UAS Flight Data Research in Support of ASIAS (Aviation Safety Information and Analysis Sharing): Final Report

May 8, 2023

v 1.1

NOTICE

LEGAL DISCLAIMER

The information provided herein may include content supplied by third parties. Although the data and information contained herein has been produced or processed from sources believed to be reliable, the Federal Aviation Administration makes no warranty, expressed or implied, regarding the accuracy, adequacy, completeness, legality, reliability or usefulness of any information, conclusions or recommendations provided herein. Distribution of the information contained herein does not constitute an endorsement or warranty of the data or information provided herein by the Federal Aviation Administration or the U.S. Department of Transportation. Neither the Federal Aviation Administration nor the U.S. Department of Transportation shall be held liable for any improper or incorrect use of the information contained herein and assumes no responsibility for anyone’s use of the information. The Federal Aviation Administration and U.S. Department of Transportation shall not be liable for any claim for any loss, harm, or other damages arising from access to or use of data or information, including without limitation any direct, indirect, incidental, exemplary, special or consequential damages, even if advised of the possibility of such damages. The Federal Aviation Administration shall not be liable to anyone for any decision made or action taken, or not taken, in reliance on the information contained herein.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.	
A11.UAS.43			
4. Title and Subtitle FINAL REPORT		**5. Report Date** April 30, 2023	
		6. Performing Organization Code ASSURE: University of North Dakota, Embry Riddle Aeronautical University	
7. Author(s) Ryan Guthridge, PhD, David Esser, PhD		**8. Performing Organization Report No.**	
9. Performing Organization Name and Address Univ. of North Dakota 3980 Campus Rd Stop 9007 Grand Forks ND 58202-9007	Embry Riddle Aeronautical Univ. 1 Aerospace Boulevard Daytona Beach, FL 32114-3900	**10. Work Unit No.**	
		11. Contract or Grant No. 15-C-UAS	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Washington, DC 20591		**13. Type of Report and Period Covered** Final Report	
		14. Sponsoring Agency Code 5401	
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Aviation Administration.			
16. Abstract The purpose of this research is to enable safe integration of UAS in the NAS through building upon existing aviation database and data-sharing efforts encouraged and endorsed by participating government-industry entities. Through this research, a data architecture for unmanned air and ground vehicles and operations will be developed in alignment with the FAA's Aviation Safety Information Analysis and Sharing (ASIAS) program.			
17. Key Words Safety Risk Management, UAS Integration, Data Analysis, Flight Data Monitoring, ASIAS		**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.	
19. Security Classification (of this report) Unclassified	**20. Security Classification (of this page)** Unclassified	**21. No. of Pages** 130	**22. Price**

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

TABLE OF ACRONYMS

ADS-B	Automatic Dependent Surveillance-Broadcast
AGL	Above Ground Level
ACAS	AirCraft Analytical System
ASAP	Aviation Safety Action Program
ASIAS	Aviation Safety Information Analysis and Sharing
ASRS	Aviation Safety Reporting System
BLOB	Binary Large Object
CAST	Commercial Aviation Safety Team
CSV	Comma Separated Value
ERAU	Embry Riddle Aeronautical University
FAA	Federal Aviation Administration
FDM	Flight Data Monitoring
FOQA	Flight Operational Quality Assurance
GA	General Aviation
GAJSC	General Aviation Joint Steering Committee
GCS	Ground Control Station
IRB	Internal Review Board
LFL	Logical Frame Layout
MHz	Megahertz
MSL	Mean Sea Level
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NGAFID	National General Aviation Flight Information Database
sUAS	Small Unmanned Aircraft System
TCAS	Traffic Alert and Collision Avoidance System
UAS	Unmanned Aircraft Systems
UAST	Unmanned Aircraft Safety Team
UFDM	Unmanned Flight Data Monitoring
UND	University of North Dakota

EXECUTIVE SUMMARY

The overarching purpose of this research was to aggregate high quality Unmanned Aircraft System (UAS) flight data with commercial and general aviation flight data and surveillance data, in order to develop enhanced safety analyses for National Airspace System (NAS) stakeholders and to support safe integration of UAS in the NAS. Through this research, a data architecture for unmanned air vehicles and operations was developed in alignment with the Federal Aviation Administration's (FAA's) Aviation Safety Information and Sharing (ASIAS) program. Over 1,000 unique UAS flights were collected, processed, and analyzed to test the functional capabilities of a flight data monitoring system for UAS. In addition, a series of novel analytics were created to monitor the risk of collision between manned and unmanned aircraft, as well as analytics to monitor flight-critical aspects of UAS operation in the NAS.

This project designed and evaluated Flight Data Monitoring (FDM) for unmanned operations and integrated that data into the ASIAS system. This project integrated the findings from ASSURE project A20, which identified current UAS FDM capabilities and practices, including refresh/recording rates and robustness, and developed guidance for an UAS FDM standard. The research team includes original members who designed and deployed the National General Aviation Flight Information Database (NGAFID), which has successfully integrated and is data-sharing with ASIAS. As a functional outcome of this research, the team created a guide for new users of the flight data monitoring architecture and identified key research areas and capabilities that would be required before an industry-wide rollout could be achieved in an effective and scalable manner.

1 INTRODUCTION

1.1 Background

The ASIAS program connects approximately 185 data and information sources across government and industry, including voluntarily provided safety data. The ASIAS program works closely with the Commercial Aviation Safety Team (CAST), the General Aviation Joint Steering Committee (GAJSC), and the United States Helicopter Safety Team (USHST) to monitor known risk, evaluate the effectiveness of deployed mitigations, and detect emerging risk. ASIAS partners with CAST to monitor the effectiveness of deployed safety enhancements. ASIAS has established metrics that enable CAST to evaluate the effectiveness of safety mitigation efforts. In addition, ASIAS also supports the GAJSC by using de-identified General Aviation (GA) operations data to help identify risks and evaluate the effectiveness of deployed solutions. The GAJSC reaches out to the GA community through stakeholders to demonstrate the benefits of sharing safety data with ASIAS in a protected, non-punitive manner.

There are currently 47 Part 121 member air carriers, more than 150 corporate/business operators, six helicopter operators, two manufacturers, and two maintenance, repair, and overhaul organizations participating in ASIAS. The program continues to evolve and has matured to contain safety data from 99% of U.S. air carrier operations, all provided voluntarily for use by the FAA and industry. To facilitate the sharing of safety issues and best practices, ASIAS partners with the industry-sponsored Aviation Safety InfoShare meeting. This partnership enables ASIAS to provide a data-driven approach for the early identification of emerging systemic safety issues within the National Airspace System (NAS).

ASIAS retains access to a wide variety of public and proprietary data sources. Each source provides information from different parts of the NAS. Current examples include:

- ACAS (AirCraft Analytical System)
- ASAP (Aviation Safety Action Program)
- ASDE-X (Airport Surface Detection Equipment – Model X)
- ASPM (Airspace Performance Metrics)
- ASRS (Aviation Safety Reporting System)
- ATSAP (Air Traffic Safety Action Program)
- FOQA (Flight Operational Quality Assurance)
- METAR (Meteorological Aviation Report)
- MOR (Mandatory Occurrence Reports)
- NFDC (National Flight Data Center)
- NOP (National Offload Program office track data)
- SDR (Service Difficulty Reports)
- TFMS (Traffic Flow Management System)

The FAA is increasing the quantity and types of participants as part of a phased expansion plan, such as expanding participation in the corporate/business and small GA communities.

ASIAS is a unique collaboration between the FAA and the aviation community to share and analyze data. All participants in the program benefit from the proactive discovery of common systemic safety problems that span the aviation community. While participation has expanded to include all major Part 121 operators, Parts 91, 135, and UAS operators are underrepresented in the ASIAS program, if at all.

1.2 Scope

This project built upon ASSURE project A20, that identified UAS parameters, exceedances, and recording rates for UAS and move to incorporate the data into the NGAFID. The project that was completed and detailed in this report involved processing UAS safety data within ASIAS in order to develop a functional capability for flight data monitoring of UAS and conduct aggregate safety risk analysis that supports the ASIAS program. The project used a combination of information technology and outreach to a diverse assortment of stakeholders (including manufacturers, unmanned operators, and regulators), to perform the following tasks included in the original statement of work:

1.2.1 Program Management

The research team managed the effort of this project to ensure all tasks were in alignment with the statement of work. The research team coordinated with the FAA through Program Management Reviews, Technical Interchange Meetings, interim reports, emails, and telephone meetings to ensure the research validation objectives were being met.

A kickoff meeting was held to review sponsor needs, clarify any issues with scope, and to discuss the technical approach and methodology. At this meeting, the research team reviewed the Research Task Plan, which included a detailed work plan covering all tasks associated with the project and a project schedule that tracked project activities, durations, and milestones.

Through the duration of the project, the team held monthly Technical Interchange Meetings with the Program Manager and sponsors to provide detailed activity status updates, review schedule progress, and document adjustments according to sponsor inputs. During these meetings, financial reports were reviewed, highlighting financial progress and any risks or areas of concern. On a bi-annual basis, FAA Leadership Briefings were delivered to the ASSURE project teams, FAA program managers, and project sponsors.

1.2.2 Task 1: Configure Storage and Formatting Requirements of Unmanned Data

Configure storage and formatting requirements of unmanned data in the NGAFID database, or a database with the same look and underlying infrastructure.

> Task 1.1: Obtain final report from ASSURE project A20, which identified UAS parameters, exceedances, and recording rates for UAS FDM.
>
> Task 1.2: Review the final report from ASSURE project A20 to establish the requirements for storage and formatting UAS data.
>
> Task 1.3: Determine UAS flight data file formats, develop methodologies to parse textual data and unpack binary formats so they can be inserted into the NGAFID, or equivalent database.

Task 1.4: Configure storage and formatting requirements of UAS data in the NGAFID, or equivalent database.

1.2.3 Task 2: Configure and implement a prototype system to collect unmanned Flight Data Monitoring (FDM) records from industry and academic participants.

Configure and implement a prototype system to collect unmanned FDM records from industry and academic participants, preferably combined with ngafid.org, or an equivalent.

Task. 2.1: Consider existing solutions of data collection for NGAFID users to determine if they can realistically be used for UAS data collection.

Task 2.2: Determine UAS identifiers from within UAS flight data files, so particular UAS can be better tracked within the NGAFID, or equivalent database.

Task 2.3: Develop a prototype system to collect UAS FDM records.

Task 2.4: Test the feasibility of the prototype system with a control group.

Task 2.5: Implement the prototype system to collect UAS FDM records from industry and academic participants that connects with ngafid.org, or equivalent database.

1.2.4 Task 3: Collect Unmanned Flight Data Monitoring records.

Collect at least 1000 flights of Unmanned Flight Data Monitoring records. Half of the flights can be simulated (FAA Tech Center and National Aeronautics and Space Administration [NASA] offer to contribute), but representative of actual drone missions. Half of the flights must be actual flights over the U.S. in the past two years. The flights must be diverse in duration (five to 90 minutes), weight (0.4 pound to 80 pounds), and configuration (transponder-equipped and not, quad-rotor and fixed wing). Create a page on the public website to display aggregate statistics and the diversity of the flights collected.

Task 3.1: Identify personal, academic, and industry participants with operations that are diverse in duration (five to 90 minutes), weight (0.4 pound to 80 pounds), and configuration (transponder-equipped and not, quad-rotor and fixed wing).

Task 3.2: Implement infrastructure to allow participants to collect and submit UAS FDM records to ngafid.org, or equivalent database, using the prototype system developed in Task 2.

Task 3.3: Monitor and facilitate collection of at least 1,000 flights of UAS FDM records by participants.

Task 3.4: Create a page on the public website (ngafid.org or equivalent database) to display aggregate statistics and the diversity of the flights collected.

1.2.5 Task 4: Interface with unmanned communities and gather industry feedback.

Interface with unmanned communities such as Unmanned Aircraft Safety Team (UAST) through conferences and symposia to determine their biggest concerns with aviation safety risk. Evaluate industry recommendations for encouraging voluntary submission of Unmanned Flight Data Monitoring. Include prioritization by industry of specific safety risks that are best analyzed with Unmanned Flight Data Monitoring.

Task 4.1: Interface with unmanned communities (UAST and similar) through conferences and symposia.

Task 4.2: Collect and aggregate the industry's biggest concern(s) with aviation safety risk, highlighting UAS operations risk.

Task 4.3: Prioritize aviation safety risk concerns and identify which are best analyzed with UAS FDM.

Task 4.4: Evaluate the industry's recommendations for encouraging voluntary submission of UAS FDM to the ASIAS program.

1.2.6 Task 5: Measure the risk of collision between unmanned and manned aircraft.

Measure the risk of collision between unmanned and manned aircraft. The risk must be calculated using the flights collected in earlier tasks, but can be limited to one specific month, with at least four models of unmanned aircraft. Information on manned flights can come from FAA ANG-E272 at the FAA Tech Center in New Jersey, or other similar surveillance sources. Risk of collision should be calculated and modeled with proximity and closure rate, at a minimum. Measure how closely this model approximates the performance of Traffic Alert and Collision Avoidance System (TCAS), ACAS, or similar algorithms currently used in aviation.

Task 5.1: Obtain information on manned flights through surveillance sources and connect with FAA ANG-E272 (FAA Tech Center in New Jersey) to ensure validity and completeness of data sources.

Task 5.2: Integrate UAS FDM records collected in Task 4, with information from manned aircraft surveillance sources.

Task 5.3: Consider risk of collision models used in existing FDM solutions and develop a risk of collision metric for conditions between unmanned and manned aircraft. The risk of collision metric will be calculated and modeled with proximity and closure rate, at a minimum.

Task 5.4: Measure how closely the risk of collision model approximates the performance of industry-standard systems, such as TCAS, ACAS, or similar algorithms.

Task 5.5: Create visualization(s) on ngafid.org website, or equivalent database website, to display risk of collision metrics created in this task.

1.2.7 Task 6: Measure a novel risk identified through the community outreach above.

Measure a novel risk identified through the community outreach in previous tasks. The risk must be calculated using the flight data collected in this effort. The risk must be displayed on the public webpage at an aggregate level.

Task 6.1: By using the priority list of industry feedback in Task 4, choose at least one novel risk to measure using flight data.

Task 6.2: Create novel analytics to measure and display the risk event chosen in Task 6.1.

Task 6.3: Display aggregate statistics of risk analytics developed in this task on the ngafid.org webpage, or equivalent database website.

1.2.8 Task 7: Create visualizations of collision risk and battery performance.

Create visualizations of collision risk and battery performance. These visualizations must be available at an aggregate level on the website. The visualization must show locations and configurations with more than five incidents of high risk as calculated previously. The visualization must include enough unmanned data to show at least ten locations, each with more than five incidents of high risk.

> Task 7.1: Using the risk of collision metric developed in Task 5, create novel analytics to measure the risk of collision versus UAS battery performance.
>
> Task 7.2: Create visualizations of risk event location and configuration. The visualization must show at least ten locations, each with more than five incidents of high risk.
>
> Task 7.3: Display visualizations of risk analytics developed in this task on the ngafid.org webpage, or equivalent database website.

1.3 Research Framework

The research conducted as part of this project used a multi-disciplinary approach and combined quantitative and qualitative techniques. The primary focus of this project was to provide useful and usable information to be used in the analysis and incorporation of UAS FDM in the ASIAS program. Many of the tasks listed were run in parallel, however some tasks required the completion of previous tasks before they began.

The expertise of the researchers at the universities involved in this project allowed small teams of experts to complete tasks in the most efficient means possible. With the support of the lead ASSURE university, the researchers assigned to tasks met the milestones set forth in the project schedule.

The collection of flight data was conducted with available UAS flights conducted by universities, independent organizations, and industry partners. The overarching goal of the data collection process was to procure a range of aircraft and operational types.

The lead ASSURE university, the University of North Dakota (UND), led the program management tasks, as well as produced meeting minutes for all meetings throughout the duration of the project.

2 A PROTOTYPE SYSTEM FOR UNMANNED FLIGHT DATA MONITORING

2.1 Database Overview

This project utilized the existing structure of the NGAFID to process and aggregate data from UAS operators for data sharing. The NGAFID is part of ASIAS, FAA funded, joint government-industry, collaborative information sharing program that proactively analyzes broad and extensive data sources towards the advancement of safety initiatives and the discovery of vulnerabilities in the NAS. The NGAFID was originally conceived to bring voluntary FDM capabilities to general aviation, but has now expanded to include the UAS community.

The NGAFID allows UAS operators to upload, analyze, and share their flight data with the ASIAS program. Operators that are able to extract flight data from the UAS by means of the platform or ground station can upload their data for processing through the NGAFID.org website.

The NGAFID allows users to upload data files from their aircraft or ground station for automatic processing. This upload functionality is initiated by the operator through the NGAFID.org website, where processing status can be observed. After the data is uploaded by the operator, it is processed in a relational database where the operator-specific flight parameters are mapped to logical parameters that can be compared across platforms and UAS type. A simplified diagram of the data flow can be found in Figure 1.

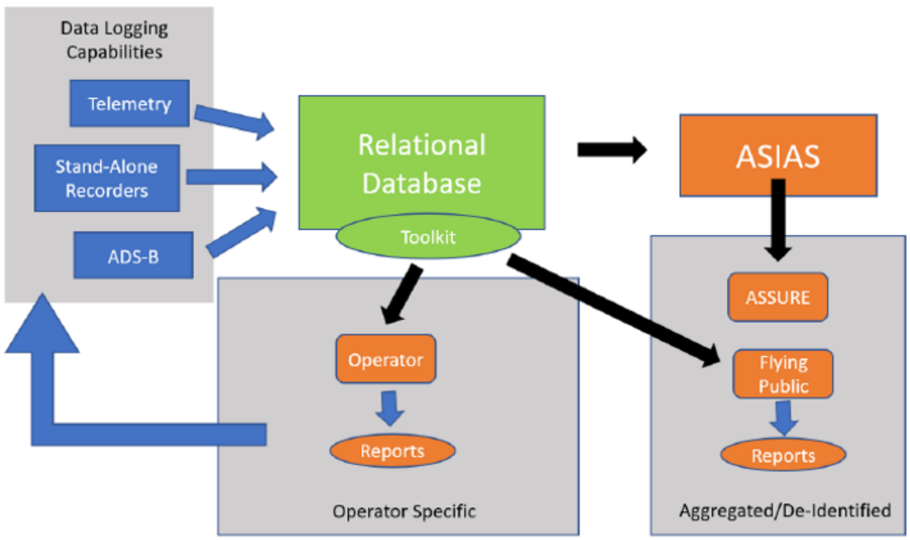


Figure 1. Database Structure Diagram.

Once the flight data is mapped to logical parameters, it is processed to identify events significant to the safety of UAS operations. Specific details of the flight, including events triggered, can be observed through the NGAFID.org website. UAS operators will receive detailed information on their flights, which can be exported in Comma Separated Value (CSV) form or viewed graphically through the website. Detailed information available to users includes a set of recorded parameters for the duration of the flight (Figure 2), as well as a depiction of their flight track (Figure 3). Figures 2 and 3 were obtained from a sample Insitu Scan Eagle UAS flight. Additionally, processed data from the relational database is available for use in ASIAS in a deidentified fashion. From this, the ASIAS team can export and integrate UAS data for internal analysis and external communication to the flying public.

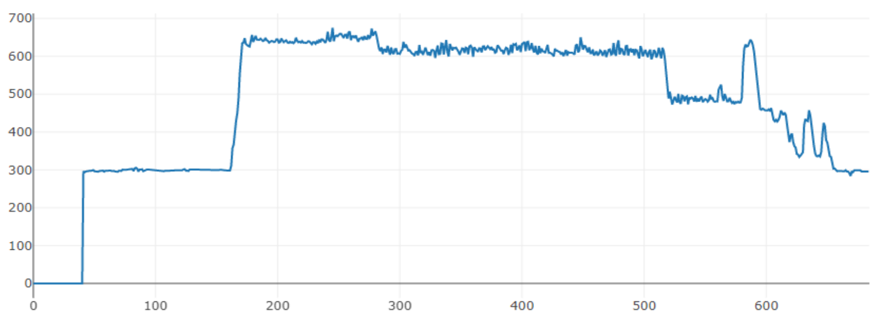


Figure 2. GPS Altitude Plot from Insitu Scan Eagle.

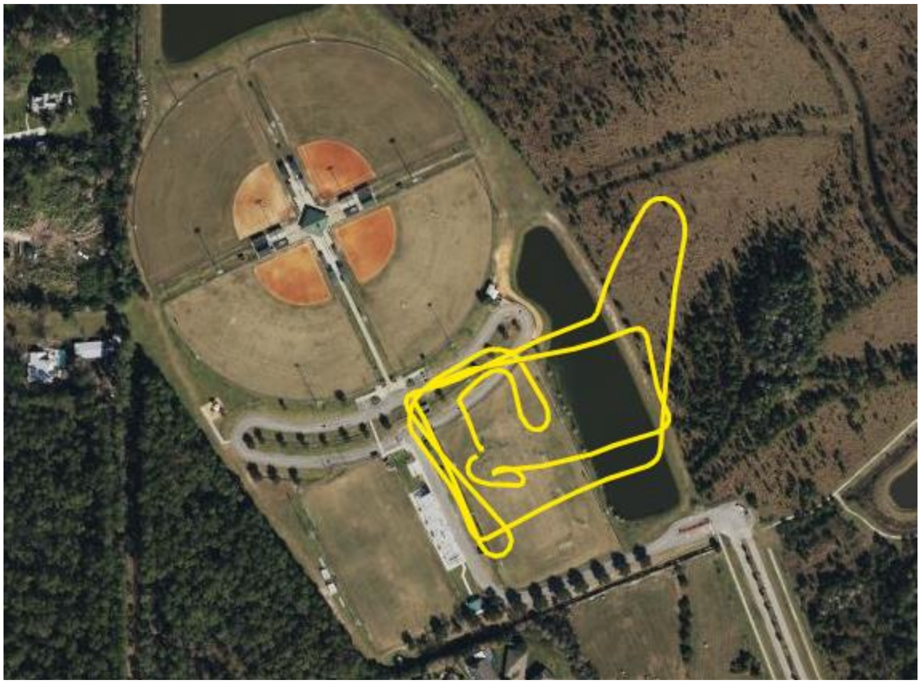


Figure 3.GPS Position Plot from DJI M600.

All data collected from the UAS system is anonymous and deidentified so pilots can voluntarily share their data without any fear of reporting or reprisal. UAS operators will be able to view data from their own aircraft fleets, as well as comparing their own performance against deidentified aircraft from other UAS operators that have data in the system. This allows operators to see an overview of their fleet, understand the number of events by UAS platform type, and compare their own safety trends to other UAS fleets with similar operating characteristics. Examples of this can be found in Figure 4, Figure 5, and Figure 6. These examples are inclusive of a UAS operator that uploads data regularly to the NGAFID. An operator of a single UAS may see substantially fewer flights and events, however the display of this information will be identical.

Figure 4. Sample NGAFID.org Fleet Dashboard for a UAS Operator.

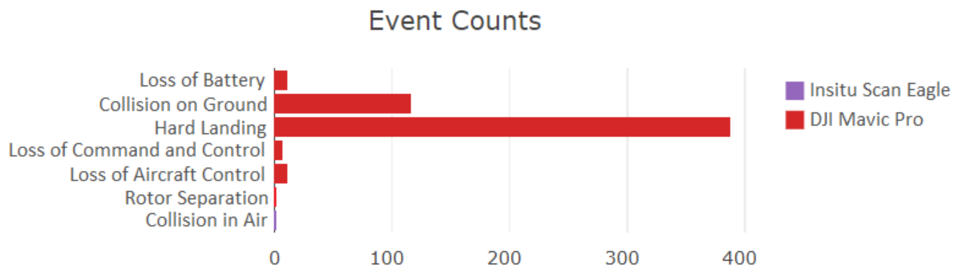


Figure 5. Sample NGAFID.org Events by Aircraft Type.

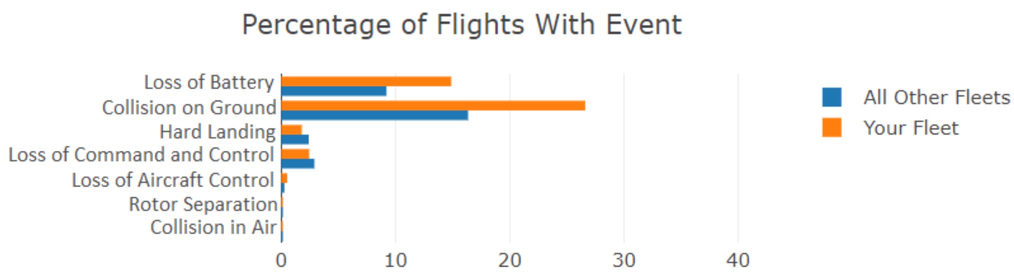


Figure 6. NGAFID.org Event Comparison (Operator's Fleet versus All Other Fleets).

