{Y}^{des}\cos\psi_T)$$
$$\theta^{des} = \frac{1}{g}(\ddot{X}^{des}\cos\psi_T - \ddot{Y}^{des}\sin\psi_T) \quad (12)$$

Where $\ddot{X}^{des}$ and $\ddot{Y}^{des}$ are the intended X and Y accelerations. ψ_T is targeted yaw angle the same as the yaw angle that will be tracked ψ^{des}.

II. SOFTWARE USED

Mission Planner:

Mission Planner can be used to configure the autonomous vehicle or as an extra dynamic control strategy. Here are some examples of how Mission Planner might be used:

• Install the firmware (software) for the Pixhawk series autopilot board that runs the automobile.

• Set up, fine-tune, and setup the drone for optimal performance.

• Using simple point-and-click way-point entry on Google or other maps, plan, save, and load autonomous missions into your autopilot.

• Load and study the logs of mission generated by your autopilot.

Figure 6: Mission Planner Software - Homepage

UAV Log Viewer:

Unmanned aerial vehicles (UAVs), often known as drones, produce logs, which are analyzed and visualized by the software program known as the UAV Log Viewer. For UAV operators, engineers, and researchers, this tool is essential for gaining insights on flight performance, sensor data, and other crucial information gathered during UAV operations. Following are UAV Log Viewer's primary roles and objectives:

Log Analysis: The program examines log files produced by UAV flight controllers and extracts information on factors such as motor outputs, GPS coordinates, sensor readings, and flight characteristics.

The UAV Log Viewer transforms raw log data into illustrative displays like graphs, charts, and maps.

Gyroscopes, accelerometers, magnetometers, barometers, and other sensors' sensor data are all examined by the application. To maintain steady flight, it is essential to comprehend how the UAV interacts with its surroundings.
Mission Planning: This helps to improve mission planning for efficiency and precision.

Regulatory Compliance: For professional and research purposes, UAV flight operations may need to adhere to regulatory guidelines. UAV Log Viewer can assist in documenting flight data for compliance and reporting purposes.

Research and Development: Engineers and researchers can utilize UAV Log Viewer to gather data for improving UAV designs, testing new algorithms, and developing autonomous flight systems.

The output or pilot's visualization of the flight that crashed while on mission is shown in figure 7 below.

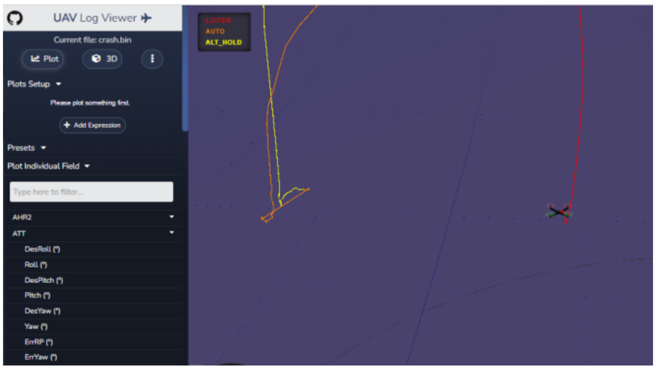

Figure 7: UAV Log viewer Outline

Python Environment:
Visual Studio Code is used as an IDE(Integrated Development Environment) to implement the corresponding programs using python containing the listed packages are Pandas, Numpy, Matplotlib, Scipy, Path, and Pymavlink .

III. EXPLORATORY DATA ANALYSIS

Exploratory analysis with drone data involves the use of unmanned aerial vehicles (drones) to collect various types of data, such as images, videos, and sensor readings, for the purpose of gaining insights and understanding about a particular area or subject. Due to the distinctive perspective and in-depth data that drones can supply; this strategy has become widely popular in a variety of fields. The following is a summary of significant points regarding exploratory analysis using drone data:

Data Gathering Drones have cameras, sensors, and other data collection equipment that allow them to gather data from a perspective that is difficult for people to reach. They might gather data from multispectral imaging, thermal imaging, LiDAR scanning, and visual data.

Applications: There are numerous uses for drone-based exploratory analysis. It is utilised in agriculture for yield estimation, disease diagnostics, and crop monitoring. In environmental study, drones are used to track animals, spot deforestation, and examine natural calamities. Urban planning is aided by aerial assessments of construction sites and infrastructure.