2.2 Functionality and Configuration of the Database for UAS FDM

The schema in Figure 7 gives a somewhat simplified representation of data processing in the NGAFID. One important distinction between the NGAFID and a traditional spreadsheet data import structure, is that each column of data from a flight file is stored as an entry in either a *StringSeries* table for textual data, or a *DoubleSeries* table for numerical data. These entries contain the name of the column, along with its given data type from the CSV file (e.g., feet, temperature, mph, date, etc.), and for the numerical data the minimum, average and maximum values which are calculated during the initial data ingestion step. The actual column values are stored as an array of bytes in a Binary Large Object (BLOB), which allows for storing these data in a compressed format providing a significant reduction in storage as well as faster data ingestion. One significant benefit of this structure is that there are no hardcoded or expected columns for a CSV file, which allows the NGAFID to identify and ingest data from various aircraft manufacturers and data suppliers that may provide data in unique and proprietary data formats.

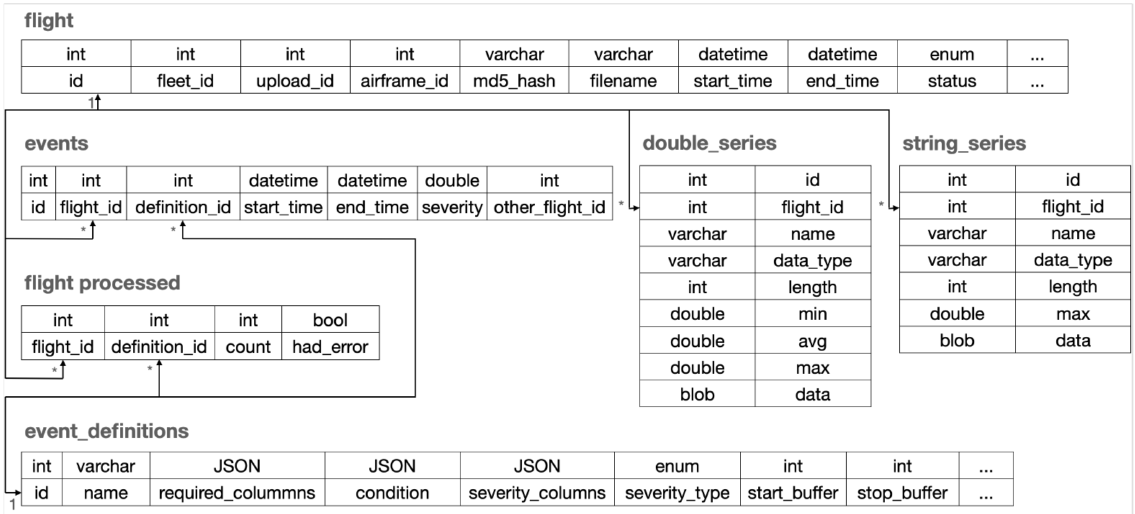


Figure 7. Schema for Data Processing in the NGAFID.

Documentation and installation instructions for the NGAFID are kept up to date on the project's Github webpage:

https://github.com/travisdesell/ngafid2.0

This includes instructions for setting up and configuring the database and webserver, along with a list of dependencies and additional downloads of related data used to determine altitude above ground level from GPS coordinates, and to determine distances from airports and runways. Additionally, a full up-to-date version of the database schema can be found at the following webpage:

https://github.com/travisdesell/ngafid2.0/blob/main/db/create_tables.php

This database schema will create all database tables used by the NGAFID. The project also generates javadocs for all the webserver software developed using Java Spark.

The current design of the NGAFID has been in development for over three years, and was done in a manner to support generic imports of any kind of time-series flight data. Due to this, the majority of the effort to get the NGAFID to support UAS FDM data was in working with specific Manufacturers to first provide software and the ability to generate CSV files readable by the NGAFID, as some UAS platforms record data in a proprietary binary format not readable by third parties. For example, the Insitu Scan Eagle provides data in a binary format that must first be processed by the manufacturer’s ground control software. This processing step can be prone to errors, as the user must specify the export parameters and sample rates prior to unpacking the binary into a readable format. Other manufacturers, such as DJI, had available code that could be incorporated into the NGAFID software in order to automatically unpack the binary files for processing. In this case, the end user did not have to manually process the data before submitting the files to the NGAFID. In all cases, the NGAFID conducts a data validation step that alerts the user of data files that exceed a 1Hz sample rate. No interpolation is conducted on data that exceeds 1 Hz, however the user is aware of the risk and can then choose to reupload a new data file, if a more robust data file exists for that flight.

After the data configuration step, the NGAFID software needed to be updated to properly relate the different sensor names from the UAS platform to the names used to conduct various NGAFID calculations (e.g., DID_GPS_LON to LONGITUDE, DID_GPS_ALT to AltAGL, etc). This was done in a straightforward way to support additional data with differing names or formats.

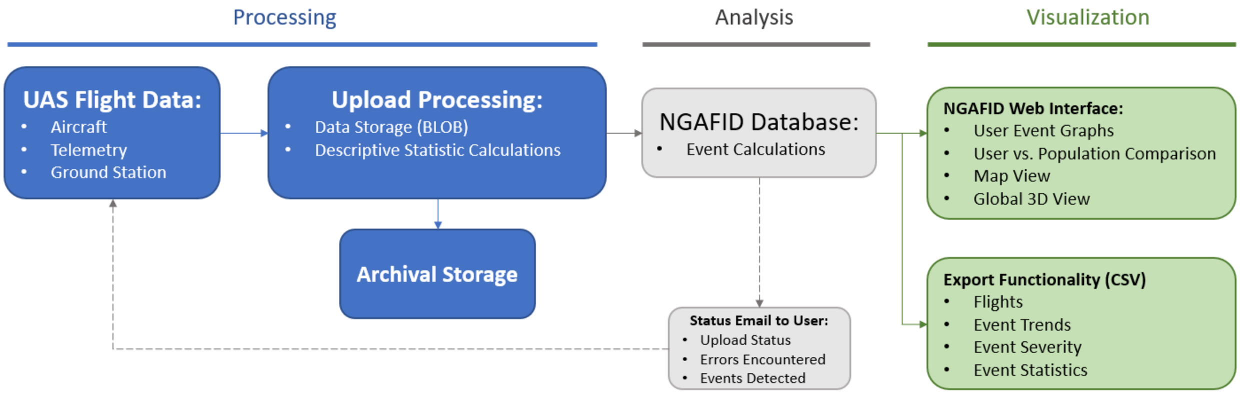


Figure 8. NGAFID Data Flow Diagram.

Figure 8 presents the general pipeline for data ingestion into the NGAFID. It is set up in a generic way that is equally applicable to manned aircraft (rotorcraft and fixed-wing) and unmanned data. Similarly, for unmanned data, the setup will apply to both fixed-wing and multi-rotor UAS. The presented schema operates identically for manned and unmanned data, with the only major difference being names of time series columns in recorders with differing platforms.

In contrast, the original NGAFID utilized a single massive database table with a column for every possible data recorder parameter that could be entered into the database. Not only was this slow and bloated, but it also prevented easy extension as new aircraft were added.

It would be unfeasible to enforce one data standard, given the rapidly changing UAS landscape. Given this, mapping and standardization is done at the events and definitions level. Admins (and later fleet administrators) can create airframe specific events and compare fleets (Figure 9).

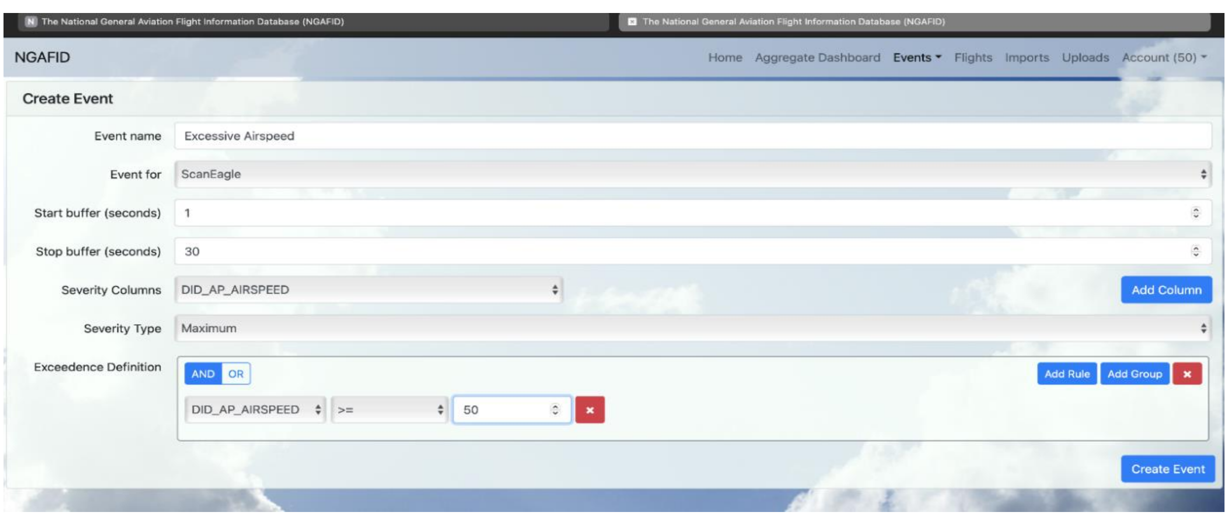


Figure 9. NGAFID Event Creation Dialogue.

The events are stored in the database (in the event_definitions) table in a generic JSON format. Figure 10 shows an example representation for Engine Shutdown below 3000 Ft for four different airframe types.

```
+----+-------------+--------------------------------+---------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
------------------------------------------------------+

| id | airframe_id | name                           |
condition_json
|

+----+-------------+--------------------------------+---------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
------------------------------------------------------+

| 46 |           2 | Engine Shutdown Below 3000 Ft |
{"type":"GROUP","condition":"AND","filters":[{"type":"RULE","inputs":["E1
RPM","<","100"]},{"type":"RULE","inputs":["AltAGL",">","500"]},{"type":"RULE","inputs":["AltAGL",
"<:["AltAGL","<","3000"]}]}
|

| 47 |           1 | Engine Shutdown Below 3000 Ft |
{"type":"GROUP","condition":"AND","filters":[{"type":"RULE","inputs":["E1
RPM","<","100"]},{"type":"RULE","inputs":["AltAGL",">","500"]},{"type":"RULE","inputs":["AltAGL",
"<","3000"]}]}
|

| 48 |           4 | Engine Shutdown Below 3000 Ft |
{"type":"GROUP","condition":"AND","filters":[{"type":"RULE","inputs":["E1
RPM","<","100"]},{"type":"RULE","inputs":["AltAGL",">","500"]},{"type":"RULE","inputs":["AltAGL",
"<","3000"]}]}
|

| 49 |           3 | Engine Shutdown Below 3000 Ft |
{"type":"GROUP","condition":"AND","filters":[{"type":"RULE","inputs":["AltAGL",">","500"]},{"type
":"RULE","inputs":["AltAGL","<","3000"]},{"type":"GROUP","condition":"OR","ftion":"OR","filters":
[{"type":"RULE","inputs":["E1 RPM","<","100"]},{"type":"RULE","inputs":["E2 RPM","<","100"]}]}]}
|

+----+-------------+--------------------------------+---------------------------------------------
--------------------------------------------------------------------------------------------------
--------------------------------------------------------------------------------------------------
------------------------------------------------------+
```

Figure 10. Example Event Definition for Engine Shutdown below 3000 feet.

With the information shown in Figure 10, researchers can, for example, calculate events/exceedances for different aircraft, of the same event/exceedance type, but with different data recorder and airframe specific parameters (the above shows one airframe having two engines vs. the others having one, as an example). There will similarly be the same concern with UAS, as there may be different definitions for high altitude, excessive speed or any of many other events/exceedances dependent on the platform (i.e., multi-rotor, fixed wing, gyrocopter, etc.). With predetermined calculated events, fleet types (or specific platform types) can be compared to each other by the number of flagged events of whichever types have been specified.

At this point the NGAFID framework is ready to support UAS data (and was designed so it could be easily extended to do so), however platform specific event/exceedance definitions need to be developed so the event definitions for those UAS platform types can be put in the database, such that the event calculation daemon (above) can go through the UAS data in the NGAFID and calculate them. The event calculation daemon has been developed such that whenever any new event definition is created it will calculate those for all available data of the airframe type for that event definition. This is another reason storing the UAS (and other) data in the NGAFID, as close to what comes from the data recorders as possible, is valuable, so new events can be added as needed without having to do re-imports of data. This is a significant improvement over the old version of the NGAFID, as data ingestion and event calculation were not separated out into different processes, due to data being stored in a single database table.

2.3 Exceedances and Parameter Ranking

ASIAS project A20: UAS Parameters, Exceedances, Recording Rates for ASIAS, lists the suggested requirements for data to be collected for basic FDM analysis. From the 2020 report:

> "One of the most fundamental data analysis tasks conducted within an FDM program is the calculation and recording of an exceedance. An exceedance is an event or occurrence wherein an aircraft was operated outside of a predetermined range. For example, if the maximum RPM for a UAS rotor is set at 2,450 RPM, and during a flight a UAS rotor RPM reaches 2,451 RPM, an exceedance will have occurred. By tracking exceedances, an analyst can examine the event on a singular basis to prospect for accident precursors, or the number of exceedances fleetwide can be recorded and analyzed for trends. Based upon the availability of flight data, there are many different types of exceedances that can be recorded.
>
> Appendix A lists some exceedances that can be used for UAS platforms. This list contains events that are relevant to UAS operations and is partially derived on events created for the manned Cessna 172 fleet. The first column is the event to be measured, the second column is the phase of flight the event will be measured in. The third and fourth columns are event values that would "trigger" an exceedance (although it is important to note these can vary based upon the UAS platform). Level 1 values are not as severe as a level 2 value. The fifth column contains notes for UAS operations. As noted earlier, these events were built from an event set for the Cessna 172. As this is the case, some of the events can be used for UAS operations with some modification, while others will probably apply as is.
>
> All data and recorded parameters could be considered valuable depending upon the exceedance being measured. There are, however, different tranches of data that contain useful information. Some data can be used for basic data analysis while other data types are needed for more comprehensive analysis. At a minimum, GPS position (latitude, longitude and altitude) and time are necessary for all analysis. Accordingly, these parameters can be considered to be of the highest ranking and required. From these basic parameters, other metrics can be derived, such as direction, groundspeed and altitude of terrain. Another important data group is onboard telemetry metrics. Specifically, battery life, status of communication link, and some engine performance parameters would also be highly beneficial to an FDM analyst."

Largely, the current research team believes this analysis is sound and provides a quality description of the fundamental parameters required for a functional UAS FDM system. To simplify and consolidate the analysis, the following parameters are suggested to be included in a UAS data stream, at a minimum:

- GPS Position (Latitude, Longitude, Altitude),
- Time,
- Battery Life,
- Communication Link Status,
- Engine Performance Information.

It is important to note that additional parameters will provide a more robust safety analysis program. The current lack of a data standard is listed in the limitations and considerations section of this report. Appendix B was replicated from the ASSURE A20 Final Report and includes a data dictionary for a common fixed-wing platform. This is provided as an example of the types of data that are commonly available, along with the data format and units for recording. Finally, Appendix C of this report lists examples of the source data files of common small Unmanned Aircraft System (sUAS) platforms. These show the richness of the available data, but also highlight the lack of standardization of the UAS manufacturers and recorders that are available today.

3 COLLECTING AND PROCESSING UNMANNED FLIGHT DATA

3.1 Flight Data Recording and Retrieval

As FDM systems are developed, the first step revolves around the availability and retrieval capabilities of flight data. Flight data can be generated by various types of equipment. Larger unmanned aircraft may have standalone, dedicated flight recorders, but smaller UASs will not. In lieu of flight recording devices, many UASs do record various flight parametrics, which are generally accessible via download from an onboard stored memory device or Ground Control Station (GCS). These data are known as telemetry, and so long as a digital recording of that telemetry is made and is available for retrieval, those data can be used for Unmanned Flight Data Monitoring (UFDM).

The smallest UASs rely on positive control by a human pilot who maintains constant line-of-sight contact with the platform. Many of these recreational devices may not return telemetry as the pilot is responsible for flight. However, there are some emerging telemetry capabilities among smaller UASs. Additionally, as has been found with manned aircraft, even a sparse telemetry stream could prove useful for an UFDM solution. At a minimum, a timestamp (with GPS trilateration correction and verification), geolocation (with preference to latitude/longitude, or the capability to derive such points) and altitude (above Mean Sea Level [MSL] and Above Ground Level [AGL] or derived AGL) would be able to provide useful information to a UFDM system.

sUAS have varied capabilities for data recording, retrieval, and analysis. Such data collection methods, formats, sample rates, and data points do not generally follow an established standard, and vary widely based on the sUAS manufacturer. Small UAS platforms manufactured by the DJI Technology Company, based in Shenzhen, China, represent approximately 72% of the market share of consumer sUAS products (Skylogic Research, 2017). DJI platforms have several modalities of data collection that may provide safety-centric information. For DJI devices, flight data may be derived from one of three primary storage locations: (a) the tablet or phone connected to the remote controller and used to run the DJI GO Application (which serves as the user interface display during flight); (b) the external SD card storing geotagged image and video data; and (c) the embedded SD card attached to the flight controller board.

The highest fidelity flight data for DJI products is derived from the internal flight controller SD card, which records a vast array of flight telemetry, system status, and time-indexed data signals in (.DAT) format. In total the system tracks 19 aircraft status parametrics and 172 time-series signals (see Appendix C).

Parrot UAS platforms use a different data architecture and access methodology for retrieving flight data logs. For these platforms, users must connect their UAS to Parrot's SenseFly eMotion 3 application. This not only manages flight logs, but also performs mission planning, post flight processing, and other functionality for Parrot products. Using eMotion's data flight manager, the user can download a flight log file in SenseFly's proprietary (.BBX) format. Additionally, flight data can also be extracted from collected geotagged imagery (EXIF), geolocation information (.KML), and tabular flight log information in (.CSV) formats. These file types can be opened and analyzed using a wide variety of software solutions.

Yuneec platforms log telemetry in tabular format to a microSD card downloadable from the controller via a USB PC connection. Graphical display of flight logs can be viewed in the DataPilot proprietary application (Yuneec, n.d.). Alternatively, flight logs can be opened in .CSV format, using other tabular display programs. Yuneec platforms record 86 parameters, including telemetry, navigational sensor inputs, platform status, faults, flight controller modes, and other miscellaneous remote information (Elsner, 2017).

In the ASSURE A20 project, researchers were able to obtain information from common command-and-control sources, such as the Piccolo Command Center Ground Station. They noted, "while the available data and recording rates may vary by platform type, such telemetry information may be useful in performing detailed flight analysis. For this study, Piccolo telemetry data was obtained on MASC Tigershark Block 3 and Griffon Aerospace Outlaw G2E platforms. An assessment of the Piccolo data architecture indicated the system records 201 parametrics at a rate of 1 Hz." (ASSURE, 2020). In this research project, the scope was to develop a data processing capability and test the functionality on a representative set of UAS. A handful of platforms were able to be collected at scale and provided variance in the data file types, operational diversity, and locationality, which was appropriate for vetting the data processing capability of the NGAFID for processing UAS data.

3.2 Technical Diversity of Data Collected

This project required the research team to collect at least 1000 flights of Unmanned Flight Data Monitoring records. Half of the flights must be actual flights over the U.S. in the past two years. The flights must be diverse in duration (five to 90 minutes), weight (0.4 pound to 80 pounds), and configuration (transponder-equipped and not, quad-rotor and fixed wing). The research team was able to collect this dataset, of which demographics can be found in Table 1.

As seen in the data examples in Appendix C, each manufacturer and platform records a unique set of parameters, rates, and exceedances within the raw data files. Much like airline and fixed-wing platforms in early FDM days of the 1990s, we must explore which data parameters are required for a robust FDM program. Also, researchers in this field must explore a possible global format to facilitate data capture, such as the ARINC 717/767 standard. As such, it is important through this step in research to not only determine what data is available by common systems, but also what data is necessary to facilitate a FDM program for the UAS industry.

Through the flights collected, data recorders proved to be robust enough to calculate a set of exceedances unique to UAS. Follow-on tasks addressed which events were created as part of this research project, however the project team has already created events to monitor proximity of UAS to manned aircraft, as well as point-in-time exceedances of battery capacity and altitude exceedances. Furthermore, this report conducts an in-depth analysis of common FDM events that could be monitored with the collected data. Also included are recommendations for additional research to finalize a complete FDM program that compares UAS events to common manned aircraft events, as part of the ASIAS program.

Table 1. Demographics of 1,000 collected flights.

	Number of Flights	**Operator**	**Type**	**Weight (lbs)**	**Motors**
Insitu Scan Eagle	175	University	fixed wing	39.7-55	1
DJI Matrice 600	32	University	multirotor	33.22	6
DJI Matrice 210	83	University	multirotor	13.51	4
DJI Inspire 2	107	University	multirotor	9.37	4
Parrot Anafi	613	Industry Partner	multirotor	0.71	4

	Flight Duration (hh:mm:ss)		
	min	**avg**	**max**
Insitu Scan Eagle	0:27:09	2:32:01	6:41:12
DJI Matrice 600	0:04:15	0:22:47	1:04:11
DJI Matrice 210	0:03:54	0:29:06	0:43:18
DJI Inspire 2	0:08:17	0:42:35	1:12:53
Parrot Anafi	0:02:01	0:25:19	0:39:04

	Altitude (ft AGL)

	min	avg	max
Insitu Scan Eagle	0.00	347.77	1164.45
DJI Matrice 600	0.00	103.91	213.04
DJI Matrice 210	0.00	43.27	179.91
DJI Inspire 2	0.00	68.12	187.55
Parrot Anafi	0.00	38.87	234.72

3.3 Comparison with Large UAS

It is noted that the data collected by the research team does not include an aircraft between 55 to 80 pounds. The team took this into careful consideration and elected to mitigate this risk with dialogue, explanation, and a detailed analysis of large UAS capabilities.

One research gap that was found is that of data diversity between small UAS and larger UAS platforms. The challenge is related to the availability of large UAS datasets, which often pertain to classified operations and are not available to the public for observation or consumption. That being said, the researchers can relate large UAS to manned aircraft when it comes to comparing the data recording capabilities of those aircraft. Keep in mind, the diversity of aircraft and the diversity of data recording capabilities are two distinctly different topics of conversation. As shown in Appendix C, small UAS can have a quite robust data recording capability, that compares to many larger UAS and manned aircraft platforms. Additionally, large UAS may have a comparably less robust data recording capability, which does not have the same functionality as a smaller UAS. It is for this reason that the aircraft size must be disassociated with the functional requirements of the data recorder capability.

Larger UAS platforms such as General Atomics' Predator or Northrop Grumman's Global Hawk already generate large quantities of flight data and transmit such data via telemetry to storage devices in the GCS. It should be noted that while the authors of this report have received Global Hawk data from Northrop Grumman, these data are unable to be shared either in this report or publicly as it has been determined to be Controlled Unclassified Information. Although these data cannot be shared in this report, several characteristics of those data can be analyzed and described.

By any measure, Global Hawk telemetry data is very robust. In contrast to manned aircraft, including modern transport category aircraft, the Global Hawk data far exceeds what is normally found in Flight Data Recorder or Quick Access Recorder files. Most onboard sensors appear to send data through the telemetry stream. The refresh rates also appear to occur at levels exceeding 10 Hz (and in most cases much higher). The overall size of the data is also quite large, and in its raw and unfiltered format can be larger than several hundred Gigabytes per flight.

The research team has accommodated data recorder outputs of this size before, however all have been seen on manned aircraft. The unique research gap with data accessibility for large UAS can be addressed with the functional capabilities of the NGAFID database, as it pertains to the large datasets of manned aircraft. As mentioned previously, disassociating aircraft type/size with data recording capabilities is a necessary step when creating a functional capability for FDM. When large UAS datasets become available to the NGAFID project team, the functionality exists to process the data appropriately, as data of this fidelity already exist in the database. For example, as of March 2022, the NGAFID has successfully processed and managed over 2 million flights from manned aircraft, with data recorders capable of storing over 70 parameters per second, at a recording rate commonly exceeding 1 Hz.

3.4 Operational Diversity of Data Collected

The research team identified personal, academic, and industry participants with operations that were diverse in duration, weight, and configuration. As a result, UAS data were collected from academic UAS operators at the University of North Dakota (UND) and Embry Riddle Aeronautical University (ERAU), along with industry participants conducting outreach activities. A comprehensive sample of small UAS and medium UAS were collected from these operators and will be described in the respective sections.

3.4.1 The University of North Dakota

UND Aerospace's Unmanned Aircraft Systems Operations curriculum provides the breadth and depth of education needed to ensure graduates are prepared to work as effective operational team members of UAS, while fully understanding the operational and safety environments of the National Airspace System.

The UAS program offers two career tracks within the UAS Operations major, to specialize in a) advanced high-altitude, autonomous commercial systems, or b) the emerging and dynamic small UAS market. Non-aviation majors also have the opportunity to obtain a minor in UAS to use in their particular discipline.

In the advanced, high-altitude, autonomous commercial systems specialization, students train and fly with the Insitu ScanEagle. Operations with this aircraft are predominantly conducted at the Mayville, ND, airport (D56) at flight altitudes from 2,000 feet MSL and below.

In the small UAS program, students gain experience flying various models of DJI UAS, which are largely conducted in and around campus. Operations with these UAS are conducted at altitudes 200 feet AGL and below.

3.4.2 Embry Riddle Aeronautical University

The Aeronautical Science Department’s UAS research and service learning efforts include various remote sensing activities. These include capturing RGB images that are processed using photogrammetric techniques, thermal remote sensing and applying LiDAR. The research activities include archaeology support, cultural heritage preservation and environmental monitoring. Environmental monitoring includes marine life, sea turtles, Atlantic Right Whales, oyster reefs, and Florida’s saltwater lagoon systems. Partners include the Brevard Zoo, NOAA, Northrop Grumman, University of New Mexico, Cultural Heritage with Borders, and the U.S. Embassy Kosovo.

3.4.3 Industry Partners

To date, one industry partner has contributed data to this project. This operator conducts Science, Technology, Engineering, and Math (STEM) outreach programs with elementary, middle, and high school students. This program uses Parrot Anafi UAS, while conducting most of their operations at an altitude of approximately 40 feet AGL for an average duration of 25 minutes.

3.5 Data Upload Overview

The NGAFID allows UAS operators to upload, analyze, and share their flight data with the ASIAS program. Operators that are able to extract flight data from the UAS can upload their data for processing through the NGAFID.org website.

The NGAFID allows users to upload data files from their aircraft or ground station for automatic processing. This upload functionality is initiated by the operator through the NGAFID.org website, where processing status can be observed. After the data is uploaded by the operator, it is processed in a relational database where the operator-specific flight parameters are mapped to logical parameters that can be compared across platforms and UAS type. A simplified diagram of the data flow can be found in Figure 8. The section titled "Processing" is where the prototype data upload functionality exists, which allows users to send their UAS flight data to the NGAFID for processing.

3.6 Configuration and Functionality

3.6.1 Operator Data Collection

In order to use the data upload functionality of the NGAFID, an operator must first collect flight data from their UAS. Depending on the manufacturer and style of the UAS, data may be collected from a number of locations, which may commonly include the UAS itself, the telemetry stream, or the UAS ground station. The NGAFID is configured in such a way that multiple file types are supported, freeing the operator from concerns about the source of the data retrieval. It is important to note that different locations may record the data in different formats, which could potentially restrict the quantity or quality of the data analytics. In its current state, the NGAFID supports the following platforms and data types, as shown in Figure 11. As development continues as part of this project, additional platform and data types will be supported.

Manufacturer	Platform	Data Type
Insitu	Scan Eagle	CSV (converted from binary)
DJI	Ground Station	CSV
	DJI Go Application	DAT
	DJI Fly Application	DAT
	Inspire 1	DAT
	Inspire 2	DAT
	Matrice M100	DAT
	Matrice M600	DAT
	Mavic Pro	DAT
	Phantom 3	DAT
	Phantom 4	DAT
	Phantom 4 Pro	DAT

Figure 11. UAS Platform and Data Types available to Import in the NGAFID (current as of July 2023).

3.6.2 Uploading Data to the NGAFID

Once the flight data is collected from the UAS, the operator will place their flight data into a zip file prior to uploading to the NGAFID. A detailed User's Guide is included in Appendix D, which describes how to create a user account and upload data for processing in the NGAFID. After the data is successfully uploaded to the NGAFID, it is automatically placed in archival storage and marked as ready for processing. The archival storage is regularly backed up for long-term preservation, which allows the database to rebuild in the event of catastrophic failure.

3.6.3 Data Processing and Storage

When an upload is marked ready for processing, the zip file is read into memory and flight files are individually extracted. The schema for data processing, shown in Figure 7, gives a somewhat simplified representation of the data ingestion and processing steps in the NGAFID. This process step is where the data is stored in a BLOB and descriptive statistics are calculated for event exceedance calculations.

One important distinction between the NGAFID and a traditional spreadsheet data import structure, is that each column of data from a flight file is stored as an entry in either a *StringSeries* table for textual data, or a *DoubleSeries* table for numerical data. These entries contain the name of the column, along with its given data type from the CSV file (e.g., feet, temperature, mph, date, etc.), and for the numerical data the minimum, average and maximum values which are calculated during the initial data ingestion step. The actual column values are stored as an array of bytes in a BLOB, which allows for storing this data in a compressed format providing a significant reduction in storage as well as faster data ingestion. One significant benefit of this structure is that there are no hardcoded or expected columns for a CSV file, which allows the NGAFID to identify and ingest data from various aircraft manufacturers and data suppliers that may provide data in unique and proprietary data formats.

3.6.4 Calculating Events and Notifying the User

After the data is stored, processing continues on the flight data, to include a number of pre-created analyses that are run on the flights to detect any exceedances. If an exceedance is calculated, the severity is recorded and logged with the flight record. This will show the user which events were exceeded, along with how severe the exceedance was during the occurrence.

Once the processing steps are complete, an email is sent back to the user to indicate the status of their data upload. This email will include the number of flights uploaded, a description of any processing errors, and a list of event exceedances that were calculated. An example email is included in Figure 12.

```
importing 2 flight files to the database took 0.058
    0 imported with no issues.
    0 imported with warnings.
    2 had errors and could not be imported.

calculating flight events for 0 took 39.647s
    0 events were found.
    0 events could not be calculated due to data issues.

calculating proximity events for 0 took 0.983s, averaging Infinitys per flight, time bound matching averaged Infinitys and location bound matching took Infinity on average.
    0 proximity events were found.
    0 flights could not be processed for proximity due to data issues.

calculating turn to final information for 0 took 0.0s     0 flights could not be processed for turn-to-final due to data issues.

flight details:
    flight 0 imported with errors:
        Unknown file type contained in zip file (flight logs should be .csv files).
    flight 0 imported with errors:
        Unknown file type contained in zip file (flight logs should be .csv files).
```

Figure 12. Sample Upload Status Email to NGAFID User.

4 GATHERING INDUSTRY FEEDBACK

4.1 Survey

To understand how FDM could be used to monitor safety concerns of UAS, a survey was designed to gather feedback of UAS industry professionals on their biggest concerns with aviation safety risk of UAS integration into the NAS. The survey asked demographic questions to help categorize the results, then proceeded to ask the respondents to prioritize specific safety risks that are best analyzed with Unmanned Flight Data Monitoring. Finally, the survey asked respondents to list any additional safety risks that were not covered by the survey and asked for recommendations that might enhance voluntary submission of UAS FDM records to the ASIAS program.

The survey was jointly developed by project team members and submitted for Internal Review Board (IRB) review at the University of North Dakota. The survey protocol was approved by IRB #0005080 and subsequently cleared by the IRB office at Embry-Riddle Aeronautical University for dissemination through both lead universities on the project.

Qualtrics was used to create and manage survey responses. This tool was chosen for its flexibility and approved use for researchers in the project team. Respondents were "recruited" through an email request to take the survey, which was disseminated to industry groups, professional organizations, industry partners, and students within collegiate UAS programs. Records will be stored in Qualtrics, behind duo-authentication for the duration of the project. No personally identifying information was obtained through the survey. Per IRB protocol, survey data will be stored for at least 3 years after the study has ended and all data have been analyzed.

4.2 Participants and Recruitment Methods

Participants were selected from professional industry groups that have membership in UAS Organizations and/or have recently enrolled in symposia for UAS conferences. As such, the survey was submitted to a list of established industry professionals, UAS operators, business managers, and students within collegiate UAS training programs. A total of 119 UAS Operators, Business Managers, and Students in collegiate UAS programs were offered an opportunity to complete the survey, of which 31 participants responded, yielding a response rate of 26.1%. Demographics of the participants can be found in Table 2.

Table 2. Survey Respondent Demographics.

	n	Average Years of UAS Experience
Business Manager	6	5.3
Data Analyst	4	4.9
Student in a UAS Program	8	1.8
UAS Operator / Pilot	12	5.0
UAS Regulator (FAA)	1	4.0

4.3 Prioritization of Risk

Through the survey, participants were asked to prioritize the risk of UAS integration into the National Airspace System. Options were provided as part of the prioritization and rank ordering questions of the survey. Additionally, open ended questions were provided to capture any risks not specifically identified in the quantitative part of the survey. Results from each of the questions, along with dialogue, follow in this section.

When developing risk options for survey questions, prior research in the field was leveraged. In the Phase I Final Deliverable (ASSURE A20), specific UAS events were proposed through an accident-risk category identification and selection technique. Knowledge of the research team and industry professionals was used to develop a list of seven event categories, which were published in Section 9 of the report, titled "Applying ASIAS Methodology." These categories and their associated definitions are: Loss of Battery (LOB) – a loss of power due to a complete dissipation or failure of the onboard battery or batteries; Loss of Command and Control (LOC2) – a loss of signal or transmission from an associated GCS; Rotor Separation – an occurrence where one or more rotor blades physically separate from their UAS motor mounts; Loss of Control (LOC) – an inflight condition wherein the UAS becomes uncontrollable; Hard Landing – a harder than expected contact between the UAS and surface during a landing maneuver; Collision on Ground – a collision between the UAS and an object or person on the ground; and Collison in Air – a collision between the UAS and another object while airborne.

4.3.1 Q1: What is the biggest safety risk of UAS integration into the National Airspace System?

Survey participants were asked to choose their biggest safety risk from generalized categories, then rank-order the seven categories based on their perceived risk of UAS integration into the National Airspace System. Figure 13 shows the results of the survey question, “What is the biggest safety risk of UAS integration into the National Airspace System?” Five options were provided, along with an option of “other,” by which the participant would type the risk into a text box. This last option was not used by any of the 31 participants in the survey. Number of respondents is listed on the y-axis.

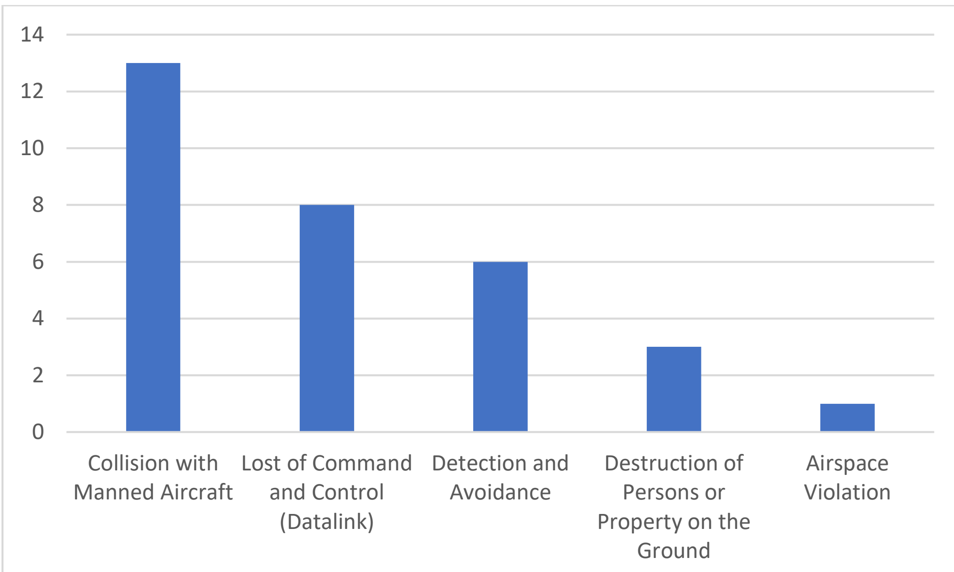


Figure 13. Biggest Safety Risk of UAS Integration into the National Airspace System.

4.3.2 Q2: Rank the following concerns in order of safety risk for UAS.

The second question asked participants to rank the seven risk categories into an order based on safety risk for UAS. This question was asked slightly differently from question one, in that this question pertains to the specific risks to UAS, rather than risk of integrating UAS into the NAS. In any case, the results of this question align with question one, with collision in the air being the biggest risk category. Subsequently, Loss of Control aligns as the second highest risk, with the other risk categories resulting in significantly lower prioritization by the survey participants. Results of the survey question can be found in Figure 14. A sum of the rank-order weighting is listed on the y-axis (highest risk = 7, Lowest risk = 1).

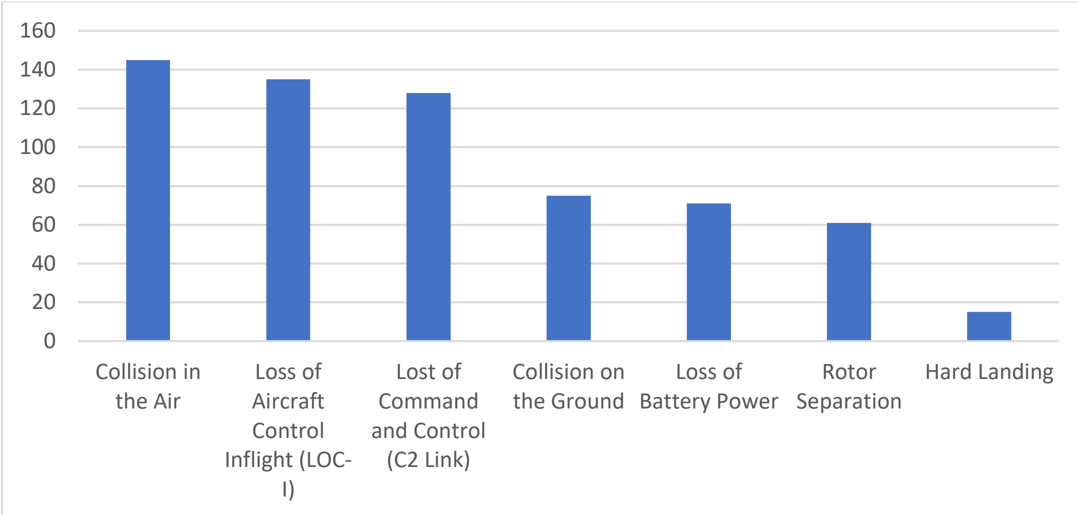


Figure 14. Rank Order of Safety Risks for UAS.

4.3.3 Q3: Are there any other safety risks associated with UAS integration into the National Airspace System that haven't been addressed in this survey?

After the participants were asked what was the biggest safety risk of UAS integration into the National Airspace System, the survey prompted the participants to describe what safety risks were not covered by the initial question. By allowing the participants to qualitatively explain their thoughts, the survey gathered more context on the perceptions of safety risk by the participants in the survey. Below are the top three additional risks that were identified by the participants in the survey. The full results of this survey question can be found in Appendix E.

4.3.3.1 Requirements for Identification (Remote ID and ADS-B)

The most commonly mentioned risk was the concept of aircraft identification. While both manned aircraft and UAS have remote identification requirements (Automatic Dependent Surveillance-Broadcast [ADS-B] for manned aircraft and Remote ID for UAS), the ability for cross-identifying those aircraft during flight is unknown. For instance, the ADS-B technology has both "in" and "out" functionality, which allows operators to broadcast their position ("out") and receive traffic information from other aircraft ("in"). This provides the aircraft operator with an enhanced situational awareness of other aircraft operating in their vicinity. On the UAS side, Remote ID allows requires operators to broadcast a number of variables from takeoff to shutdown, however it is unknown if UAS operators are able to view the position of manned aircraft, or vice versa.

4.3.3.2 Loss of Navigational Capability

The second most common risk mentioned was an inadvertent loss of UAS control, due to navigational capability. This loss of control condition is unique, as UAS utilize GPS positioning to improve their spatial accuracy, as well as navigate in automated flight. With UAS operating at low altitudes and in obstacle-rich environments, loss of navigational accuracy can be more prevalent. When this event occurs, UAS positional stability and certain features of automated flight become unavailable. This can increase the risk of loss of aircraft controllability, maneuverability, and loss of control.

4.3.3.3 Pilot Proficiency

The third most common risk mentioned was the topic of pilot proficiency. In manned aviation, pilots are subject to strict proficiency requirements before being legally allowed to operate an aircraft in solo or passenger-carrying operations. These requirements are conducted by a qualified flight instructor, who observes the pilot in the flight environment. For UAS pilots, there are requirements for certification and recurrent training, however much of the process is conducted individually without a formal flight test or oversight procedure. A number of respondents to the survey indicated the risk of pilots operating with little experience, or after extensively long durations between flights.

4.4 Recommendations for Voluntary Submission of UAS FDM

The final question in the survey asked participants how UAS operators could be encouraged to voluntarily submit their flight data for safety analysis and sharing through the ASIAS program. Participants offered a wide range of possibilities, including freedom from certificate action (similar to ASAP), reduced insurance premiums, priority or increased success in FAA waiver authorizations, and equipment subsidies for flight data recording capabilities.

One option that has gained significant traction in dialogue with operators is the idea of easy and automated data transfer. If flight data could be automatically transferred after flight, or there was an easy transfer mechanism through the GCS, operators would be much more likely to submit the data. Conversely, if the operators are required to download, process, and use alternative processes to submit the data, they are much more unlikely to use the system, unless there is a significant incentive to do so.

Numerous respondents in the survey noted that future research and capabilities should center around data harmonization, establishing a flight data recording standard for UAS, and establishing recommendations for automated data transfer mechanisms from the UAS to the ASIAS data repository.

5 CREATING ANALYTICS

5.1 Mapping the Data to Logical Parameters

When it comes to flight data, manufacturers may use any naming convention to identify parameters on a recorder. Flight data recorders may come with a data dictionary or Logical Frame Layout (LFL) document to help the analyst decode the parameters, however these documents may also not exist or be difficult to obtain. In the case of UAS flight data, the latter case is common. Given the pace of technological changes and lack of standardization, LFL documents may not have been created or are outdated. In any case, before meaningful analysis can be conducted, the manufacturer’s naming convention must be mapped to a logical parameter. This logical parameter would be used to later compare common events across various fleet types and aircraft operators.

There are two common methods to map flight data to logical parameters. The first method is to map the parameters during the initial data importing step. This method is commonly used with the flight data monitoring programs employed by commercial airlines, as the flight data recorders are held to strict standards for data standardization and configuration. Additionally, this method requires the LFL to be supplied before data can be imported into the database and no analysis can be conducted without proper data onboarding and mapping. As such, this method requires the most up-front work effort and may delay the use of a flight data monitoring program for a new customer or new fleet type. This method is not applicable to UAS flight data monitoring since data standardization and configuration has not yet been addressed.

The second method for mapping flight data to logical parameters occurs at the event definition step. This method is what the NGAFID uses and provides more flexibility in working with UAS flight data, while providing an equivalent level of harmonization across fleets. In this method, flight data is imported to the database as-is, without initial mapping to logical parameters. When a user creates a new event type, they will select the appropriate parameters from the raw flight data to define the event. Additionally, the user will set fleet specific limits, based on the limitations of the manufacturer or operator. Once the event is created, only the required parameters will be retrieved from the database and used to compute the event. This provides a number of benefits, including the ability to immediately import flight data once it becomes available, not requiring a manufacturer-provided LFL, and significant savings to database storage and processing power. The main drawback to this method is that the operator must have some knowledge of their flight data in order to determine which raw parameters to use during the event definition. That being said, this method is currently the ideal process for UAS flight data until the issue of data standardization and configuration is solved and guidance is provided to manufacturers for implementation across their fleets. At the time of this report, the expertise of the NGAFID project team is provided to all users of the system, so that event definitions can be developed by NGAFID developers and not the end users. The project team believes this is a significant opportunity for future research and an obstruction that must be overcome in order to ensure the safe integration of UAS into the NAS through flight data monitoring.

5.2 Comparison and Harmonization Across Fleets

Ultimately, the goal of a flight data monitoring program is to monitor trends in flight safety within singular fleets, as well as compare event rates across varying fleet types. To effectively compare events across fleets, there must be some level of harmonization at the event level. In the NGAFID, this occurs when defining events of a common type. This is where the operator will select the parameters appropriate to their specific aircraft's flight data to create the event definition. To harmonize across fleets, the event must share a common name with events defined for other fleet types. This step is monitored and assisted by the NGAFID project team to ensure that event comparisons can be effectively made across fleet types in the database. An example is provided below.

Let's say an operator wishes to define an event related to exceeding the maximum flight altitude. In this case, an operator might set the limit at 400 feet AGL, but select the specific recorded altitude from the aircraft's available parameters. In this example, the NGAFID computes the difference in the operator-selected MSL altitude and subtracts the terrain elevation to compute an AGL height. For one aircraft, the operator-selected altitude might be GPS recorded altitude, while another, more sophisticated aircraft might record barometric altitude. In both cases, the end result is an event that monitors when an aircraft exceeds 400 feet AGL, which can be equally compared across fleet types.

6 RISK OF COLLISION ANALYTICS

6.1 Background

Risk of Collision between manned and unmanned aircraft is considered one of the leading risk factors of UAS integration into the NAS. Earlier in this project, the research team asked industry participants what the biggest safety risk of UAS integration into the NAS and the leading response was "collision with manned aircraft," as shown in Figure 13. To this end, the researchers pursued a risk of collision metric, which evaluated available flight data and calculated a proximity event to identify when two aircraft passed in close proximity, thus providing detailed information to assist operators in assessing their risk of collision. This metric also will provide the ASIAS team with tools to identify airspace locations where risk of collision may be higher, thus requiring procedural intervention.

To analyze the risk of collision between manned and unmanned aircraft in the NAS, the project team created an algorithm to identify close-proximity events. This algorithm is predicated on the closest point of approach and is calculated by comparing the aircraft's position versus all other aircraft.

The proximity event was created as one mechanism to evaluate the safety risk of close proximity events. It is important to note that the tools and algorithms created do not constitute a proper risk metric, as risk analysis is a functional component of each individual organization. Rather, the project team has developed tools, analytics, and visualizations that will assist an organization in conducting a proper safety risk assessment as part of their internal processes for risk analysis.

Limitations of this event primarily center around the coverage of flight data within the NGAFID database. The algorithm itself is sound, however it depends on the flight data from other aircraft to compare location and trigger a proximity event. To date, participation in the NGAFID has been voluntary and most data is located around general aviation "hubs" (flight schools and airports with a high density of general aviation operations). To increase the capabilities of this event, additional track data could be added to the NGAFID in the form of historical and real-time radar track data and ADS-B Out data streams.

6.2 Event Logic for Risk of Collision Metric

Risk of collision was addressed by analyzing aircraft flight data and triggering a proximity event, which was calculated as a closest point of approach between that aircraft and other aircraft with processed flight data in the NGAFID. This algorithm could be applied to any database of flight data to achieve similar results, or additional flight data could be added to the NGAFID to improve the coverage of the proximity event logic.

The proximity event is triggered when two aircraft pass within 500 feet of each other. Due to the close proximity of both aircraft, the proximity is calculated as a Euclidian distance, which is a linear closest point of approach, rather than a spherical approach. This reduces the computational cost of calculating the event step, without sacrificing a significant amount of accuracy in the proximity calculation. Finally, a "severity" number is calculated for the event, which is determined by the closest distance recorded between the two aircraft during the event.

As the entries in the NGAFID Flights table contain the start and end time of the flight, it is first possible to select flights which could possibly be in the air at the same time: where si and ei are the start and end times of the flight being processed, and sj and ej are the start and end times of potential matches, time matched flights can be found querying for $sj \leq ei \wedge ej \geq si$. Following this, those flights can then be further filtered by latitude and longitude, in a similar fashion, as only flights with $(min(latj) \leq max\ (lati)) \wedge (max\ (latj) \geq min(lati)) \wedge (min(lonj) \leq max\ (loni)) \wedge (max\ (lonj) \geq min(loni))$, given the precomputed minimum and maximum latitude values. Each of these query parameters are indexed as b-trees in the mySQL database to further improve performance. Given this filtered subset of flights, the proximity events can then be calculated for matching flights by skipping forward in time to the beginning of flight with the later start time and then for each moment in time calculating the two flights proximity given their altitude (above sea level), latitude and longitude, with a distance under 500 ft triggering the event when both flights have an altitude (above ground level) higher than 50 feet. It is important to note that the data is taken and computed at the recording rate provided by the aircraft. In this case, dissimilar time series are taken at the points recorded, rather than interpolating intermediate positions between data points. Finally, the NGAFID calculates the minimum value of proximity and identifies the "severity" of the event as the closest point of approach during the event period (see Appendix H).

Closure rate is also shown in the NGAFID for a Proximity Event, which provides a time-series depiction of the closure rate (and divergence rate) between the two aircraft during the event. This aids the operator in assessing the level of risk of an event. For instance, if two aircraft are engaged in formation flying activities, they may have a close proximity, however their closure rate will be small, and thus carry a relatively low risk. Likewise, two aircraft converging head-on would have a high closure rate, and likely be assessed as a higher risk of collision. Figure 15 shows an example of the closure rate chart, as provided in the NGAFID.