Data Processing: The data acquired by drones is frequently large and needs specialised processing. Image stitching, geo referencing, and 3D reconstruction are all standard techniques for converting raw data into useful representations.

Drones make extensive spatial analysis possible. They are capable of producing high-resolution maps and models that highlight patterns, anomalies, and changes in the terrain, which can benefit in decision-making and planning.

Drones are more cost-effective and efficient than traditional data collection methods such as piloted flights or ground surveys. They can cover wider regions in less time.

Environmental Monitoring: Drone-based exploratory analysis is critical for environmental monitoring. They can measure pollutants, habitat health, and ecological changes, all of which can help conservation efforts.

Drones play a key role in checking essential infrastructure such as bridges, electricity lines, and pipelines. They lower human danger while providing precise visual data.

Regulatory limits, privacy issues, and technological limitations in terms of battery life, data transfer, and poor weather conditions all pose hurdles to drone-based exploratory investigation.

Integration with AI: Integrating artificial intelligence and machine learning algorithms with drone data can automate tasks like object detection, anomaly identification, and image classification, enhancing the efficiency of analysis.

Future Potential: As technology advances, drones are likely to become even more sophisticated and capable of gathering diverse data types. This will lead to expanded applications across sectors and improved accuracy in analysis.

IV. STEPS INVOLVED AND PROPOSED ALGORITHM

The steps involved in development of EDA implementation is enumerated as shown

- Load the binary file in the mission planner and get it converted to log file for the purpose of reference.
- Load the binary file in the code and collect the required parameters.

- From the collected parameters extract the required data of actual and desired roll, pitch and yaw values.
- Perform exploratory data analysis, preferably EDA (Exploratory Data Analysis) and save the data in a csv or an excel file.
- Observe the patterns and predict if drone crash is likely to occur.

The binary file is given as input to the python program. The pymavlink package associated with the flight controller extracts the requirements in the form of parameters containing des_roll, des_pitch, des_actual, Error_roll, error_pitch and error yaw. The data now is extracted to a table using the pandas package associated with python. Further the table is checked for null values and inf values. In our case, there wasn't any. Then the next process was to find the gain or loss margin between them. But when in particular cases where actual values and desired values where divided one by the other yielded infinite values. Hence their difference was considered. Their differences were plotted on a separate column. The algorithm for roll, pich and yaw values after observing the threshold limits are presented in the following table 1. It the obtained difference values were found to cross the threshold limits then negative (-1) is assigned and if not positive 1 (+1) is assigned. The mode of this series is considered as final output. If the mode of the series is -1 then the drone has high chances to crash the very next flight is attempted. In case +1 is the output then prediction predicts for safe operation for the next flight.

S.No	Attitude Difference Parameter	Threshold Value
1	Diff_roll	1
2	Diff_pitch	3
3	Diff_yaw	10

Table 1: The threshold limits

V. RESULTS & DISCUSSION

The results for a full non crashed mission could be seen below in the figure below. It describes the difference in roll, pitch and yaw values. Following figure shows the result of the crashed drone flight. The difference plotted against the time instances indicated a higher margin value in the difference values. The table 2 lists the prediction for the mission flight with crash and without crash.

Parameter	Non Crash	Crash
Roll_mode	1	-1
Pitch_mode	1	-1
Yaw_mode	1	1

Table 2: Prediction Table

Normal Mission Flight Results:

The following figures correspond to the

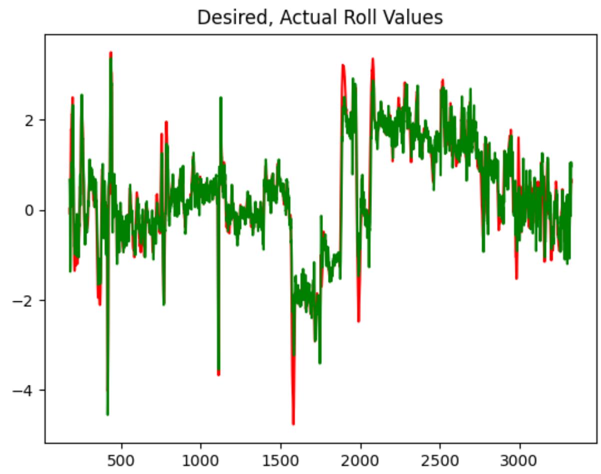

Figure 8: Attitude Roll Parameters

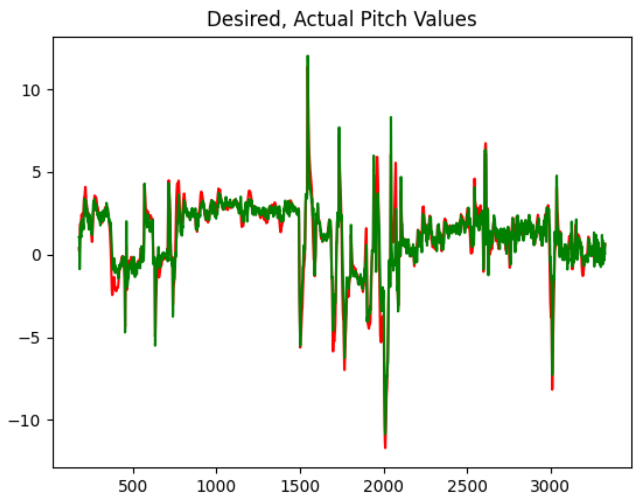

Figure 9: Attitude Pitch Parameters

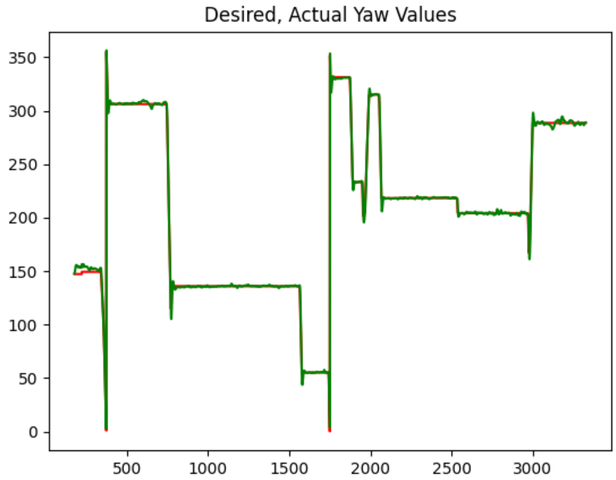

Figure 10: Attitude yaw Parameters

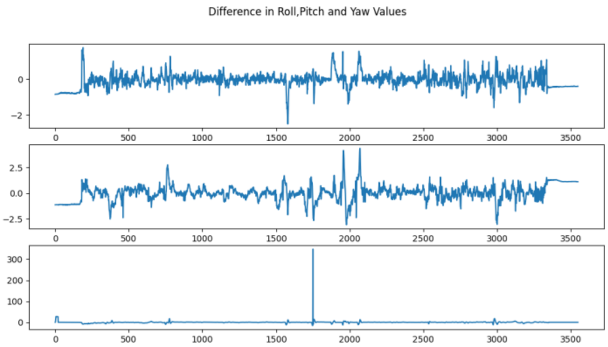

Figure 11: Attitude Roll, Pitch and Yaw Difference

Crashed Mission Flight Results:

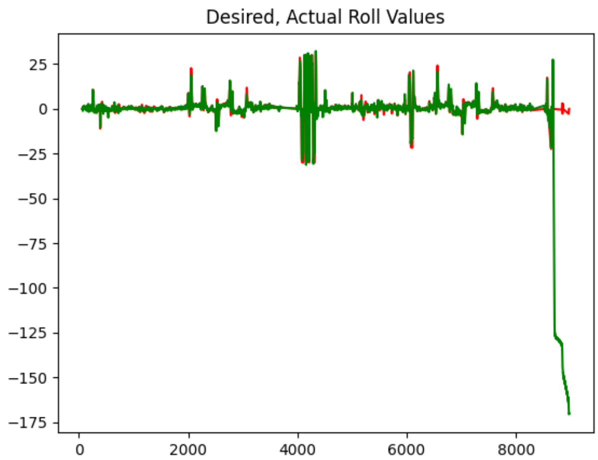

Figure 12: Attitude Roll Parameters

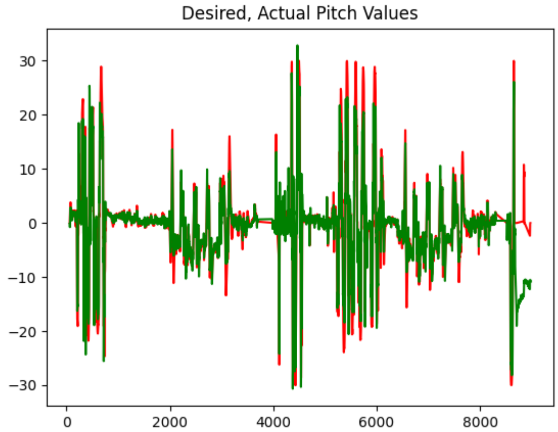

Figure 13: Attitude Pitch Parameters

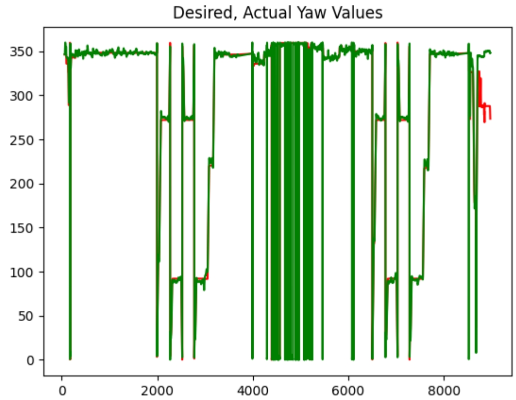

Figure 14: Attitude Yaw Parameters

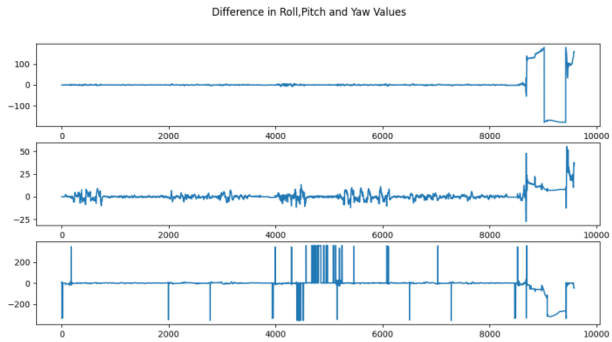

Figure 15: Attitude Roll, Pitch and Yaw Difference

VI. CONCLUSION

In conclusion, we have performed exploratory data analysis over the data extracted from dataflash log files. This is a post flight analysis and not a real time analysis. The extracted data fed to analysis model where in threshold limits are defined for their difference in attitude parameter values. The results are further shown in results column. The threshold limits defined can be referred to table 1 for further implementation process.

VII. REFERENCES

[1] Bouabdallah, S., & Siegwart, R. (2007). Full control of a quadrotor. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 153-158).

[2] Mellinger, D., & Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2520-2525).

[3] Lee, T. D., Leoky, M., & McClamroch, N. H. (2010). Geometric tracking control of a quadrotor UAV on SE (3). IEEE Transactions on Control Systems Technology, 18(3), 679-686.

[4] Faessler, M., Franchi, A., & Scaramuzza, D. (2017). Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. IEEE Transactions on Robotics, 33(6), 1464-1477

[5] Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robotics & Automation Magazine, 19(3), 20-32.

[6] Bresciani, T., & Chatila, R. (2013). Robust visual servoing for UAVs. IEEE Transactions on Robotics, 29(1), 33-46.

[7] Ma, J., & Hovakimyan, N. (2016). Adaptive backstepping control for quadrotor attitude and position tracking. Journal of Intelligent & Robotic Systems, 84(1), 227-244.

[8] Ryll, M., Sieberling, S., Bülthoff, H. H., & Giordano, P. R. (2013). A modeling approach for quadrotor vehicles subject to rotor aerodynamics. In AIAA Guidance, Navigation, and Control Conference (p. 4676). American Institute of Aeronautics and Astronautics.

[9] Exploratory Data Analysis in Remote Sensing" by Lu, D., & Weng, Q. (2007)

[10] "UAV-based Remote Sensing for Urban Vegetation Mapping Using Random Forest and Texture Analysis" by Wang, L., Sousa, W. P., & Gong, P. (2014)

[11] "Unmanned Aerial Vehicles for Urban Air Quality Measurements" by Zhang, C., Wu, Y., Yang, F., Zhang, J., & Xia, X. (2016)

[12] Drones in Environmental Science: A Review" by Anderson, K., & Gaston, K. J. (2013)

[13] Drones in Environmental Science: A Review" by Anderson, K., & Gaston, K. J. (2013)

[14] "The X-Y-Z Approach for Graphical Data Analysis" by Cleveland, W. S. (1993)