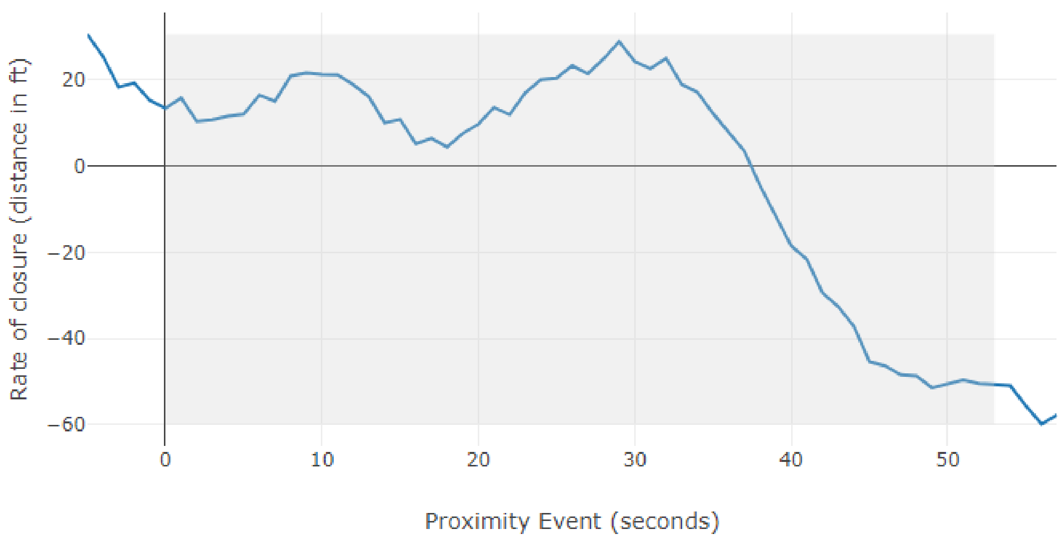


Figure 155. Closure Rate Chart in NGAFID.

6.3 Ongoing Work for Risk of Collision Metric

While creating the metrics for Risk of Collision, the team identified one additional capability that is planned for development in the NGAFID. This capability is not directly part of this task order, but is planned for development and implementation in the near future. The project team found it important to broadly outline this capability, as it will assist the ASIAS team in further evaluating future collision risks.

The capability identified is to add a density map for locations where proximity events occur. As indicated above, the data in the NGAFID is currently located mostly around flight schools and high-traffic general aviation airports. While more data is added to the NGAFID, higher density of proximity events will allow the ASIAS team to identify pockets of airspace that could benefit from procedural control or traffic decongestion.

7 ADDITIONAL NOVEL ANALYTICS FOR UAS FDM

7.1 Background

As a benefit of the scope and nature of this project, the structure of the NGAFID has been developed to accommodate flight data from various UAS, which allows for the calculation of novel risk elements specific to UAS flight. Risk of Collision between manned and unmanned aircraft is considered one of the leading risk factors of UAS integration into the NAS. Earlier in this project, the research team asked industry participants what the biggest safety risk of UAS integration into the NAS and the leading response was “collision with manned aircraft”, as shown in Figure 13. To this end, the researchers pursued a risk of collision metric, which evaluated available flight data and calculated a proximity event to identify when two aircraft passed in close proximity, thus providing detailed information to assist operators in assessing their risk of collision. This metric also will provide the ASIAS team with tools to identify airspace locations where risk of collision may be higher, thus requiring procedural intervention. Details of the Risk of Collision metric can be found above, which describes the details associated with calculating and displaying this metric on the NGAFID website.

Beyond the Risk of Collision metric, the project team was asked to develop analytics for at least one novel risk that was identified in the industry survey. When evaluating the survey, the researchers determined multiple analytic events that could be reasonably developed within the scope of this task, which further enhances the capabilities of this research moving forward.

7.2 Novel Analytics and Event Logic

A thorough review of the survey revealed that industry professionals view risk of collision as the biggest safety risk of UAS integration into the NAS. Secondarily, a number of flight safety risks presented themselves that could be reasonably monitored in the NGAFID database. From the survey, the research team found four main themes and chose to develop one novel analytic event from each of the four themes. These themes were identified as (1) a low battery event, (2) an operational exceedance of an aircraft limitation, (3) a procedural exceedance of a regulatory requirement, and (4) a loss of operational or navigational capability.

7.2.1 Low Battery Level and Battery Not Charging Events

One of the most critical events to monitor on any aircraft is the status of the powerplant. In UAS aircraft, the battery is the system that directly relates to the airborne capabilities of the aircraft. For small UAS, the battery may be the only source of power available for the aircraft, whereas in larger UAS that are powered by traditional fuel propulsion, the battery may be supplemented by an alternator or generator, which charges the battery during powered flight. Because both of these systems may be common in UAS, the researchers chose to develop two events related to battery power on the aircraft, (1) Low Battery Level and (2) Battery Not Charging.

7.2.1.1 Low Battery Level

Low Battery Level is an event that was created to detect and monitor flight conditions where a UAS is airborne with less than 25% battery power remaining. All UAS flight data that have been collected for this project record battery level as a percentage of maximum capacity. That is not to say that battery level will always be recorded, but as it is a critical parameter for monitoring battery capacity and performance, the research team expects most UAS to have this parameter available in the data stream.

Monitoring the safety risk of operations at low battery levels is a challenging task. At the most fundamental level, the research team chose to set a fleet constant of 25% to trigger this event. That being said, many manufacturers calculate a dynamic "return to home" safety feature that calculates the time required to fly the UAS to the landing point, while also maintaining some reserve battery performance. This method is the most accurate assessment of risk, however it becomes challenging to calculate this in the flight data, since the "home" location is not a recorded parameter. Future development efforts could improve upon this event by considering the "starting point" of the flight data as the "home" location and derive a time-based calculation of flight time required to fly the UAS to the "home" location.

Logical event definition for the Low Battery Level event can be found in Appendix I.

7.2.1.2 Battery Not Charging

In UAS with alternative power systems, such as a fuel-based internal combustion engine, the battery is not the main source of power. However, since much of the UAS is functionally dependent on electrical power to operate its payload and systems, a monitoring of the electrical status is an important analytic to include in a safety data monitoring program. For this reason, the research team developed an analytic to monitor electrical parameters and determine when the battery is not charging. This would be an indication that the alternator or generator is inoperative and the battery is the main source of electrical power. In these situations, it would be important for the UAS operator to land the aircraft as soon as practical.

Logical event definition for the Battery Not Charging event can be found in Appendix J.

7.2.2 Operational Exceedance of an Aircraft Limitation

All aircraft have an operational envelope by which controlled flight is considered safe. These operational envelopes are developed during the design, development, and testing of the airframe. Documentation of these operational envelopes can often be found in the operator's handbook. The researchers determined that an important step in demonstrating the functional capabilities of a flight data monitoring system would be to create an event related to a single aircraft operational exceedance. The event chosen for this project was a Low Battery Temperature event.

Logical event definition for the Low Battery Temperature event can be found in Appendix K.

7.2.3 Procedural Exceedance of a Regulatory Requirement

Part 107 of the Federal Regulations outlines important operational requirements for UAS. As a demonstration of the analytical capabilities, the researchers chose one regulatory requirement to monitor for all UAS. This event is a High Altitude Limit Exceeded event, which monitors the aircraft's altitude AGL and compares it to the regulatory requirement to be at, or below, 400 feet AGL. This event uses the aircraft's recorded altitude (in barometric or GPS altitude) and compares it against publicly-available terrain data to calculate an AGL height at any given point along the flight path. In most cases, GPS resolution is sufficient to be confident in the aircraft's geospatial position above the ground, which improves the accuracy of the AGL calculation in the NGAFID.

Logical event definition for the High Altitude Limit Exceeded event can be found in Appendix L.

7.2.4 Loss of Positional Accuracy or Navigational Capability

UAS are dependent on an accurate GPS position for many of their flight functions, including auto flight and stability control. The operational challenge UAS face is that they are often flown in terrain-rich, obstacle-rich, and low altitude environments, where GPS reception can be minimal and prone to signal dropouts. As GPS degradation is a critical function of the controllability of the UAS, the researchers created an analytic to monitor Low Positional Accuracy for UAS, which monitors the GPS capabilities of the UAS throughout the flight. For some UAS, this is monitored through a unique "GPS accuracy" parameter, however GPS resolution could also be derived from the dilution of precision recorded parameters within the flight data stream.

In the example of DJI manufactured UAS, the parameter to monitor positional accuracy is titled, "GPS Health." This parameter is on a scale of 0 to 5, with values directly tied to the operational functionality of the UAS. For instance, a GPS Health value of 0, 1, or 2 results in no GPS Lock, a value of 3, 4, or 5 provides GPS lock, and a value of 4 or 5 provide the ability to set a Home Point for the aircraft to return in low battery situations.

Logical event definition for the Low Positional Accuracy event can be found in Appendix M.

8 VISUALIZING UNMANNED FLIGHT DATA

Display of all data-driven events and access to the data output can be conducted in a number of ways. This section of the deliverable outlines the various methods created by the project team to analyze the newly-created events. The intent with this was to provide a variety of means of analysis, so that an operator can accurately assess their level of risk, as it pertains to their internal organization's safety risk assessment process.

8.1 Raw Data Exports

To allow an operator to download raw data exports of their flights for analysis, the NGAFID allows a few options. First, the operator can download the full flight data from the individual flights. This allows additional flexibility in the event an operator wishes to analyze the flight in a third-party software outside of the NGAFID web portal. Secondarily, the NGAFID configures each event into a categorical export, which can be downloaded as a CSV file. These categorical exports identify the flight number, aircraft type time of event, and the severity (which is the closest proximity, in feet, during the event). The data is exported as a CSV file and can be accessed through the "Flights" tab or the "Events" tab on the NGAFID. Table 3 shows the export functionality currently available, while Table 4 shows a data dictionary for the "Event Severities" CSV export to describe the type of data available for export.

Table 3. NGAFID Data Export Functionality.

Export Name	Description

Flights	Allows the user to export the time-series data of all parameters in the original flight file. This data is pulled from the database and recompiled into a CSV file for a single flight.
Event Trends	Allows the user to view events plotted on a chart over time. This chart can be graphically viewed on the web interface, as well as exported as a CSV file, formatted to show Total Events, Flights With Events, and Total Events for each calendar month in the database.
Event Severity	Allows the user to view events plotted on a chart, based on their Severity value. The Severity is the maximum or minimum value of the most pertinent parameter exceeded during the event. This data can be exported to a CSV file, which shows the event and severity for all flights in the database. A sample data dictionary for this file type is provided in Figure 11.

Table 4. Data Dictionary for “Event Severities” in CSV Export.

Field Name	**Data Format**	**Description**	**Example**
Event Name	Text	Provides a short name for the type of event triggered during the flight	VSI on Final
Airframe	Text	Provides the common name of the airframe.	Cessna 172S
Flight ID	NNNNNN	Indicates the flight identification number applied to the flight during processing.	66951
Start Time	Date	Start date and time when the event began.	1/3/2020 11:32:39 AM
End Time	Date	Start date and time when the event ended.	1/3/2020 11:32:41 AM
Start Line	NNNNN	Line within the single flight when the event started.	3293

		Identified as seconds from beginning of file.	
End Line	NNNNN	Line within the single flight when the event ended. Identified as seconds from beginning of file.	3295
Severity	Numeric	The maximum, or minimum, value during the event period. Indicated as the value of the parameter indicated in the Event Name.	-1562.91

8.2 Event Counts

On the NGAFID homepage, a user will be able to see a running sum of their event counts over the duration of their data. Figure 17 shows one operator's fleet of aircraft, along with the associated event counts. This operator contains a mixed fleet of aircraft, including manned and unmanned, as well as fixed wing and rotorcraft. In this case, the operator can choose to see all of their aircraft events (as shown in the figure) or they can filter by a specific airframe type and only see those events. The Events displayed will be those that are applicable for that particular operator's fleet of aircraft.

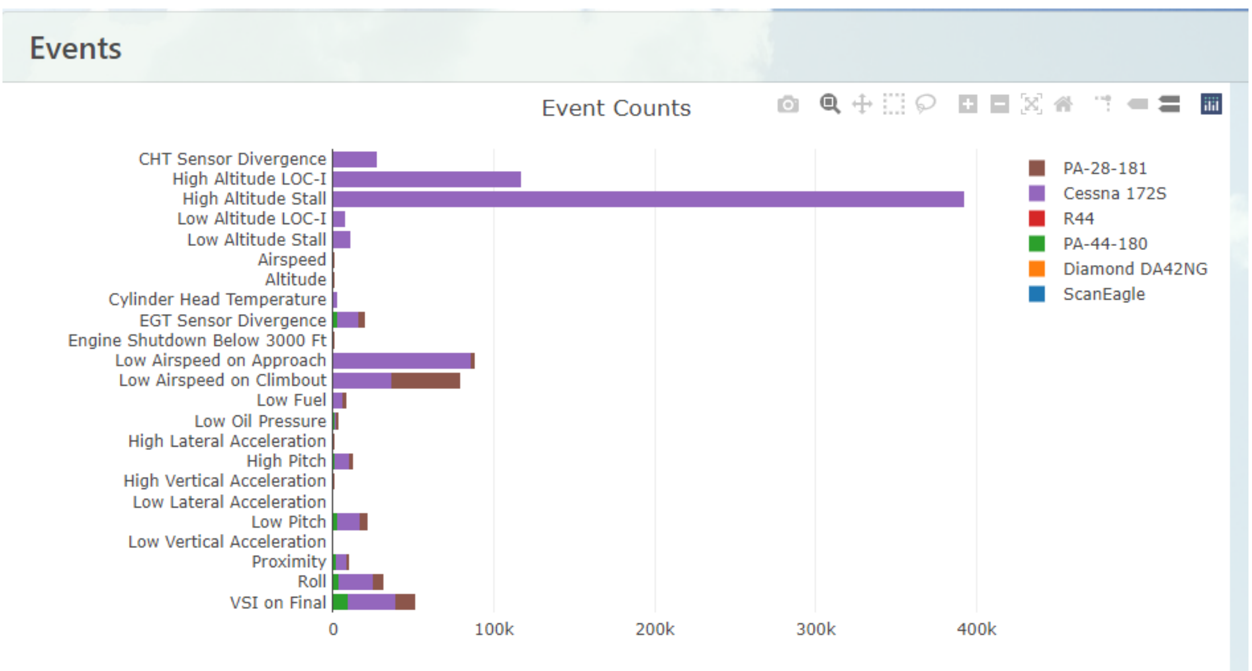


Figure 16. Event Counts Chart on NGAFID.org.

8.3 Event Rates

In a similar method to Event Counts, the NGAFID home page includes a chart that shows Percentage of Flights with Events. This chart is particularly useful in comparing an operator's event rates to the event rates of other similar operators with data in the NGAFID. Again, the chart shows the percentage of flights that triggered each type of event, pertaining only to those events that apply to the operator's fleet. In this case, the operator can choose to see all of their aircraft events (as shown in the figure) or they can filter by a specific airframe type and only see those events.

This functionality of the NGAFID allows the operator to compare their operational risk relative to other operators. In the case of the novel events created, it would assist the operator in addressing the "likelihood" of the event occurring, when using the risk assessment matrix in the safety risk assessment process.

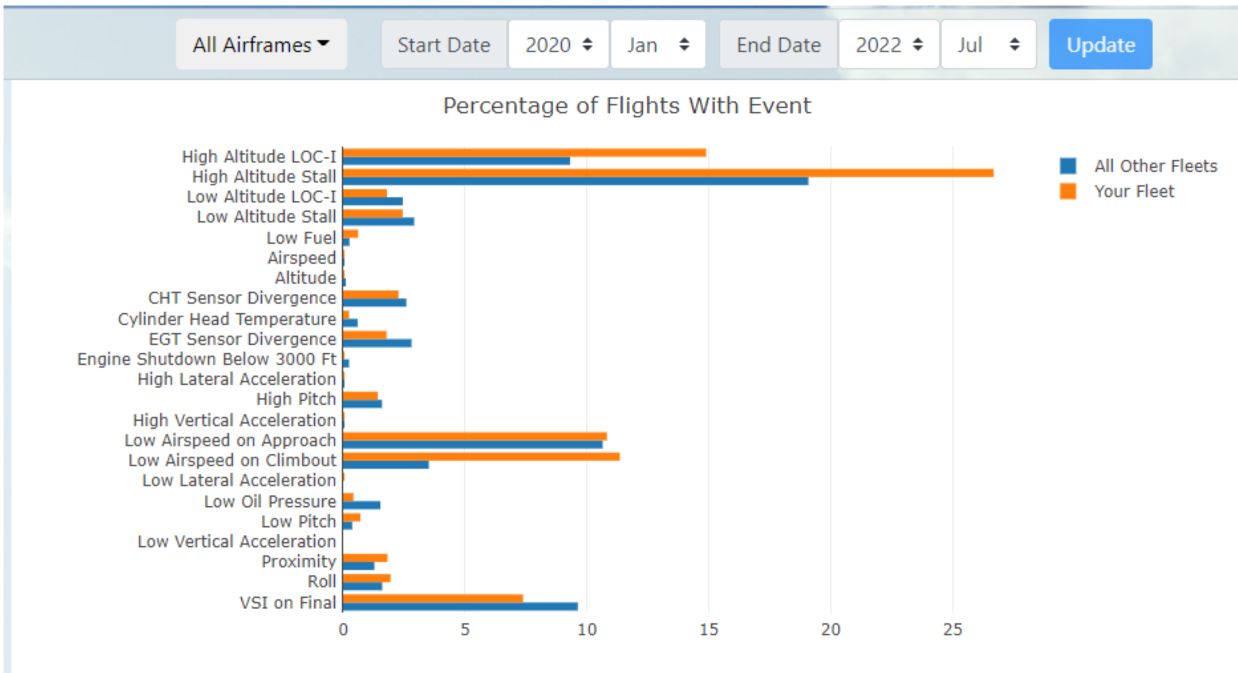


Figure 17. Percentage of Flights With Event Chart on NGAFID.org.

8.4 Map View

After an operator selects the applicable flight with an event, they are taken to a map view page. This page allows the operator to display the flight track in its entirety. Additionally, the operator can select the specific event to visually display the segment of the flight when the event was active. The active segment is the duration of the flight when the aircraft were within designated conditions that triggered the event.

Within the map view display is the details of the event. This detail shows the start time, end time, and severity of the event. Particularly important for a safety risk analysis is the "severity" metric, which is calculated as the maximum or minimum parameter units during the event. For instance, for High Altitude Limit Exceeded, the "severity" parameter would be the maximum altitude the aircraft achieved during the event. Alternatively, in the Low Battery Temperature event, the "severity" would be the minimum recorded battery temperature during the event. In Figure 19, an example of the Proximity event is shown, where two aircraft are conflicting while entering and exiting a holding pattern. The event "severity" shows the aircraft approached as close as 478.08 feet, before diverging paths.

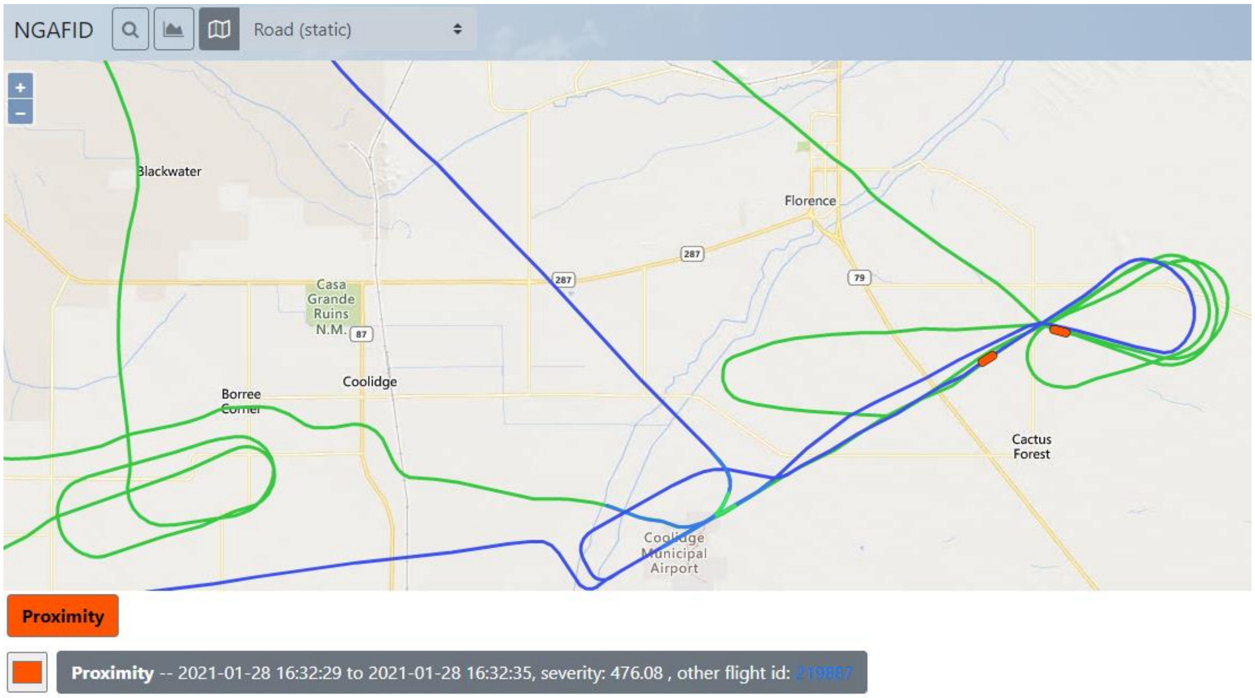


Figure 18. Map View of Proximity Event on NGAFID.org.

8.5 Cesium Flight Replay

The final visualization available on NGAFID to view the event is a three-dimensional replay feature through the Cesium interface. This functionality allows the operator to view the flight track of the aircraft and dynamically adjust the replay speed and camera angles to understand the flight dynamics during the event. While it is difficult to describe the three-dimensional dynamics of this visualization feature in a static report, the figures below help to show the capabilities of the system, with an example of the newly-created Proximity event.

Figures 20, 21, and 22 show the same Proximity event as Figure 19, but through the Cesium flight replay interface. Figure 20 shows the two aircraft approaching the Proximity event. Figure 21 shows an overhead view of the two aircraft at the closest point of approach during the Proximity event. Finally, Figure 22 shows a profile view at the closest point of approach during the Proximity event. By using the Cesium replay feature, the operator is able to determine if the Proximity event was vertically separated as the aircraft entered and exited the holding pattern.

Figure 19. Cesium Replay Feature – Approaching the Proximity Event.

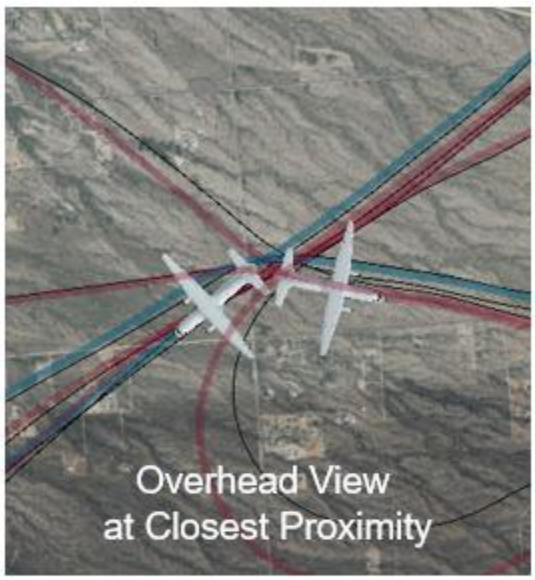


Figure 20. Cesium Replay Feature – Overhead View at Closest Proximity.

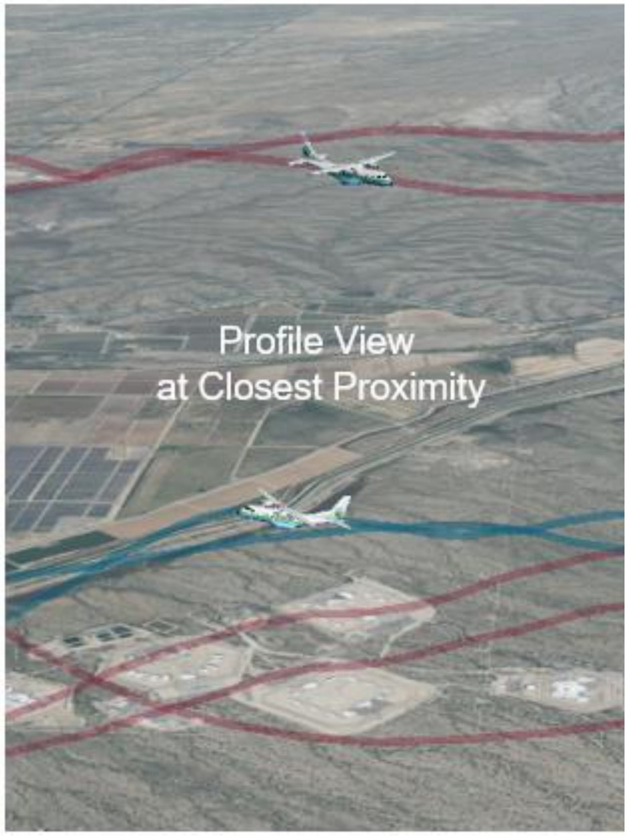


Figure 21. Cesium Replay Feature – Profile View at Closest Proximity.

9 LIMITATIONS AND FUTURE RESEARCH

As the research team outlined throughout the project in deliverables, interchange meetings, and project management reviews, the consistency, accuracy, and cleanliness of UAS data will prevent a scalable flight data monitoring solution for UAS data in the ASIAS program. Throughout the project, the researchers continued to uncover challenges with data standardization, including consistency in the recorded parameters, cleanliness of the data provided, and an understanding of what the parameters were recording within the flight data. As listed in the following sections, the researchers encourage consideration of continuing UAS flight data research to include recommendations for data standardization, data harmonization, and a minimum data standard for incorporation and comparability with other flight data in the ASIAS program.

9.1 Data Accessibility

UAS flight data can typically be accessed through a number of methods, which each require a different means of access and return a different data product. For this project, the team collected flight data from the UAS platform, through an on-board data card and through the aircraft's telemetry stream. Alternatively, data could be collected from the ground station or the telemetry between the control station and the UAS platform. In any case, flight data is not provided in a consistent or easily accessible method for the end user.

For UAS FDM to be a viable solution for the ASIAS program, manufacturers should be encouraged to provide an efficient means of access to the UAS flight data, as well as provide a data dictionary to decode the flight data into logical parameters for flight data analysis.

As UAS operators typically collect data products from their platforms (LIDAR, photo, video, or other sensor data), collecting a separate data stream for the purposes of FDM is a functional hurdle. To improve the adoption and reduce the technical barriers to entry for UAS FDM in the ASIAS program, research is encouraged to determine which means of data access are most likely to be used by a UAS operator after their flights are completed.

9.2 Data Quality

The quality of data from UAS aircraft and systems is largely uncontrolled and widely varied. Because of the lack of a data standard, data quality is a challenge with using UAS data for a FDM solution. With no data quality controls, the ASIAS program and the end user lack the necessary confidence in the analytic results that are generated from the data. Additionally, comparison of UAS data and existing manned aircraft in the ASIAS program becomes troublesome, as the data quality standards and expectations between the two datasets are unmatched.

The research team saw data quality as a challenge at numerous points during this project and noted it as a significant opportunity for future research in this space. In order to adopt UAS flight data into the ASIAS program, steps must be made to ensure data quality matches the minimum standard required to produce valid safety analytic results. Future research would be warranted on this topic in particular before a widespread adoption of UAS data into ASIAS could be achieved.

9.3 Data Consistency and Standardization

The biggest challenge is the concept of a data standard for UAS flight data. Currently, each manufacturer formats their data in a different method, with a lack of a common naming convention or data unit standard. It is for this reason that the NGAFID is configured in such a way as to collect and store all available information from each UAS data file, so as not to lose any valuable information. One additional consideration when it comes to a data standard, would be to apply a minimum recording standard for each manufacturer to comply. This would enable a minimum set of exceedances to be calculated on all UAS, regardless of manufacturer.

10 CONCLUSION

The project built upon ASSURE project A20, which identified UAS parameters, exceedances, and recording rates for UAS and moved to incorporate the data into the NGAFID. The project successfully achieved its objectives by configuring storage and formatting requirements for unmanned data, developing a prototype system to collect unmanned Flight Data Monitoring records, collecting over 1000 flights of UAS data, and interfacing with unmanned communities to gather industry feedback. The project team managed the effort and coordinated with the FAA through Program Management Reviews, Technical Interchange Meetings, interim reports, emails, and telephone meetings to ensure the research validation objectives were being met.

The successful implementation of this project is a significant milestone for UAS safety. It provides valuable information and data on the safety and performance of UAS operations, which can help improve safety risk management and enable the development of new safety technologies for UAS. The project used a combination of information technology and outreach to a diverse assortment of stakeholders, including manufacturers, unmanned operators, and regulators, to ensure that the data collected and analyzed are relevant and representative of the industry. The project's success is a testament to the collaboration between industry and government in promoting UAS safety.

Based on the findings of the research team, there are numerous future research opportunities in the area of UAS flight data for the ASIAS program. The team identified challenges with data consistency, accuracy, and cleanliness, and recommended future research to develop a data standardization framework. This framework should include data harmonization and a minimum data standard to ensure comparability with other flight data in the ASIAS program. Another area for future research is data accessibility, including identifying efficient means of accessing flight data and providing a data dictionary to decode flight data into logical parameters for analysis. Additionally, the quality of UAS flight data is largely uncontrolled and varies widely, so future research should focus on establishing data quality controls to ensure that safety analytic results are valid. Finally, future research should address the lack of a common naming convention or data unit standard for UAS flight data, with the aim of developing a minimum recording standard that all manufacturers can comply with.

Moving forward, the data collected by this project can be used to support research, policy development, and safety risk management activities related to UAS. The project's achievements have contributed to the development of a robust system to collect and process UAS flight data for the purposes of safety monitoring and system integration, which will continue to evolve as the UAS industry matures. It is essential to continue to collect and analyze data on UAS operations to improve safety and enable the safe integration of UAS into the National Airspace System.

11 APPENDIX A

Common Manned Aviation Flight Events Used in FDM

Event	Phase of Flight	Level 1 (Fixed Wing Example)	Level 2 (Fixed Wing Example)	Notes for UAS
Excessive Power on the Ground	Before Takeoff	>=2000 RPM	>=2300 RPM	Applies, but limits need to be adjusted.
Excessive Starter Engagement	Before Takeoff	RPM range over 10 secs	RPM range over 15 secs	Won't apply for most commercial UAS, but would apply for large UAS with conventional engines requiring a starter.
Taxi Speed - Ramp	Before Takeoff	6 kts	8 kts	Applies for fixed wing UAS operating at an airport.
Taxi Speed - Taxiway	Before Takeoff	20 kts	25 kts	Applies for fixed wing UAS operating at an airport.
Hard Breaking - Taxi	Before Takeoff			Applies for fixed wing UAS operating at an airport.
Engine Run-Up - Excessive RPM Drop	Before Takeoff	200 RPM	300 RPM	Applies for reciprocating engines driven with magnetos.
Heading Variation at Power Application	Takeoff	10 deg	20 deg	Applies for fixed wing UAS, including both powered takeoffs and assisted takeoffs (catapult or bungee).
Low RPM at Rotation	Takeoff	2200 RPM	2000 RPM	Applies.
Airspeed at Liftoff (Non-Soft Field)	Takeoff	44 kts	40 kts	Applies.
Angle of Attack	Takeoff			Applies.
Pitch Attitude at Liftoff	Takeoff	10.5 deg	12 deg	Applies.

Flap Position on Takeoff	Takeoff		0 deg or greater than 10 deg	Applies depending on aircraft configuration and manufacturer recommended takeoff configuration.
Bank Angle	Takeoff	20 deg	25 deg	Applies.
Lateral g Loads	Takeoff			Likely doesn't apply, unless manufacturer designates a maximum G load limit.
Runway Distance Remaining @ Liftoff	Takeoff	700 ft	400 ft	Applies for UAS operating at airports.
Tail Wind Component	Takeoff	10 kts	15 kts	Applies.
Cross Wind Component @ 100 ft	Climb	>15 kts	>=20 kts	Applies if manufacturer designates a limit.
Airspeed on Climb Above 100 and Below 500 ft	Climb	57 kts	52 kts	Applies.
Bank Angle Below 400 ft	Climb	>30 deg	>=45 deg	Applies.
Flap Retraction	Climb	<60 kts	<55 kts	Applies depending on manufacturer recommended configuration and takeoff profile.
Altitude Decrease Below 500 ft	Climb	< 0 fpm	<-200 fpm more than 2 secs	Applies for safety of flight for fixed wing UAS, however some UAS operations may require this maneuver as part of the mission.
Max Altitude	Cruise	1 sec	30 min	Applies, but we should consider setting maximum altitudes based

				on FAA requirements at UAS position of operation.
Minimum Recovery Altitude	Cruise	500 ft	<=400 ft	Likely does not apply due to nature of UAS operations.
Turbulence Encounter	Cruise			Likely does not apply, depending on manufacturer recommended limitations.
Turbulence Penetration Speed	Cruise		>105 kts	Likely does not apply, depending on manufacturer recommended limitations.
VNE	ALL	158 kts	163 kts	
Vertical g Load	ALL	3.0g	3.8g	Likely doesn't apply, unless manufacturer designates G load limits.
Vertical g Load - Min	ALL	-1	-1.52	Likely doesn't apply, unless manufacturer designates G load limits.
Lateral g Limit	ALL			Likely doesn't apply, unless manufacturer designates G load limits.
Oil Temp - Max	ALL		245 F	Applies if UAS has a reciprocating engine with an oil system.
Oil Pressure - Min	ALL		20 psi	Applies if UAS has a reciprocating engine with an oil system.
Oil Pressure - Max	ALL		115 psi	Applies if UAS has a reciprocating engine with an oil system.
Max RPM	ALL	2700 RPM >=1 sec	2700 RPM >5 seconds	Applies
Fuel Quantity	ALL	8 gal	5 gal	Applies

Max CHT	ALL		500 F	Applies if UAS has a reciprocating engine
CHT Differential	ALL		?	Applies if UAS has a reciprocating engine
VFE 10 deg	ALL	>=108 kts	>110 kts	Applies
VFE >10 deg	ALL	>=84 kts	>85 kts	Applies
Data Error Detection	ALL		ANY	Applies
System/Equipment Failure Detection	ALL		ANY	Applies
Bank Angle	ALL	60 deg	>=65 deg	Applies
Bank Angle Below 1300 agl	ALL	50 deg	>=55 deg	Applies
Pitch Attitude (pos)	ALL	30 deg	>=35 deg	Applies
Pitch Attitude (neg)	ALL	-30	<=-30	Applies
Terrain Warnings	ALL			Applies
Engine Shutdown @ Altitude	ALL		EGT, Fuel Flow/Pressure & RPM	Applies
Stall Detection Below 1300 AGL (Using AoA)	ALL	Approach to Stall (Within 1 deg of AoA)	Stall	Applies
Glideslope Deviation	Appraoch			Unstable Approach event - Altitude depends on Specific operator limits
CDI Deviation	Approach			Unstable Approach event - Altitude depends on Specific operator limits
Vertical Speed Below 1000 AGL	Approach	>=800 fpm	>=1000 fpm	Unstable Approach event - Altitude depends on Specific operator limits
Airspeed @ or below 200 AGL (High Speed - Full Flaps)	Approach	66 kts @ 2 secs	71 kts @ 2 secs	Unstable Approach event - Altitude depends on Specific operator limits

Airspeed @ or below 200 AGL (High Speed - No Flaps)	Approach	75 kts @ 2 secs	80 kts @ 2 secs	Unstable Approach event - Altitude depends on Specific operator limits
Airspeed @ or below 200 AGL (Low Speed - Full Flaps)	Approach	60 kts @ 2 secs	<=56 kts @ 1 secs	Unstable Approach event - Altitude depends on Specific operator limits
Airspeed @ or below 200 AGL (Low Speed - No Flaps)	Approach	69 kts @ secs	<=65 kts @ 1 secs	Unstable Approach event - Altitude depends on Specific operator limits
On Extended Centerline @ 200 AGL	Approach	2 deg	3 deg	Unstable Approach event - Altitude depends on Specific operator limits
Glideangle (High) @ 200 AGL	Approach	4 deg	5 deg	Unstable Approach event - Altitude depends on Specific operator limits
Glideangle (Low) @ 200 AGL	Approach	2 deg	1 deg	Unstable Approach event - Altitude depends on Specific operator limits
Flap Position	Approach	Position and Changes Below 100 AGL		Unstable Approach event - Altitude depends on Specific operator limits
Bank Angle @ or below 200 AGL	Approach	20 deg	25 deg	Unstable Approach event - Altitude depends on Specific operator limits
Tail Wind Component @ 200 AGL	Approach	10 kts	15 kts	Unstable Approach event - Altitude depends on Specific operator limits
Cross Wind Component @ 200 AGL	Approach	>15 kts	>=20 kts	Unstable Approach event - Altitude depends on Specific operator limits
Pitch Attitude (High) @ Touchdown	Touchdown	10.5 deg	12 deg	Applies
Pitch Attitude (Low) @ Touchdown	Touchdown	3 deg	1 deg	Applies

Airspeed (High - Full Flap) @ Touchdown	Touchdown	55 kts	60 kts	Applies
Airspeed (High - No Flap) @ Touchdown	Touchdown	63 kts	68 kts	Applies
Hard Landing	Touchdown			Applies
Latteral g	Touchdown			Specific to aircraft type
Centerline Tracking	Touchdown/Rollout			Applies
Bounced Landing	Touchdown	Multiple Ver g Spike		Applies
Excessive Braking	Touchdown			Applies
Touchdown Point	Touchdown	1500 ft remaining	1000 ft remaining	Applies
EXTRA – UAS SPECIFIC				
Inconsistent RPM during start-up	Before Takeoff			Considers inconsistent RPM on multi-rotor UAS during the startup sequence.
Low power remaining - Caution	ALL			
Low power remaining - Warning	ALL			
GPS Resolution Lost	ALL			
Telemetry Lost	ALL			
Maximum wind limit	Takeoff			Considers manufacturer recommended wind limits.
Airspace proximity - Caution	ALL			Considers UAS position and airspace proximity.
Airspace proximity - Warning	ALL			Considers UAS position and airspace proximity.

Strength of Signal	ALL			Indication of radio signal strength between UAS and ground transmitter
Battery/Power Excessive Dissipation	ALL			Power remaining is dissipating quicker than expected
Battery Capacity Reduction	ALL			Battery has a low capacity – may occur over time
Battery Overheat	ALL			
Sensor Overheat				
Sensor Platform Jam				
Airborne Risk of Collision	ALL			Post hoc analysis of risk of collision based upon proximity to other aircraft

12 APPENDIX B

Data Types Common Across Commercial and General Aviation

1	**Field:**	id	**Type:**	bigint(20)	**Required**
	Units:	Integer	**Range:**	0-18,446,744,073,709,551,615	
	Description:	Individual record id, will be auto-incremented			
	Example:	1			

2	**Field:**	flight	**Type:**	bigint(20)	**Required**
	Units:	Integer	**Range:**	0-18,446,744,073,709,551,615	
	Description:	Used for flight identification. *Note: flight field will be foreign keyed to other tables which will allow for an individual organization to control the level of identification maintained in the overall database.			
	Example:	52			

3	**Field:**	phase	**Type:**	tinyint(3)	**Required**
	Units:	Integer	**Range:**	0-255	
	Description:	Phase of flight, to be foreign keyed to a master phase of flight table. Phase field will be used in the development of exceedances and other concept tools.			
	Example:	15			

4	**Field:**	time	**Type:**	Bigint(20)	**Required**
	Units:	Milliseconds	**Range:**	0-18,446,744,073,709,551,615	
	Description:	The millisecond that the field recorded occurred during flight (**not** the time the data was entered in the database).			
	Example:	29888824			

5	**Field:**	pressure_altitude	**Type:**	float(7,2)	**Not Required**
	Units:	Feet	**Range:**	-99,999.99 - 99,999.99	
	Description:	Pressure altitude if recorded (not derived).			
	Example:	12,432.11			

6	**Field:**	msl_altitude	**Type:**	float(7,2)	**Not Required**
	Units:	Feet	**Range:**	-99,999.99 - 99,999.99	
	Description:	Altitude above mean sea level.			
	Example:	12,432.11			

7	**Field:**	indicated_airspeed	**Type:**	float(6,2)	**Not Required**
	Units:	Knots	**Range:**	-9,999.99 - 9,999.99	
	Description:	Indicated airspeed.			
	Example:	124.21			

8	**Field:**	tas	**Type:**	float(6,2)	**Not Required**
	Units:	Knots	**Range:**	-9,999.99 - 9,999.99	
	Description:	True airspeed (not derived)			
	Example:	124.21			

9	**Field:**	mach	**Type:**	float(3,2)	**Not Required**
	Units:	Mach	**Range:**	-9.99 - 9.99	
	Description:	Mach number (not derived)			
	Example:	.86			

10	**Field:**	heading	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees	**Range:**	0-359.99	
	Description:	Compass heading, as recorded.			
	Example:	227.41			

11	**Field:**	course	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees	**Range:**	0-359.99	
	Description:	Magnetic course (not derived)			
	Example:	301.34			

12	**Field:**	pitch_attitude	**Type:**	float(7,4)	**Not Required**

	Units:	Degrees	**Range:**	-180.0000 - 180.0000	
	Description:	Pitch attitude, negative denotes down, positive denotes up.			
	Example:	6.8724			

13	**Field:**	Roll_attitude	**Type:**	float(7,4)	**Not Required**
	Units:	Degrees	**Range:**	-180.0000 - 180.0000	
	Description:	Roll attitude, negative denotes left, positive denotes right.			
	Example:	6.8724			

14	**Field:**	radio_transmit	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Radio transmission in progress.			
	Example:	no			

15	**Field:**	eng_1_rpm	**Type:**	float(7,2)	**Not Required**
	Units:	RPM	**Range:**	0 - 99999.99	
	Description:	Engine #1 RPM			
	Example:	2315.62			

16	**Field:**	eng_2_rpm	**Type:**	float(7,2)	**Not Required**
	Units:	RPM	**Range:**	0 - 99999.99	
	Description:	Engine #2 RPM			
	Example:	2315.62			

17	**Field:**	eng_3_rpm	**Type:**	float(7,2)	**Not Required**
	Units:	RPM	**Range:**	0 - 99999.99	
	Description:	Engine #3 RPM			
	Example:	2315.62			

18	**Field:**	eng_4_rpm	**Type:**	float(7,2)	**Not Required**
	Units:	RPM	**Range:**	0 - 99999.99	

	Description:	Engine #4 RPM			
	Example:	2315.62			

19	**Field:**	eng_1_mp	**Type:**	float(6,3)	**Not Required**
	Units:	Inches of HG	**Range:**	0 - 999.999	
	Description:	Engine #1 Manifold Pressure			
	Example:	25.812			

20	**Field:**	eng_2_mp	**Type:**	float(6,3)	**Not Required**
	Units:	Inches of HG	**Range:**	0 - 999.999	
	Description:	Engine #2 Manifold Pressure			
	Example:	25.812			

21	**Field:**	eng_3_mp	**Type:**	float(6,3)	**Not Required**
	Units:	Inches of HG	**Range:**	0 - 999.999	
	Description:	Engine #3 Manifold Pressure			
	Example:	25.812			

22	**Field:**	eng_4_mp	**Type:**	float(6,3)	**Not Required**
	Units:	Inches of HG	**Range:**	0 - 999.999	
	Description:	Engine #4 Manifold Pressure			
	Example:	25.812			

23	**Field:**	prop_1_angle	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Propeller blade angle, engine #1			
	Example:	54.1092			

24	**Field:**	prop_2_angle	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Propeller blade angle, engine #2			

	Example:	54.1092			

25	**Field:**	prop_3_angle	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Propeller blade angle, engine #3			
	Example:	54.1092			

26	**Field:**	prop_4_angle	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Propeller blade angle, engine #4			
	Example:	54.1092			

27	**Field:**	autopilot	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"off", "on"	
	Description:	Status of autopilot (is the autopilot on or off?)			
	Example:	off			

28	**Field:**	pitch_control_input	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Pitch control input at the control yoke			
	Example:	-14.871			

29	**Field:**	lateral_control_input	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Aileron control input at the control yoke			
	Example:	19.212			

30	**Field:**	rudder_control_input	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Rudder control input at the rudder pedals			
	Example:	-6.691			

31	**Field:**	pitch_control_surface_position	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Position of pitch control surface (elevator or stabilator)			
	Example:	-6.691			

32	**Field:**	lateral_control_surface_position	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Position of aileron control surface			
	Example:	4.812			

33	**Field:**	yaw_control_surface_position	**Type:**	float(7,3)	**Not Required**
	Units:	Degrees	**Range:**	-9999.999 - 9999.999	
	Description:	Position of rudder control surface			
	Example:	1.772			

34	**Field:**	vertical_acceleration	**Type:**	float(6,3)	**Not Required**
	Units:	g's	**Range:**	-999.999 - 999.999	
	Description:	Amount of vertical g's recorded			
	Example:	1.282			

35	**Field:**	longitudinal_acceleration	**Type:**	float(6,3)	**Not Required**
	Units:	g's	**Range:**	-999.999 - 999.999	
	Description:	Amount of longitudinal g's recorded			
	Example:	-0.113			

36	**Field:**	lateral_acceleration	**Type:**	float(6,3)	**Not Required**
	Units:	g's	**Range:**	-999.999 - 999.999	

	Description:	Amount of lateral g's recorded			
	Example:	1.102			

37	**Field:**	pitch_trim_surface_position	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Deflection of pitch trim surface			
	Example:	-2.881			

38	**Field:**	trailing_edge_flap_selection	**Type:**	Tinyint(4)	**Not Required**
	Units:	Degrees	**Range:**	-128 - 127	
	Description:	Flap selection from cockpit, trailing edge device			
	Example:	15			

39	**Field:**	leading_edge_flap_selection	**Type:**	Tinyint(4)	**Not Required**
	Units:	Degrees	**Range:**	-128 - 127	
	Description:	Flap selection from cockpit, leading edge device			
	Example:	15			

40	**Field:**	thrust_reverse_position_1	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Amount of thrust reverse lever application, engine #1			
	Example:	0.000			

41	**Field:**	thrust_reverse_position_2	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Amount of thrust reverse lever application, engine #2			
	Example:	0.000			

42	**Field:**	thrust_reverse_position_3	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Amount of thrust reverse lever application, engine #3			
	Example:	0.000			

43	**Field:**	thrust_reverse_position_4	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Amount of thrust reverse lever application, engine #4			
	Example:	0.000			

44	**Field:**	ground_spoiler_speed_brake_ position	**Type:**	Tinyint(4)	**Not Required**
	Units:	Degrees	**Range:**	-128 - 127	
	Description:	Cockpit control position of speed brake selector			
	Example:	5			

45	**Field:**	oat	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Outside Air Temperature			
	Example:	-28.31			

46	**Field:**	afcs_mode	**Type:**	smallint(6)	**Not Required**
	Units:	NA	**Range:**	0 - 65,535	
	Description:	Autopilot mode. *Note: separate table will be available to describe various modes. afcs_mode will be foreign keyed into the other table.			
	Example:	3			

47	**Field:**	radio_altitude_actual	**Type:**	mediumint(9)	**Not Required**
	Units:	feet	**Range:**	0 - 16,777,215	

	Description:	Radio (radar) altitude of aircraft as recorded.			
	Example:	1,672			

48	**Field:**	radio_altitude_derived	**Type:**	mediumint(9)	**Not Required**
	Units:	feet	**Range:**	0 - 16,777,215	
	Description:	Radio (radar) altitude of aircraft as calculated from msl altitude minus terrain altitude.			
	Example:	21,199			

49	**Field:**	localizer_deviation	**Type:**	float(5,3)	**Not Required**
	Units:	Degrees	**Range:**	-99.999 - 99.999	
	Description:	Degrees off of localizer course, negative denotes left, positive right.			
	Example:	3.012			

50	**Field:**	glideslope_deviation	**Type:**	float(5,3)	**Not Required**
	Units:	Degrees	**Range:**	-99.999 - 99.999	
	Description:	Degrees off of glideslope, negative denotes low, positive high.			
	Example:	-1.912			

51	**Field:**	marker_beacon_passage	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Outer marker beacon being overflown.			
	Example:	no			

52	**Field:**	master_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Master warning indication displayed.			
	Example:	no			

53	**Field:**	weight_on_wheels	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"ground", "air"	
	Description:	Weight on wheels sensed.			
	Example:	air			

54	**Field:**	aoa	**Type:**	float(5,3)	**Not Required**
	Units:	Degrees	**Range:**	-99.999 - 99.999	
	Description:	Angle of attack.			
	Example:	7.183			

55	**Field:**	hydraulic_pressure_low	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Hydraulic pressure low indication.			
	Example:	no			

56	**Field:**	groundspeed	**Type:**	float(7,3)	**Not Required**
	Units:	Knots	**Range:**	-9,999.999 - 9,999.999	
	Description:	True airspeed (not derived)			
	Example:	124.219			

57	**Field:**	terrain_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Terrain warning present.			
	Example:	no			

58	**Field:**	landing_gear_position	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"up", "down", "transit"	
	Description:	Position of landing gear.			

	Example:	up			

59	**Field:**	drift_angle	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Drift angle.			
	Example:	17.227			

60	**Field:**	wind_speed	**Type:**	float(6,3)	**Not Required**
	Units:	Knots	**Range:**	0.000 - 999.999	
	Description:	Speed of wind			
	Example:	119.426			

61	**Field:**	wind_direction	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
			Actual:	0.000-359.999	
	Description:	Magnetic direction of wind			
	Example:	340.736			

62	**Field:**	latitude	**Type:**	float(8,6)	**Not Required**
	Units:	Degrees	**Range:**	-99.999999 - 99.999999	
			Actual:	-90.0000000 - 90.0000000	
	Description:	Latitude of aircraft, negative denotes southern hemisphere, positive denotes northern.			
	Example:	43.567143			

63	**Field:**	longitude	**Type:**	float(9,6)	**Not Required**
	Units:	Degrees	**Range:**	-999.999999 - 999.999999	
			Actual:	-180.0000000 - 180.0000000	
	Description:	Longitude of aircraft, negative denotes western hemisphere, positive denotes eastern.			
	Example:	-121.387255			

64	**Field:**	stall_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Stall warning present.			
	Example:	no			

65	**Field:**	stick_shaker	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Stick shaker activated.			
	Example:	no			

66	**Field:**	stick_pusher	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Stick pusher activated.			
	Example:	no			

67	**Field:**	windshear	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Windshear warning active.			
	Example:	no			

68	**Field:**	throttle_lever_position_1	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Position of throttle lever, engine #1			
	Example:	58.712			

69	**Field:**	throttle_lever_position_2	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Position of throttle lever, engine #2			

	Example:	58.712		

70	**Field:**	throttle_lever_position_3	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Position of throttle lever, engine #3			
	Example:	58.712			

71	**Field:**	throttle_lever_position_4	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Position of throttle lever, engine #4			
	Example:	58.712			

72	**Field:**	traffic_alert	**Type:**	smallint(6)	**Not Required**
	Units:	NA	**Range:**	0 - 65,535	
	Description:	Traffic alert status.. *Note: separate table will be available to describe various modes. traffic_alert will be foreign keyed into the other table.			
	Example:	3			

73	**Field:**	dme_1_distance	**Type:**	float(6,3)	**Not Required**
	Units:	DME units	**Range:**	-999.999 - 999.999	
			Actual:	-199.999 - 199.999	
	Description:	Distance Measuring equipment (DME) #1 receiver distance.			
	Example:	72.192			

74	**Field:**	dme_2_distance	**Type:**	float(6,3)	**Not Required**
	Units:	DME units	**Range:**	-999.999 - 999.999	
			Actual:	-199.999 - 199.999	
	Description:	Distance Measuring equipment (DME) #2 receiver distance.			
	Example:	72.192			

75	**Field:**	nav_1_freq	**Type:**	float(6,3)	**Not Required**

	Units:	MHz	**Range:**	-999.999 - 999.999	
			Actual:	110.000 - 118.000	
	Description:	Selected frequency Nav 1.			
	Example:	114.30			

76	**Field:**	nav_2_freq	**Type:**	float(6,3)	**Not Required**
	Units:	MHz	**Range:**	-999.999 - 999.999	
			Actual:	110.000 - 118.000	
	Description:	Selected frequency Nav 2.			
	Example:	112.725			

77	**Field:**	obs_1	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees	**Range:**	00.000-359.99	
	Description:	Course set into Omni Bearing Selector (OBS) 1			
	Example:	125.00			

78	**Field:**	obs_2	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees	**Range:**	00.000-359.99	
	Description:	Course set into Omni Bearing Selector (OBS) 2			
	Example:	125.00			

79	**Field:**	altimeter	**Type:**	float(4,2)	**Not Required**
	Units:	Inches of HG	**Range:**	00.00-99.99	
			Actual:	20.00-35.00	
	Description:	Altimeter setting			
	Example:	29.92			

80	**Field:**	selected_altitude	**Type:**	mediumint(9)	**Not Required**
	Units:	feet	**Range:**	0 - 16,777,215	
	Description:	Selected altitude in altitude setting system (or alerter).			

	Example:	15000			

81	**Field:**	selected_speed	**Type:**	smallint(4)	**Not Required**
	Units:	knots	**Range:**	0 - 9999	
	Description:	Selected speed in AFCS.			
	Example:	150			

82	**Field:**	selected_mach	**Type:**	float(3,2)	**Not Required**
	Units:	mach	**Range:**	-9.99 - 9.99	
	Description:	Selected Mach number in autopilot system.			
	Example:	.86			

83	**Field:**	selected_vertical_speed	**Type:**	smallint(5)	**Not Required**
	Units:	Feet per minute	**Range:**	-99,999 - 99,999	
	Description:	Selected vertical speed in autopilot			
	Example:	-1500			

84	**Field:**	selected_heading	**Type:**	smallint(3)	**Not Required**
	Units:	Degrees	**Range:**	0-359	
	Description:	Selected heading in autopilot			
	Example:	047			

85	**Field:**	selected_flight_path*	**Type:**	tinyint(3)	**Not Required**
	Units:	NA	**Range:**	0 - 256	
	Description:	Selected flight path mode in autopilot. *Note: separate table will be available to describe various modes. selected_flight_path will be foreign keyed into the other table.			
	Example:	3			

13 APPENDIX C

Table 5. Data Architecture of DJI Telemetry Recording (.DAT) Files for Modern Platforms.

Prefix	Name	Description	Freq (HZ)	Derive	UAS Unique?
General	Tick#	Internal bus clock	Varies	No	
	relativeHeight	Meters. Altitude above Home Point	10	No	
	absoluteHeight	Meters. Populated if the Home Point Elevation has been set.	200	Yes	
	flightTime	Milliseconds. Can be used to sync with .txt log files. I.e., HealthyDrones, DJI Go App, Litchi	10	No	
	gpsHealth	[0 -5] 5 is a measure of the FC's confidence in the lat, long coords that are computed from the GPS and IMU data	200	No	
	vpsHeight	Meters. Height from VPS sensor. Blank if VPS height isn't valid.	200	No	
	flyCState	Duplicate of flyCState field in the .txt file. Manual, Atti, Atti_CL, Atti_Hover, Hover, GPS_Blake, GPS_Atti, GPS_CL, GPS_HomeLock, GPS_HotPoint, AssitedTakeoff, AutoTakeoff, AutoLanding,AttiLangding,NaviGo, GoHome, ClickGo, Joystick, Atti_Limited, GPS_Atti_Limited, NaviMissionFollow, NaviSubMode_Tracking, NaviSubMode_Pointing, PANO, Farming, FPV, SPORT, NOVICE, FORCE_LANDING, TERRAIN_TRACKING, NAVI_ADV_GOHOME, NAVI_ADV_LANDING, TRIPOD_GPS, TRACK_HEADLOCK, ASST_TAKEOFF, GENTLE_GPS,OTHER	10	No	Yes

flycCommand	AUTO_FLY, AUTO_LANDING, HOMEPOINT_NOW, HOMEPOINT_HOT, HOMEPOINT_LOC, GOHOME, START_MOTOR, STOP_MOTOR, Calibration, DeformProtecClose, DeformProtecOpen, DropGohome, DropTakeOff, DropLanding, DynamicHomePointOpen, DynamicHomePointClose, FollowFunctioonOpen, FollowFunctionClose, IOCOpen, IOCClose, DropCalibration, PackMode, UnPackMode, EnterManaualMode, StopDeform), DownDeform, UpDeform, ForceLanding, ForceLanding2, OTHER			Yes
flightAction	NONE, WARNING_POWER_GOHOME, WARNING_POWER_LANDING, SMART_POWER_GOHOME, SMART_POWER_LANDING, LOW_VOLTAGE_LANDING, LOW_VOLTAGE_GOHOME, SERIOUS_LOW_VOLTAGE_LANDIN G, RC_ONEKEY_GOHOME, RC_ASSISTANT_TAKEOFF, RC_AUTO_TAKEOFF, RC_AUTO_LANDING, APP_AUTO_GOHOME, APP_AUTO_LANDING, APP_AUTO_TAKEOFF, OUTOF_CONTROL_GOHOME, API_AUTO_TAKEOFF, API_AUTO_LANDING, API_AUTO_GOHOME, AVOID_GROUND_LANDING, AIRPORT_AVOID_LANDING, TOO_CLOSE_GOHOME_LANDING, TOO_FAR_GOHOME_LANDING,AP P_WP_MISSION, WP_AUTO_TAKEOFF, GOHOME_AVOID, GOHOME_FINISH, VERT_LOW_LIMIT_LANDING, BATTERY_FORCE_LANDING, MC_PROTECT_GOHOME	10	No	Yes
nonGPSCause	Duplicate of nonGPS_Cause field in the .txt file. A value other than	10	No	

		ALREADY means a "compass error". Other possible values are FORBIN, GPSNUM_NONENOUGH), GPS_HDOP_LARGE, GPS_POSITION_NONMATCH, SPEED_ERROR_LARGE, YAW_ERROR_LARGE, COMPASS_ERROR_LARGE, UNKNOWN			
	connectedToRC	Connected, NotConnected	10	No	Yes
	gpsUsed	True/False. GPS is used by FC to compute horizontal velocity	10	No	
	visionUsed	True/False. Vision system is used by FC to compute horizontal velocity	10	No	
IMU_ATTI(IMU#)	Longitude	degrees. Computed by the FC from GPS, Accelerometer, and Gyro data.Blank until valid.	200	No	
	Latitude		200	No	
	numSats				
	barometer:Raw	Meters. Raw data from barometer.	200	No	
	barometer:Smooth	Meters. Smoothed barometer data	200	No	
	accel:<Axis>	Meters/second. Acceleration along the X, Y and Z axes	200	No	
	accel:Composite	Meters/second. sqrt (accelX**2 + accelY**2 + accelZ**2)	200	Yes	
	gyro:<Axis>	Degrees/second. Rotation about the X, Y and Z axes	200	No	
	gyro:Composite	sqrt(gyroX**2 + gyroY**2 + gyroZ**2)	200	Yes	
	mag:<Axis>		50	No	
	mag:Mod	sqrt(magX**2 + magY**2 +magZ**2)	50	Yes	
	Vel:<North, East, Down>	Meters/second. Velocity North, East, Down	200	No	
	velComposite	Meters/sec. Velocity. Sqrt(velN*velN + velE*velE +velD*velD)	200	Yes	
	velH	Meters/sec. Horizontal velocity. Sqrt(velN*velN + velE*velE)	200	Yes	

GPS-H	Meters/second. Difference between velocity computed from successive GPS coordinates and horizontal velocity computed from IMU sensors(Vel:Horizontal).	200	Yes
quat<W,X, Y, Z>	Quaternion	200	No
roll	Degrees. Note, the yaw value will be corrected for geomagnetic declination after GPS data is valid. I.e. Yaw will be true and not magnetic.	200	Yes
pitch		200	Yes
yaw		200	Yes
yaw360	Degrees. Range 0 -360.	200	Yes
totalGyro:<Axis>	Degrees. Integration and summation of Gyro:<Axis>. Can be used to compute Gyro:<Axis> error. Also useful for checking roll, pitch, and yaw values coming from Flight Controller.	200	Yes
magYaw	Yaw value computed from magnetometers and corrected with pitch and roll. Not the same as Yaw which comes from the Flight Controller.	200	Yes
Yaw-magYaw		200	Yes
distanceHP		200	Yes
distanceTravelled	Meters. Computed from successive latitude/longitude coordintes	1	Yes
directionOfTravel[mag]	Degrees. Range = [-180,180]. Computed from successive latitude/longitude coordinates. Not corrected with local geomagnetic declination. I.e. value can be compared against P3 yaw.	1	Yes
directionOfTravel[true]	Degrees. Range = [-180,180]. Computed from successive latitude/longitude coordinates. Corrected with local geomagnetic declination. I.e. value can not be compared against P3 yaw.	1	Yes
temperature	IMU temp. Steady state = 65 C	200	No
ag_<Axis>		200	No

	gb_<Axis>		200	No	
Battery	lowVoltage	lowVoltage warning; 1 = warning, 0 = normal	1	No	
	status	OK, NotReady, Commerror, VolVeryLow, VolNotSafe	1	No	
Battery(Batt#)	cellVolts<Cell#>		1	No	
	current		1	No	
	totalVolts		1	Yes	
	Temp	Celcius	1	No	
	battery%				
	FullChargeCap	Battery Full Charge Capacity	1	No	Yes
	RemainingCap	Battery Remaining Capacity	1	No	Yes
	voltSpread	maximum cell voltage - minimum cell voltage	1	Yes	
	watts		1	Yes	
	minCurrent	Minimum Current since Battery On	1	Yes	Yes
	maxCurrent	Maximum Current since Battery On	1	Yes	Yes
	avgCurrent	Average Current since Battery On	1	Yes	Yes
	minVolts	Minimum totalVolts since Battery On	1	Yes	Yes
	maxVolts	Maximum totalVolts since Battery On	1	Yes	Yes
	avgVolts	Average totalVolts since Battery On	1	Yes	Yes
	minWatts	MinimumWatts since Battery On	1	Yes	Yes
	maxWatts	Maximum Watts since Battery On	1	Yes	Yes
	avgWatts	Average Watts since Battery On	1	Yes	Yes
BattInfo	Vol		50	No	
	Current		50	No	
	remainingTime		50	No	Yes
	CellVol		50	No	Yes
	LowVolThreshold		50	No	Yes
	BatVol		50	No	

	BatCurrent		50	No	
	FullChargeCap		50	No	
	Remaining%		50	No	
	BatTemp		50	No	
	BatDataCnt		50	No	
	OriginalCap		50	No	
	Ad_v		50	No	
	r_time		50	No	
	AvgCurrent		50	No	
	vol_t		50	No	
	Pack_ve		50	No	
	RemainingCap		50	No	
	Temp		50	No	
	right		50	No	
	l_cell		50	No	Yes
	dyna_cnt		50	No	
	FullCap		50	No	
	out_ctl		50	No	
	out_ctl_f		50	No	
SMART_BATT	goHome%	percentage at which a go home will be requested	1	No	Yes
	land%	percentage at which landing will be requested	1	No	Yes
	goHomeTime	time at which a go home will be requested	1	No	Yes
	landTime	time at which landing will be requested	1	No	Yes
	voltage%	current battery percentage			
	Status	OK, NotReady, Commerror, VolVeryLow, VolNotSafe	1	No	Yes
	GHStatus	None, GoHome, GoHomeAlready	1	No	Yes
Controller	gpsLevel	Same as General:gpsHealth. Useful when looking at a tablet .DAT	50	No	

	ctrl_level	Unknown, maybe a gpsHealth for the RC	50	No	
GPS(gps#)	Long	Degrees. May not be valid if DOP is large.	5	No	
	Lat	Degrees. May not be valid if DOP is large.	5	No	
	Date	Integer that contains date, e.g. 20171003 means 2017-10-03 GMT	5	No	
	Time	Integer that contains time, e.g. 100334 means 10:03:34 GMT	5	No	
	dateTime	DateTime in ISO-8601 format. Not available in CsvView	5	No	
	heightMSL	Meters, Height above mean sea level	5	No	
	hDOP	Horizontal dilution of precision. Units unknown.	5	No	
	pDOP	Position dilution of precision. Units unknown.	5	No	
	sAcc	Some kind of accuracy measure.			
	numGPS	Number of GPS satellites	5	No	
	numGLNAS	Number of GLONAS satellites	5	No	
	numSV	Total number of satellites	5	No	
	vel:<North, East, Down>	Meters/second. Velocity North, East, Down	200	No	
HP	Longitude	Coordinates of Home Point. Obtained from eventLog. Altitude is set by A/C to be 20 meters higher than the barometric altitude.	N/A	No	Yes
	Latitude		N/A	No	Yes
	Altitude		N/A	No	Yes
	rthHeight	meters	N/A	No	Yes
IMUEX(imu#)	vo_v<Axis>		200		
	vo_p<Axis>		200		
	us_v		200		
	us_p		200		

	vo_flag_Navi		200	
	cnt		200	
	rtk_Longitude		200	
	rtk_Latitude		200	
	rtk_Alti		200	
	err	None, SPEED_LARGE_ERROR, GPS_YAW_ERROR, MAG_YAW_ERROR, GPS_CONSIST_ERROR, US_FAIL_ERROR	200	
Motor	Speed:<motor>	Actual Motor Speed. RPM.	50	No
	EscTemp:<motor>	ESC temperature, not motor temperature	50	No
	PPMrecv:<motor>		50	No
	V_out:<motor>		50	No
	Volts:<motor>		50	No
	Current:<motor>		50	No
	Status:<motor>	0 = Normal, other values unknown	50	No
	PPMsend:<motor>			
	thrustAngle	Degrees. Computed from motor speeds. Direction the A/C is being pushed by the motors. Relative to the A/C, not the inertial frame.	200	Yes
MotorCtrl	Status	0 = Normal, other values unknown	50	No
	PWM:<motor>	Pulse Width Modulation. Can be used to determine commanded motor speed. Range 0 - 100%	50	No
MotorPwrCalcs	Volts:Avg:<motor>		50	Yes
	Volts:Avg:All		50	Yes
	Current:Avg:<motor>		50	Yes
	Current:Avg:All		50	Yes
	Watts:Avg:<motor>		50	Yes
	Watts:Avg:All		50	Yes
	WattSecs:<motor>		50	Yes

	WattSecs:All		50	Yes
	WattSecs/Dist:<motor>		50	Yes
	WattSecs/Dist:All		50	Yes
	WattSecs/TotalDist:<motor>		50	Yes
	WattSecs/TotalDist:All		50	Yes
	Watts/VelH:<motor>		50	Yes
	Watts/VelH:All		50	Yes
	Watts/VelD:<motor>		50	Yes
	Watts/VelD:All		50	Yes
MVO	vel<Axis>		10	No
	pos<Axis>		10	No
	hoverPointUncertainty1		10	No
	hoverPointUncertainty2		10	No
	hoverPointUncertainty3		10	No
	hoverPointUncertainty4		10	No
	hoverPointUncertainty5		10	No
	hoverPointUncertainty6		10	No
	velocityUncertainty1		10	No
	velocityUncertainty2		10	No
	velocityUncertainty3		10	No
	velocityUncertainty4		10	No
	velocityUncertainty5		10	No
	velocityUncertainty6		10	No
	height		10	No
	heightUncertainty		10	No
OA	avoidObst		10	No
	emergBrake	Off, On	50	No

	radiusLimit		10	No
	airportLimit		10	No
	groundForceLanding		10	No
	horizNearBoundary		10	No
	vertLowLimit		10	No
	vertAirportLimit		10	No
	roofLimit		10	No
	hitGroundLimit		10	No
	frontDistance		10	No
RC	Aileron	Range [-10000, 10000] Neutral = 0. Stick left or down = -10000. Stick right or up = 10000.	50	No
	Elevator		50	No
	Rudder		50	No
	Throttle		50	No
	ModeSwitch	P, Sport	50	No
	sigStrength	Percentage based on the number of valid frames per unit time. I.e., not an RF measurement.	50	Yes
	failSafe	Hover, Landing, GoHome, Unknown	50	No
	dataLost	"", lost	50	No
	appLost	"", lost	50	No
	connected	Connected, Disconnected	50	No
InertialOnlyCalcs(imu#)	Vel:<North, East, Down>	Meters/sec^2. Velocity	200	Yes
	Pos:<North, East, Down>	Meters. Position relative to HP.	200	Yes
	ag:<North, East, Down>	Meters/sec^2. Acceleration relative to ground.	200	Yes
	aB:<North, East, Down>	Meters/sec^2. Acceleration relative to AC.	200	Yes
	getVelN() - vgX	Difference between velocity computed by IMU and velocity computed here	200	Yes
	getVE() - vgY		200	Yes

	getVd() - vgZ		200	Yes
Mag(mag#)	<Axis>	Magnetometer values for each group of magnetometers. The AC uses just one group at a time with group 0 being the default.	50	No
	Mod		50	Yes
	magYaw		50	Yes
	Yaw-magYaw		50	Yes
	raw<Axis>	Raw magnetometer data. See the eventLog stream for the scale and bias values used to compute the above values.	50	No
	rawMod		50	Yes
AirComp	AirSpeedBody:X	These fields aren't fully understood.	5	No
	AirSpeedBody:Y		5	No
	Alti		5	No
	VelNorm		5	No
	VelTime:1		5	No
	VelTime:2		5	No
	VelLevel		5	No
	WindSpeed		5	No
	Wind:X		5	No
	Wind:Y		5	No
	MotorSpeed		5	No
	WindHeading	Computed from some of above values.	5	Yes
	WindMagnitude		5	Yes
	WindMagnitude:2		5	Yes
AirCraftCondition	int_fsm		50	No
	last_fsm		50	No
	UP_state		50	No
	safe_fltr		50	No
	launch_acc_dur		50	No

launch_free_fall_dur	50	No
launch_free_fall_delta_v	50	No
thrust	50	No
gyro	50	No
land_dur_press	50	No
land_dur_sonic	50	No
thrust_body	50	No
thrust_gnd	50	No
thrust_gnd_compen	50	No
safe_tilt_raw	50	No
sat_timer	50	No
fsmState	50	No
landState	50	No
UP_acc_t	50	No
UP_TF_t	50	No
craft_flight_mode	50	No
launch_acc_duration	50	No
launch_delta_v	50	No
launch_state	50	No
thrust_proj_gnd	50	No
thrust_proj_gnd_compen	50	No
thrust_compensator	50	No
hover_thrust	50	No
dynamic_thrust	50	No
cos_safe_tilt	50	No
safe_tilt	50	No
nearGround	50	No
gyro_acc	50	No
land_dur	50	No

Derived from CsvView/DatCon (n.d.)

Table 6. Data Architecture of DJI Telemetry Recording (.DAT) Files for Phantom 3 / Inspire 1 Platforms.

Name	Description	Freq (HZ)	Derived	UAS Unique
tickNo	P3 internal bus clock	600	No	
offSetTime	See User Manual	200	Yes	
longitude	degrees. Converted from radians	200	No	
latitude	degrees. Converted from radians	200	No	
numSats	Number of Satellites	N/A	No	
gpsHealth	0 - 5. 5 is best condition.	N/A	No	
baroRaw	Meters. Raw data from barometer.	50	No	
baroAlt	Meters. Smoothed barometer data	200	No	
vpsHeight	Meters. Height from VPS sensor. Blank if VPS height isn't valid (generally > 3 meters above ground)	200	No	
accelX	Meters/second. Acceleration along the X, Y and Z axes	200	No	
accelY		200	No	
accelZ		200	No	
accel	Meters/second. sqrt (accelX**2 + accelY**2 + accelZ**2)	200	Yes	
gyroX	Degrees/second. Rotation about the X, Y and Z axes	200	No	
gyroY		200	No	
gyroZ		200	No	
gyro	sqrt(gyroX**2 + gyroY**2 + gyroZ**2)	200	Yes	
errorX	Precise description unknown. Probably an error term representing the difference between the measured and predicted orientation	200	No	
errorY		200	No	
errorZ		200	No	
error	sqrt (errorX**2 + errorY**2 +errorZ**2)	200	Yes	
magX		50	No	
magY		50	No	
magZ		50	No	

magMod	sqrt(magX**2 + magY**2 +magZ**2)	50	Yes
quatW	Quaternion. The orientation of the P3. QuatW is the scalar. (QuatX, QuatY, QuatZ) is the vector part. See https://en.wikipedia.org/wiki/Quaternion	200	No
quatX		200	No
quatY		200	No
quatZ		200	No
Roll	Degrees. Computed from the Quaternion above. Note, the yaw value appears to be corrected for geomagnetic declination; I.e. yaw is true and not magnetic.	200	Yes
Pitch		200	Yes
Yaw		200	Yes
Yaw360	Degrees. Range 0 -360.	200	Yes
totalGyroZ	Degrees. Integration and summation of gyroZ . Can be used to compute gyroZ drift.	200	Yes
magYaw	Yaw value computed from magnetometers and corrected with pitch and roll. Not the same as Yaw which comes from the Flight Controller.	200	Yes
thrustAngle	Degrees. Computed from motor speeds. Direction the A/C is being pushed by the motors. Relative to the A/C, not the inertial frame.	200	Yes
velN	Meters/second. Velocity North, East, Down	200	No
velE		200	No
velD		200	No
vel	Meters/sec. Speed. Sqrt(velN*velN + velE*velE +velD*velD)	200	Yes
velH	Meters/sec. Horizontal speed. Sqrt(velN*velN + velE*velE)	200	Yes
velGPS-velH	Meters/second. Difference between velocity computed from successive GPS coordinates and velocity computed from IMU sensors(velH).	200	Yes

homePointLongitude	Coordinates of Home Point. Obtained from eventLog. Altitude is set by A/C to be 20 meters higher than the barometric altitude.	N/A	No	Yes
homePointLatitude		N/A	No	Yes
homePointAltitude		N/A	No	Yes
geoMagDeclination	degrees	N/A	Yes	Yes
geoMagInclination	degrees. Down is positive, up is negative	N/A	Yes	Yes
distanceHP	Meters. Distance from Home Point	200	No	Yes
distanceTraveled	Meters. Computed from successive latitude/longitude coordintes	1	Yes	Yes
relativeHeight	Meters. Altitude above Home Point	10	No	Yes
flightTime	Milliseconds. Can be used to synch with .txt log files. I.e., HealthyDrones, DJI Go App, Litchi	10	No	
directionOfTravel	Degrees. Range = [-180,180]. Computed from successive latitude/longitude coordinates. Corrected with local geomagnetic declination. I.e. value can be compared against P3 yaw.	1	Yes	
directionOfTravelTrue	Degrees. Range = [-180,180]. Computed from successive latitude/longitude coordinates. Not corrected with local geomagnetic declination. I.e. value can not be compared against P3 yaw.	1	Yes	
Control:Aileron	Range [-10000, 10000] Neutral = 0. Stick left or down = -10000. Stick right or up = 10000.	50	No	
Control:Elevator		50	No	
Control:Throttle		50	No	
Control:Rudder		50	No	
Control:ModeSwitch	2=P, 1=A, 0=F, 4 = remote control switched off	50	No	

flightMode	Derived from eventLog. Deprecated, use flyCState below. 1 = ATTI, 2 = GPS_ATTI. Removed in version 2.2.8 and later	N/A	Yes
flightMode.string			
flightRegime	Derived from eventLog. Deprecated, use flyCState below. 1 = engineStart, 2 = asstTakeOff, 3 = autoTakeOff, 4 = autoLanding. Removed in version 2.2.8 and later	N/A	Yes
flightRegime.string			
navMode	Derived from eventLog. Deprecated, use flyCState below. 1 = goHome, 2 = waypoint, 3 = folowMe, 4 = hotPoint. Removed in version 2.2.8 and later	N/A	Yes
navMode.string			
flyCState	Duplicate of flyCState field in the .txt file. Manual(0), Atti(1), Atti_CL(2), Atti_Hover(3), Hover(4), GPS_Blake(5), GPS_Atti(6), GPS_CL(7), GPS_HomeLock(8), GPS_HotPoint(9), AssitedTakeoff(10), AutoTakeoff(11), AutoLanding(12), AttiLangding(13), NaviGo(14), GoHome(15), ClickGo(16), Joystick(17), Atti_Limited(23), GPS_Atti_Limited(24), Follow_Me(25),OTHER(100);	10	No
flyCState:String			
nonGPSCause	Duplicate of nonGPS_Cause field in the .txt file. ALREADY(0), FORBIN(1), GPSNUM_NONENOUGH(2), GPS_HDOP_LARGE(3), GPS_POSITION_NONMATCH(4), SPEED_ERROR_LARGE(5), YAW_ERROR_LARGE(6), COMPASS_ERROR_LARGE(7), UNKNOWN(8);	10	No
nonGPSCause:String			
DW flyCState	Dashware helper. Maps values in flyCState to a different set of values. Manual(1), Atti(2), Atti_CL(3), Atti_Hover(4), Hover(5), GPS_Blake(6), GPS_Atti(7), GPS_CL(8), GPS_HomeLock(9), GPS_HotPoint(20), AssitedTakeoff(30), AutoTakeoff(40), AutoLanding(50), AttiLangding(60), NaviGo(70), GoHome(80), ClickGo(90),	10	Yes

	Joystick(200), Atti_Limited(300), GPS_Atti_Limited(400), Follow_Me(500),OTHER(600);			
connectedToRC	0 = not connected, 1 = connected		10	No
Current	Amps		1	No
Volt1	Cell voltages. Volt5 and Volt6 will be blank unless the A/C is an Inspire.		1	No
Volt2			1	No
Volt3			1	No
Volt4			1	No
Volt5			1	No
Volt6			1	No
totalVolts			1	No
voltSpread	maximum cell voltage - minimum cell voltage		1	No
Watts	toltalVolts * Current		1	Yes
minCurrent	Minimum Current since Battery On		1	Yes
maxCurrent	Maximum Current since Battery On		1	Yes
avgCurrent	Average Current since Battery On		1	Yes
minVolts	Minimum totalVolts since Battery On		1	Yes
maxVolts	Maximum totalVoltssince Battery On		1	Yes
avgVolts	Average totalVolts since Battery On		1	Yes
minWatts	MinimumWatts since Battery On		1	Yes
maxWatts	Maximum Watts since Battery On		1	Yes
avgWatts	Average Watts since Battery On		1	Yes
batteryTemp	Celcius		1	No
ratedCapacity	maH	N/A		No
remainingCapacity	maH		1	No
percentageCapacity			1	No
percentageVolts			1	No

batteryStatus	UserBatteryReqGoHome(1), UserBatteryReqLand(2), SmartBatteryReqGoHome(4), SmartBatteryReqLand(8), MainVoltageLowGoHOme(16), MainVoltageLowLand(32), BatteryCellError(64), BatteryCommunicateError(128), VoltageLowNeedLand(256), BatteryTempVoltageLow(512), BatteryNotReady(1024), BatteryFirstChargeNotFull(2048), BatteryLimitOutputMax(4096), BatteryDangerous(8192), BatteryDangerousWarning(16384)		1	No	Yes
batteryGoHomeStatus	NON_GOHOME(0), GOHOME(1), GOHOME_ALREADY(2)		1	No	Yes
batteryGoHome	percentage at which a go home will be requested by the smart battery		1	No	Yes
usefulTime	seconds		1	No	Yes
batteryCycles		N/A		No	
batteryLife		N/A		No	
batteryBarCode	Bar Code visible on battery	N/A		No	
MotorCmnd:RFront	Commanded Motor Speed. Range 0 - 10000.		50	No	
MotorCmnd:LFront			50	No	
MotorCmnd:LBack			50	No	
MotorCmnd:Rback			50	No	
MotorSpeed:RFront	Actual Motor Speed. RPM. Blank for P3 Standard which doesn't report motor speed.		50	No	
MotorSpeed:LFront			50	No	
			50	No	
MotorSpeed:Rback			50	No	
MotorLoad:RFront	Motor Load. Blank for P3 Standard which doesn't report motor loads.		50	No	
MotorLoad:LFront			50	No	
MotorLoad:LBack			50	No	
MotorLoad:Rback			50	No	

Gimbal:Roll	Degrees. Orientation of gimbal with respect to P3. I.e. not absolute orientation	50	No
Gimbal:Pitch		50	No
Gimbal:Yaw		50	No
Gimbal:XRoll	Degrees. Related to Gimbal and A/C orientation. Precise relationship unknown	50	No
Gimbal:XPitch		50	No
Gimbal:XYaw		50	No
tabletLongitude	Degrees. Non blank only during a Follow Me mission using the DJO Go App	15	No
tabletLatitude		15	No

Derived from CsvView/DatCon (n.d.)

Table 7. Data Architecture of Telemetry Recording Files for Yuneec Platforms.

Prefix	**Name**	**Description**	**Format**	**Derive**	**UAS Unique**
Telemetry	Date / Time	Date / time including milliseconds	JJJJMMTT hh:mm:ss:zzz; poor=>2s; reasonable=600ms-2s	No	
	fsk_rssi	Received Signal Strength Indication from copter's receiver	dBm, poor=>85, reasonable=70-85, good=55-70, very good<55	supposed	Yes
	voltage	Voltage off light accu	V	No	
	current	Current of flight accu, if sensor available (not for Q500 or Typhoon H)	dA	supposed for H920	
	altitude	Ascent relative to starting point	m	No	
	latitude	Latitude - GPS coordinates of copter	decimal degrees	No	
	longitude	Longitude - GPS coordinates of copter	decimal degrees	No	

tas	True Air Speed, Speed of the aircraft, computed from GPS coordinates I guess. So it is groundspeed, not really TAS	m/s	No	
gps_used	GPS usage (true/false)	boolean	No	
fix_type	GPS Fix Type	?	?	
satelites_num	Number of detected satellites	number	No	
roll	Roll	*	suppos ed	
yaw	Gier	*	suppos ed	
pitch	Nick	*	suppos ed	
motor_status	Motor Status, bitwise. Motor numbers according to the picture in the GUI		suppos ed	
imu_status	IMU Status (intertial meassurement unit)	bits	suppos ed	
gps_status	GPS unit status	bits	suppos ed	
cgps_used	C-GPS (unknown meaning)	?		
press_compass _status	Sensor Status (Barometer/Magnetometer)	bits	suppos ed	
f_mode	Code for different flight modes	code	No	
gps_pos_used	GPS position used (true, false)	boolean	No	
vehicle_type	Copter Type	1=Yunnec H920 2=Yuneec Q500 3=Blade 350QX 4=Blade Chroma (380QZ) 5=Yuneec Typhoon H	No	Yes

	error_flags1	Error flags, sum bitwise	0=ERROR_FLAG_VOLTAGE_WARNING1 1=ERROR_FLAG_VOLTAGE_WARNING2 2=ERROR_FLAG_MOTOR_FAILSAFE_MODE 3=ERROR_FLAG_COMPLETE_MOTOR_ESC_FAILURE 4=ERROR_FLAG_HIGH_TEMPERATURE_WARNING 5=ERROR_FLAG_COMPASS_CALIBRATION_WARNING 6=ERROR_FLAG_FLYAWAY_CHECKER_WARNING 7=ERROR_FLAG_AIRPORT_WARNING	No
	gps_accH	Horicontal GPS accuracy. Seems to be HDOP	HDOP, poor=>2.5, reasonable=1.8-2.5, good=1-1.8, very good=<1	supposed
Remote GPS	Date / Time	Date / Time including miliseconds	JJJJMMTT hh:mm:ss:zzz	No
	lon	Longitude - GPS coordinates of ground station	decimal degrees	No
	lat	Latitude - GPS coordinates of ground station	decimal degrees	No
	alt	Height from GPS relative to sea level	m	supposed
	accuracy	Accuracy of GPS	?	
	speed	Speed, unknown source (maybe computed from GPS coordinates, unknown unit	?	
	angle	Angle of moving direction to north	*	supposed
Remote	Date / Time	Date / Time including miliseconds	JJJJMMTT hh:mm:ss:zzz	No
	CH0	Channel 1: J1 throttle/ascent (thr)	0=Motor start/stop (B3) 2048=neutral	No
	CH1	Channel 2: J4 roll (ail)	2048=neautral	No
	CH2	Channel 3: J3 nick (ele)	2048=neautral	No
	CH3	Channel 4: J2 yaw (rud)	2048=neautral	No

	CH4	Channel 5: S4 Flight mode	3412=Smart 2048=Angle 683=RTH	No	
	CH5	Channel 6: A02 - RTH	2048=neutral 4095=RTH	No	Yes
	CH6	Channel 7: K2 Camera Tilt	683=horizontal (0 deg) 3413=vertical down (-90 deg)	No	Yes
	CH7	Channel 8: K1 Camera pan		No	Yes
	CH8	Channel 9: S1 Gimbal Tilt Mode	A=2184, V=3412	No	
	CH9	Channel 10: S2 Gimbal Pan Mode	F=683, Center=1502, G=3412	supposed	
	CH10	Channel 11: S5 Landegestell	0.0=up 1.0=down	No	
	CH11	Channel 12: A08			
	CH12	Channel 13: A09			
	CH13	Channel 14: A10			
	CH14	Channel 15: A11			
	CH15	Channel 16: A12			
	CH16	Channel 17: A13			
	CH17	Channel 18: A14			
	CH18	Channel 19: A15			
	CH19	Channel 20: A16			
	CH20	Channel 21: A17			
	CH21	Channel 22: A18			
	CH22	Channel 23: A19			
	CH23	Channel 24: A20			
f_mode	0	FMODE_BLUE_SOLID	Stability mode (Blue Solid)		
	1	FMODE_BLUE_FLASHING	Blue flashing		
	2	FMODE_BLUE_WOULD_BE_SOLID_NO_GPS	Blue, no GPS		
	3	FMODE_PURPLE_SOLID	Angle mode (Purple solid)		
	4	FMODE_PURPLE_FLASHING	Purple flashing		
	5	FMODE_PURPLE_WOULD_BE_SOLID_NO_GPS	Angle mode (Purple solid) - no GPS		
	6	FMODE_SMART	Smart mode		
	7	FMODE_SMART_BUT_NO_GPS	Smart mode - no GPS		

8	FMODE_MOTORS_STARTING	Motor starting	
9	FMODE_TEMP_CALIB	Temperature calibration	
10	FMODE_PRESS_CALIB	Pressure calibration	
11	FMODE_ACCELBIAS_CALI	Accelerator calibration	
12	FMODE_EMERGENCY_KILLED	Emergency/killed	
13	FMODE_GO_HOME	RTH coming	
14	FMODE_LANDING	RTH landing	
15	FMODE_BINDING	Binding	
16	FMODE_READY_TO_START	Initializing, Ready to start	
17	FMODE_WAITING_FOR_RC	Waiting for RC	
18	FMODE_MAG_CALIB	Magnetometer calibration	
19	FMODE_UNKNOWN	Unknown mode	
20	FMODE_RATE	Agility mode (Rate)	
21	FMODE_FOLLOW	Smart mode - follow me	
22	FMODE_FOLLOW_NO_GPS	Smart mode - follow me - no GPS	
23	FMODE_CAMERA_TRACKING	Smart mode - camera tracking	Yes
24	FMODE_CAMERA_TRACKING_NO_GPS	Camera tracking - no GPS	Yes
26	FMODE_TASK_CCC	Task Curve Cable Cam	
27	FMODE_TASK_JOUR	Task Journey	
28	FMODE_TASK_POI	Task Point of Interest	
29	FMODE_TASK_ORBIT	Task Orbit	
32		Indoor Positioning System	

Derived from Elsner (2017).

14 APPENDIX D

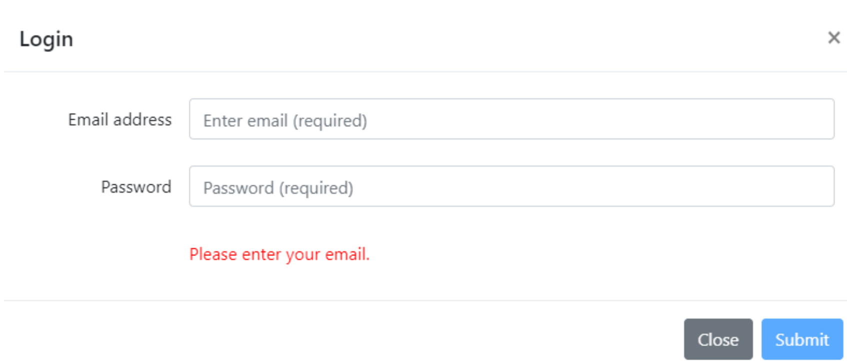
NGAFID User's Guide

Creating an Account

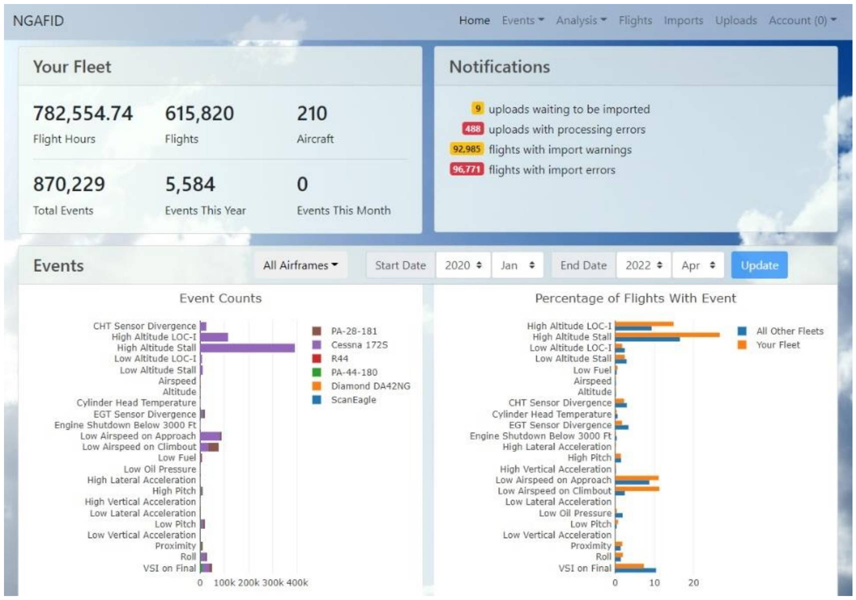
1. Navigate to www.ngafid.org
2. In the top-right corner of the screen, click "Create Account."

3. Fill out the required fields on the Account Creation page. For the option of "Account Type," select "I'm operating my own fleet" if you will be uploading your own personal data. If you are looking to be included to a larger group and wish to see all flights for the group, select "I am requesting access to an existing fleet."

4. Once your account is created, use your credentials to Login to the NGAFID website. Click on "Login" on the top ribbon and enter your Email Address and Password. Click "Submit."



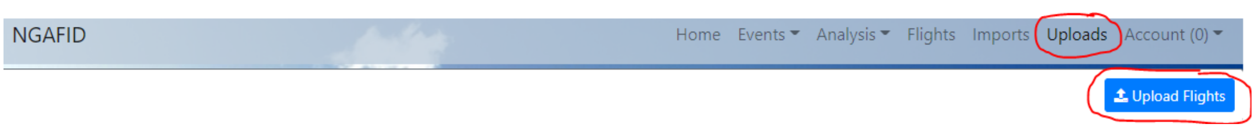
5. Once you are logged in, you will see your fleet dashboard.



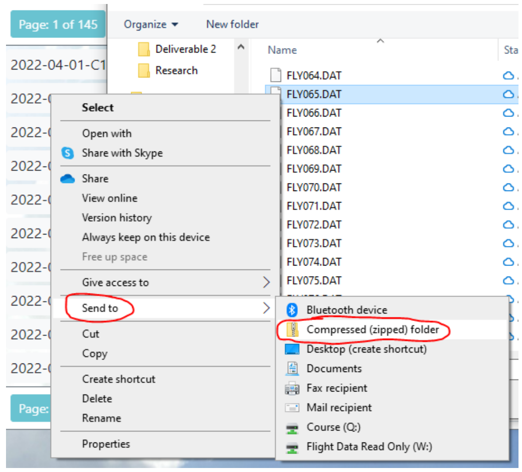
6. Your top menu options now included “Events,” “Analysis,” “Flights,” “Imports,” “Uploads,” and “Account.” To upload flight data, click on the “Uploads” menu option.

Uploading Flight Data to the NGAFID

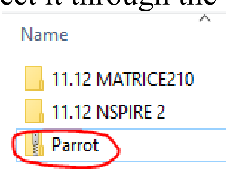
7. After clicking the “Uploads” menu option, there will be a new button titled “Upload Flights.” Click on this button to upload your new flights.



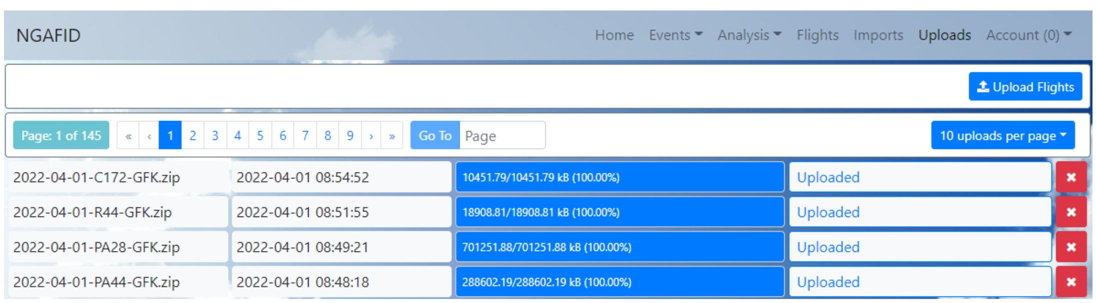
8. A “Open” window will appear. Select your flight data. It is important to note that your flight data must be in a “Zip” folder. To achieve this, you may “right click” on your flight data, and select “Send to…”, then select “Compressed (zipped) folder…”



9. Once your zip file is created, select it through the "Upload Flights" window.

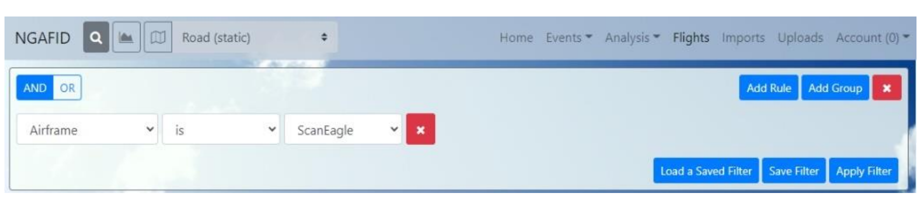


10. Monitor the status of your upload from the "Uploads" page on NGAFID.org.

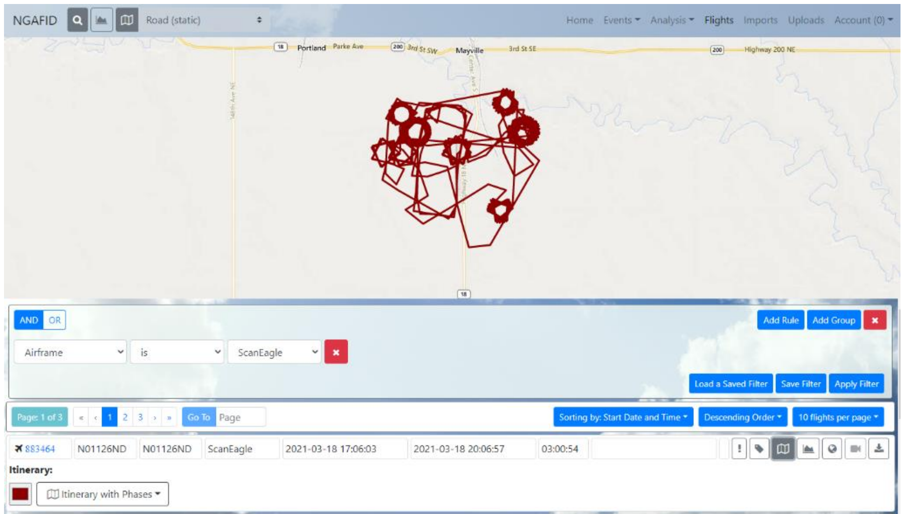


Viewing Individual Flights

11. To view successfully-processed flights, proceed to the "Flights" tab.
12. In the filtering options, select "Add Rule."
13. Under "Select Rule," choose "Airframe."
14. Select the airframe you wish to view. You can also filter the database by any number of identifying parameters, depending on what you wish to view.
15. Once your filter is correct, select "Apply Filter."



16. The next page will provide a list of all available flights with that filter. To view an individual flight overlaid on a map, you can select the "map" icon. *Helpful Tip: the buttons on the right side of the window allow you to overlay multiple flights if you wish to compare.*



15 APPENDIX E

Title of Project: **Perceptions of the Risk Associated with UAS Integration into the National Airspace System**

Principal Investigator: Ryan Guthridge, 701-777-3543 , ryan.guthridge@und.edu

Purpose of the Study:
This study aims to evaluate the highest perceived risks of integrating UAS into the National Airspace System. The study will pertain to completion of a survey instrument, which asks questions related to UAS integration and the associated flight safety risks of doing so. The results of this research will be used to inform FAA research, which is funded through the ASSURE center of excellence for UAS research.

Procedures to be followed:
After consenting to participate in this study, you will take an online survey asking you various questions about your perceptions of UAS integration into the National Airspace System. If you prefer not to answer a question or questions simply state that you do not wish to answer and the question will be skipped. You may discontinue participation and leave at any time.

Risks:
There are no risks in participating in this research beyond those experienced in everyday life.

Benefits:
You may not benefit personally from being in this study. However, we hope that, in the future, this research will inform a safer integration of UAS into the National Airspace System. This may provide vital information that could affect FAA policy and data analytic requirements for UAS operators.

Statement of Confidentiality:
Identifiable information collected as part of this study will be used for the sole purpose of linking your experience in this study to your progress in future flight labs. Examples of what future data may be collected are total flight time, total ground training time, and course on-time performance. Once a link is made between you and the flight training record, the identifiable information will be deleted from record. If this research is published, no information that would identify you will be included since any identifiable information will have been removed from all source records.

All survey responses that we receive will be treated confidentially and stored in a password-protected location. However, given that the surveys can be completed from any computer (e.g., personal, work, school), we are unable to guarantee the security of the computer on which you choose to enter your responses. As a participant in our study, we want you to be aware that certain "key logging" software programs exist that can be used to track or capture data that you enter and/or websites that you visit.

Right to Ask Questions:
The researcher conducting this study is Ryan Guthridge. You may ask any questions you have now. If you later have questions, concerns, or complaints about the research please contact Ryan Guthridge at 701-777-3543 during the day.

If you have questions regarding your rights as a research subject, you may contact The University of North Dakota Institutional Review Board at (701) 777-4279 or UND.irb@UND.edu. You may contact the UND IRB with problems, complaints, or concerns about the research. Please contact the UND IRB if you cannot reach research staff, or you wish to talk with someone who is an informed individual who is independent of the research team.

General information about being a research subject can be found on the Institutional Review Board website "Information for Research Participants" http://und.edu/research/resources/human-subjects/research- participants.html

Compensation:
You will not receive compensation for your participation.

Voluntary Participation:
You do not have to participate in this research. You can stop your participation at any time. You may refuse to participate or choose to discontinue participation at any time without losing any benefits to which you are otherwise entitled.

You do not have to answer any questions you do not want to answer. You must be 18 years of age older to participate in this research study.

By proceeding with the survey, you are providing your consent to participate in this research.

Page Break

What is your role in the UAS Industry?

- ○ Business Manager
- ○ UAS Operator / Pilot
- ○ UAS Regulator (FAA)
- ○ Student in a UAS Program
- ○ Not a UAS Industry Professional

How long have you been involved with UAS?

- ○ Less than 1 Year
- ○ 2-3 Years
- ○ 3-5 Years
- ○ 5-7 Years
- ○ 7-10 Years
- ○ More than 10 Years

Page Break

Q1 Of the following choices, what is the biggest safety risk of UAS integration into the National Airspace System?

○ Collision with Manned Aircraft

○ Detection and Avoidance

○ Airspace Violation

○ Lost of Command and Control (Datalink)

○ Destruction of Persons or Property on the Ground

○ Other ____

Q2 Rank the following concerns in order of safety risk for UAS.

______ Loss of Aircraft Control Inflight (LOC-I)
______ Lost of Command and Control (C2 Link)
______ Collision in the Air
______ Rotor Separation
______ Loss of Battery Power
______ Collision on the Ground
______ Hard Landing

Page Break

Q3 Are there any other safety risks associated with UAS integration into the National Airspace System, that haven't been addressed in this survey? *(please list below)*

____

____

____

____

____

Page Break

Q4 How could UAS operators be encouraged to voluntarily submit their Flight Data for safety analysis and sharing? (FAA ASIAS Program)

____

____

____

____

____

End of Block: Default Question Block

16 APPENDIX F

Results from Q3 - Additional Risk not Addressed

- Lack of manned aircraft using ADS-B out. Unmanned aircraft cannot rely on technology to aid in see and avoid. ADS-B our needs to become a mandate for all aircraft, just like remote ID for unmanned aircraft.
- From a manned aviation background, the biggest risk I see is collisions with manned aircraft. there are mitigation methods to address many hazards of this variety, but agricultural manned operators are very hard to mitigate for, when they fly very low, fast, and with little requirements for ADS-B.
- Weather variables and limitations of UAS
- Loss of navigational capability (GPS status or autopilot malfunction).
- Delays in FAA implementation are causing a mish mash of waivers. a robust BVLOS system would be very beneficial to UAS operators everywhere.
- Proper communication with everyone in the airspace
- "'-The pilots ability to control the UAS safely
- -remote ID security risks (data breach) "
- I hold a commercial MEL/SEL. Will my manned aircraft certifications be in jeopardy if I have a violation or crash while flying UAS?
- Regulations and laws for right of way.
- auditing and training of pilots, just because you pass a test doesn't mean an individual is following the rules or has piloting skills-how do we make certain pilots/missions are following the rules of part 107? Are the pilots skilled for the work performed?
- Use of UAS for nefarious reasons resulting in harm to others
- Use of UAS for nefarious reasons resulting in the disruption of crewed flight operations including airline operation.
- Poor public perception and trust in use of UAS
- Lack of cybersecurity resulting in data being stolen
- Lack of navigation equipment standards in UAS resulting in collision with other aircraft. (UAS GPS often do not meet aviation standards).
- Conspicuity issues related to sUAS compared to crewed aircraft.
- Development of UAS by non-Aviation related companies, GUI does not integrate with crewed flight operations, never seen a UAS that can pull up a standard instrument approach procedures for example.
- Increase use of UAS resulting in loss of privacy
- Poor infrastructure to support UAS operations resulting in limited use, including airport design, communications, having method for UAS to consistently detect non-cooperative traffic over large areas such as an old C152.
- Increase congestion of airspace and existing infrastructure.
- Lack of safety assurance processes within flight operations as well FAA resulting in not learning lessons (FDM can help this). No data showing us the reliability of UAS Systems which means we don't know actual risk.
- Lack of reporting of UAS incidents resulting in improper view of actual risk as UAS traffic increases.

- Lack of UAS focused certification standards supported by FAA that allows us to have assurance of UAS safety of flight.
- Regulating busy airspace (ORD, ATL) and also the corn fields of ND
- Incompetent UAS Operators
- Pilot proficiency with aircraft nearing 55lbs. A lot of damage can be done from someone with little training.
- Lack of FAA pilot certification for a majority of private UAS Operators. As an airline pilot, I have personally experienced two UAS incidents involving Intentional noncompliance with FAA regulations. In both situations, the UAS Operator was a private UAS operator unaware of airspace rules or did not care. We need a better education process for all UAS Operators.

17 APPENDIX G

Results from Q4 – Recommendations for Voluntary Submission

- Freedom from certificate action like ASAP.
- Inform them how it will be used and what outcomes the FAA hopes to gain, or implement. Moving away from its solely on unmanned pilots to avoid other traffic, instead towards manned and unmanned having an equal share in responsibilities to see and avoid.
- "I am not aware there is a program to submit flight data. If there is such a program, I believe the FAA should start with ensuring there are no repercussions of flight data that indicates an unintentional violation occurred."
- Not sure
- Being asked to be part of the solution and thereafter included in the process.
- Reduced insurance, priority/increased success in FAA waiver authorization (BVLOS, Ops over people, etc.).
- require it for commercial operators like FOQA, or similar GA FDM programs
- Something akin to the NASA safety reporting that is anonymous.
- By making sure that risk of voluntarily shared data isn't punished by the FAA.
- Add UAS operators to the current NASA/ASRS program.
- "Flight data could be potentially incriminating as FAA guidelines for UAS are extremely vague."
- "Incentives for submitting data would also be better"
- Explain how the data can better the industry.
- UAS is still just a tool to assist in some other type of activities yet it's being modeled like civil manned Part 121 or 135 operations. UAS is far too overregulated to be useful in any way, yet operators are desperately trying to further their business by pushing for integration. The whole UAS atmosphere is a joke. There are two types of operators: those that take it seriously and those that flagrantly violate rules for lack of awareness or care. The regulators don't care enough to enforce the flagrany rule violators while the serious operators fear the FAA worse than they fear God. I've seen it firsthand on many accounts with operators who are overly cautious. This results in regulatory and research maturity at a pace far too slow to have tangible impacts and benefits. In other words, the current state of the UAS industry is insufficient to support a program as complex and robust as safety data analysis and sharing. Operators take drones TOO seriously or not seriously enough. The latter of which will have no motivation whatsoever to go to the trouble of sharing data. UAS needs to be allowed to evolve and mature (BVLOS commercial ops, etc.) to be more useful to operators before many of them can be convinced to share safety data or even for there to be enough sources of data to be significant. UAS needs less complex regulations and barriers to entry, not more, and that's all I and most others will see this as.
- N/a
- easily make it happen "in the background" without operator intervention
- similar to commercial aviation, we all make mistakes, don't penalize unless intentionally breaking the rules and by submitting the pilot is protecting him/her self from penalties. Importantly data and cooperation will be vital for continued growth here in the US in the UAS industry. Self-governance can positively impact the overall culture. No doubt there are always a few bad apples, but national membership with qualifications may be a way to

add training for flight and safety. Like Realtors for example-code of ethics, and self-governance to penalize members who break the rules. Create a published list of wrongdoers. Similar to what states do for other licensees in real estate, insurance, financial for example.

- Make it simple
- make it automatic
- Develop a standard for data, so programs can be developed that work on ALL UAS. Get manufacturers to use standard. Right now every device is unique and proprietary making it very hard to transfer telemetry data.
- Same way as manned pilots, NASA forms
- Nothing with the FAA is easy, make it easy. If it is a burden, then individuals will not participate.
- Similar to open ASAP reporting systems. There is no punishment for incident having in flight if it’s out of your control or something that can’t be avoided
- Monetary kick back
- Incentives, not sure what exactly. It would be hard to encourage an average UAS operator without it.
- Tracking their own flight data
- Advertise the process on UAS social media and web pages.
- Financial incentives and equipment subsidies.
- UAS Operators need the same incentives given to the airlines with the ASAP programs or NASA form completed by manned aircraft. There is generally no disciplinary actions associated with reporting within the program.

18 APPENDIX H

Proximity Event Definition

```
public class CalculateProximity {

    //Proximity events (and potentially other complicated event calculations) will have negative IDs so they
    //can be excluded from the regular event calculation process
    public static final int adjacencyEventDefinitionId = -1;

    //use this to get a representation of a flight's current time, position and altitude

    public static long timeMatchFlights = 0;
    public static long locMatchFlights = 0;
    public static long eventsFound = 0;

    protected static class FlightTimeLocation {
        //set to true if the flight has the required time series values and a start and
        //end date time
        boolean valid = false;

        //set to true when the double and string time series data has been
        //read from the database, and the epochTime array has been calculated
        boolean hasSeriesData = false;

        int fleetId;
        int flightId;
        int airframeNameId;

        String startDateTime;
        String endDateTime;

        double minLatitude;
```

```
    double maxLatitude;
    double minLongitude;
    double maxLongitude;

    double minAltMSL;
    double maxAltMSL;

    long[] epochTime;
    double[] altitudeMSL;
    double[] altitudeAGL;
    double[] latitude;
    double[] longitude;
    double[] indicatedAirspeed;

    StringTimeSeries dateSeries;
    StringTimeSeries timeSeries;

    public FlightTimeLocation(Connection connection, int fleetId, int flightId, int airframeNameId, String startDateTime, String endDateTime) throws SQLException {
        this.fleetId = fleetId;
        this.flightId = flightId;
        this.airframeNameId = airframeNameId;
        this.startDateTime = startDateTime;
        this.endDateTime = endDateTime;

        //first check and see if the flight had a start and end time, if not we cannot process it
        //System.out.println("Getting info for flight with start date time: " + startDateTime + " and end date time: " + endDateTime);

        if (startDateTime == null || endDateTime == null) {
            //flight didnt have a start or end time
            valid = false;
```

```
            return;
        }

        //then check and see if this was actually a flight (RPM > 800)
        Pair<Double,Double> minMaxRPM1 = DoubleTimeSeries.getMinMax(connection, flightId, "E1 RPM");
        Pair<Double,Double> minMaxRPM2 = DoubleTimeSeries.getMinMax(connection, flightId, "E2 RPM");

        System.out.println("minMaxRPM1: " + minMaxRPM1);
        System.out.println("minMaxRPM2: " + minMaxRPM2);

        if (minMaxRPM1 != null) System.out.println("min max E1 RPM: " + minMaxRPM1.first() + ", " + minMaxRPM1.second());
        if (minMaxRPM2 != null) System.out.println("min max E2 RPM: " + minMaxRPM2.first() + ", " + minMaxRPM2.second());

        if ((minMaxRPM1 == null && minMaxRPM2 == null)  //both RPM values are null, can't calculate exceedence
                || (minMaxRPM2 == null && minMaxRPM1.second() < 800) //RPM2 is null, RPM1 is < 800 (RPM1 would not be null as well)
                || (minMaxRPM1 == null && minMaxRPM2.second() < 800) //RPM1 is null, RPM2 is < 800 (RPM2 would not be null as well)
                || ((minMaxRPM1 != null && minMaxRPM2 != null && minMaxRPM1.second() < 800 && minMaxRPM2.second() < 800))) { //RPM1 and RPM2 < 800
            //couldn't calculate exceedences for this flight because the engines never kicked on (it didn't fly)
            valid = false;
            return;
        }

        //then check and see if this flight had a latitude and longitude, if not we cannot calculate adjacency
        Pair<Double,Double> minMaxLatitude = DoubleTimeSeries.getMinMax(connection, flightId, "Latitude");
```

```
Pair<Double,Double> minMaxLongitude = DoubleTimeSeries.getMinMax(connection, flightId, "Longitude");

//if (minMaxLatitude != null) System.out.println("min max latitude: " + minMaxLatitude.first() + ", " + minMaxLatitude.second());
//if (minMaxLongitude != null) System.out.println("min max longitude: " + minMaxLongitude.first() + ", " + minMaxLongitude.second());

if (minMaxLatitude == null || minMaxLongitude == null) {
   //flight didn't have latitude or longitude
   valid = false;
   return;
}

minLatitude = minMaxLatitude.first();
maxLatitude = minMaxLatitude.second();
minLongitude = minMaxLongitude.first();
maxLongitude = minMaxLongitude.second();

//then check and see if this flight had alt MSL, if not we cannot calculate adjacency
Pair<Double,Double> minMaxAltMSL = DoubleTimeSeries.getMinMax(connection, flightId, "AltMSL");

//if (minMaxAltMSL != null) System.out.println("min max alt MSL: " + minMaxAltMSL.first() + ", " + minMaxAltMSL.second());

if (minMaxAltMSL == null) {
   //flight didn't have alt MSL
   valid = false;
   return;
}

minAltMSL = minMaxAltMSL.first();
```

```
            maxAltMSL = minMaxAltMSL.second();

            //this flight had the necessary values and time series to calculate adjacency
            valid = true;
        }

        public boolean getSeriesData(Connection connection) throws SQLException {
            //get the time series data for altitude, latitude and longitude
            DoubleTimeSeries altMSLSeries = DoubleTimeSeries.getDoubleTimeSeries(connection, flightId, "AltMSL");
            DoubleTimeSeries altAGLSeries = DoubleTimeSeries.getDoubleTimeSeries(connection, flightId, "AltAGL");
            DoubleTimeSeries latitudeSeries = DoubleTimeSeries.getDoubleTimeSeries(connection, flightId, "Latitude");
            DoubleTimeSeries longitudeSeries = DoubleTimeSeries.getDoubleTimeSeries(connection, flightId, "Longitude");
            DoubleTimeSeries indicatedAirspeedSeries = DoubleTimeSeries.getDoubleTimeSeries(connection, flightId, "IAS");

            //check to see if we could get these columns
            if (altMSLSeries == null || altAGLSeries == null || latitudeSeries == null || longitudeSeries == null || indicatedAirspeedSeries == null) return false;

            altitudeMSL = altMSLSeries.innerArray();
            altitudeAGL = altAGLSeries.innerArray();
            latitude = latitudeSeries.innerArray();
            longitude = longitudeSeries.innerArray();
            indicatedAirspeed = indicatedAirspeedSeries.innerArray();

            //calculate the epoch time for each row as longs so they can most be quickly compared
            //we need to keep track of the date and time series for inserting in the event info
            dateSeries = StringTimeSeries.getStringTimeSeries(connection, flightId, "Lcl Date");
            timeSeries = StringTimeSeries.getStringTimeSeries(connection, flightId, "Lcl Time");
```

```
        StringTimeSeries utcOffsetSeries = StringTimeSeries.getStringTimeSeries(connection, flightId, "UTCOfst");

        //check to see if we could get these columns
        if (dateSeries == null || timeSeries == null || utcOffsetSeries == null) return false;

        //System.out.println("date length: " + dateSeries.size() + ", time length: " + timeSeries.size() + ", utc length: " + utcOffsetSeries.size());
        int length = dateSeries.size();

        epochTime = new long[length];
        for (int i = 0; i < length; i++) {
            if (dateSeries.get(i) == null || dateSeries.get(i).equals("")
                    || timeSeries.get(i) == null || timeSeries.get(i).equals("")
                    || utcOffsetSeries.get(i) == null || utcOffsetSeries.get(i).equals("")) {
                epochTime[i] = 0;
                continue;
            }

            epochTime[i] = TimeUtils.toEpochSecond(dateSeries.get(i), timeSeries.get(i), utcOffsetSeries.get(i));
        }

        hasSeriesData = true;

        return true;
    }

    public boolean hasRegionOverlap(FlightTimeLocation other) {
        return other.maxLatitude >= this.minLatitude && other.minLatitude <= this.maxLatitude
                && other.maxLongitude >= this.minLongitude && other.minLongitude <= this.maxLongitude;
    }
```

```
    public boolean isValid() {
        return valid;
    }

    public boolean hasSeriesData() {
        return hasSeriesData;
    }

    public boolean alreadyProcessed(Connection connection) throws SQLException {
        PreparedStatement stmt = connection.prepareStatement("SELECT flight_id FROM flight_processed WHERE fleet_id = ? AND flight_id = ? AND event_definition_id = ?");
        stmt.setInt(1, fleetId);
        stmt.setInt(2, flightId);
        stmt.setInt(3, adjacencyEventDefinitionId);

        System.out.println(stmt.toString());

        //if there was a flight processed entry for this flight it was already processed
        ResultSet resultSet = stmt.executeQuery();
        if (resultSet.next()) {
            System.out.println("already processed!");
            resultSet.close();
            stmt.close();
            return true;
        } else {
            System.out.println("not already processed!");
            resultSet.close();
            stmt.close();
            return false;
        }
    }
```

```
public static boolean proximityAlreadyCalculated(Connection connection, FlightTimeLocation first, FlightTimeLocation second) throws SQLException {
        PreparedStatement stmt = connection.prepareStatement("SELECT flight_id FROM events WHERE flight_id = ? AND other_flight_id = ?");
        stmt.setInt(1, first.flightId);
        stmt.setInt(2, second.flightId);

        System.out.println(stmt.toString());

        //if there was a flight processed entry for this flight it was already processed
        ResultSet resultSet = stmt.executeQuery();
        if (resultSet.next()) {
            System.out.println("proximity event already exists!");
            resultSet.close();
            stmt.close();
            return true;
        } else {
            System.out.println("proximity does not already exist!");
            resultSet.close();
            stmt.close();
            return false;
        }
    }

    public void updateWithEvent(Connection connection, Event event, String startDateTime) throws SQLException {

        event.updateDatabase(connection, fleetId, flightId, adjacencyEventDefinitionId);
        event.updateStatistics(connection, fleetId, airframeNameId, adjacencyEventDefinitionId);

        double severity = event.getSeverity();
        double duration = event.getDuration();
```

```
        PreparedStatement stmt = connection.prepareStatement("UPDATE flight_processed SET count = count + 1, sum_duration = sum_duration + ?, min_duration = LEAST(min_duration, ?), max_duration = GREATEST(max_duration, ?), sum_severity = sum_severity + ?, min_severity = LEAST(min_severity, ?), max_severity = GREATEST(max_severity, ?) WHERE fleet_id = ? AND flight_id = ? AND event_definition_id = ?");
        stmt.setInt(1, fleetId);
        stmt.setInt(2, flightId);
        stmt.setInt(3, adjacencyEventDefinitionId);
        stmt.setDouble(4, duration);
        stmt.setDouble(5, duration);
        stmt.setDouble(6, duration);
        stmt.setDouble(7, severity);
        stmt.setDouble(8, severity);
        stmt.setDouble(9, severity);
        System.out.println(stmt.toString());
        stmt.executeUpdate();
        stmt.close();

        EventStatistics.updateFlightsWithEvent(connection, fleetId, airframeNameId, adjacencyEventDefinitionId, startDateTime);

      }
   }

   static String timeSeriesName = "Lcl Time";
   static String dateSeriesName = "Lcl Date";

   public static void processFlightWithError(Connection connection, int fleetId, int flightId) throws SQLException {
      PreparedStatement stmt = connection.prepareStatement("INSERT INTO flight_processed SET fleet_id = ?, flight_id = ?, event_definition_id = ?, count = 0, had_error = 1");
      stmt.setInt(1, fleetId);
      stmt.setInt(2, flightId);
```

```
        stmt.setInt(3, adjacencyEventDefinitionId);
        //System.out.println(stmt.toString());
        stmt.executeUpdate();
        stmt.close();
    }

    public static void processFlight(Connection connection, Flight flight, UploadProcessedEmail uploadProcessedEmail) {
        System.out.println("Processing flight: " + flight.getId() + ", " + flight.getFilename());

        int fleetId = flight.getFleetId();
        int flightId = flight.getId();
        int airframeNameId = flight.getAirframeNameId();
        String flightFilename = flight.getFilename();

        try {
            //get enough information about the flight to determine if we can calculate adjacencies with it
            FlightTimeLocation flightInfo = new FlightTimeLocation(connection, flight.getFleetId(), flightId, flight.getAirframeNameId(), flight.getStartDateTime(), flight.getEndDateTime());

            if (!flightInfo.isValid()) {
                uploadProcessedEmail.addProximityError(flightFilename, "could not calculate proximity for flight " + flightId + ", '" + flightFilename + "' - was missing required data columns (date, time, latitude, longitude, altitude and/or indicated airspeed)");
                processFlightWithError(connection, fleetId, flightId);
                return;
            }

            ArrayList<Flight> potentialFlights = Flight.getFlights(connection, "(id != " + flightId + " AND start_timestamp <= UNIX_TIMESTAMP('" + flightInfo.endDateTime + "') AND end_timestamp >= UNIX_TIMESTAMP('" + flightInfo.startDateTime + "'))");

            System.out.println("Found " + potentialFlights.size() + " potential time matched flights.");
```

```
//System.out.println("Flight start time: " + flightInfo.startDateTime + ", end time: " + flightInfo.endDateTime);
//System.out.println("Flight latitude min: " + flightInfo.minLatitude + ", max: " + flightInfo.maxLatitude);
//System.out.println("Flight longitude min: " + flightInfo.minLongitude + ", max: " + flightInfo.maxLongitude);

ArrayList<Event> eventList = new ArrayList<>();
String startTime = null;
String endTime = null;
String otherStartTime = null;
String otherEndTime = null;

int startLine = -1;
int endLine = -1;
int otherStartLine =-1;
int otherEndLine = -1;

int startCount = 0;
int stopCount = 0;
double severity = 0;

//TODO: should probably grab these from the database event definition instead of hard
//coding them; but we don't need to pull the event definition so this is a tad bit faster.
int startBuffer = 1;
int stopBuffer = 30;

for (Flight otherFlight : potentialFlights) {
    //System.out.println("\tmatched to flight with start time: " + otherFlight.getStartDateTime() + ", end time: " + otherFlight.getEndDateTime());
    timeMatchFlights++;
```

```
FlightTimeLocation otherInfo = new FlightTimeLocation(connection, otherFlight.getFleetId(), otherFlight.getId(), otherFlight.getAirframeNameId(), otherFlight.getStartDateTime(), otherFlight.getEndDateTime());
if (!otherInfo.isValid()) {
    //matched flight did not have all the information necessary to compute adjacency
    continue;
}

//see if proximity between these two flights was already calculated, if so we can skip
if (FlightTimeLocation.proximityAlreadyCalculated(connection, otherInfo, flightInfo)) {
    System.out.println("Not re-performing proximity calculation");
    continue;
}

//System.out.println("\t\tother latitude min: " + otherInfo.minLatitude + ", max: " + otherInfo.maxLatitude);
//System.out.println("\t\tother longitude min: " + otherInfo.minLongitude + ", max: " + otherInfo.maxLongitude);

if (flightInfo.hasRegionOverlap(otherInfo)) {
    //System.out.println("\t\tLatitude/Longitude overlap!");
    locMatchFlights++;

    if (!flightInfo.hasSeriesData()) {
        if (!flightInfo.getSeriesData(connection)) {
            //could not get the required time series data columns
            processFlightWithError(connection, fleetId, flightId);
            return;
        }
    }

    if (!otherInfo.getSeriesData(connection)) {
```

```
    //the other flight didn't have the necesary time series data columns
    continue;
}


//skip the first 30 seconds as it is usually the FDR being initialized
int i = 30, j = 30;

int totalMatches = 0;
//System.out.println("\t\tgot series data for both flights, iterate over times");
while (i < flightInfo.epochTime.length && j < otherInfo.epochTime.length) {
    //skip entries where the epoch time was 0 (the date/time was null)
    if (flightInfo.epochTime[i] == 0) {
        i++;
        continue;
    }

    if (otherInfo.epochTime[j] == 0) {
        j++;
        continue;
    }

    //make sure both iterators are for the same time
    if (flightInfo.epochTime[i] < otherInfo.epochTime[j]) {
        i++;
        continue;
    }

    if (otherInfo.epochTime[j] < flightInfo.epochTime[i]) {
        j++;
        continue;
    }
```

```
double distanceFt = Airports.calculateDistanceInFeet(flightInfo.latitude[i], flightInfo.longitude[i], otherInfo.latitude[j], otherInfo.longitude[j]);
double altDiff = Math.abs(flightInfo.altitudeMSL[i] - otherInfo.altitudeMSL[j]);
distanceFt = Math.sqrt((distanceFt * distanceFt) + (altDiff * altDiff));

if (distanceFt < 1000.0 && flightInfo.altitudeAGL[i] >= 50 && otherInfo.altitudeAGL[j] >= 50 && flightInfo.indicatedAirspeed[i] > 20 && otherInfo.indicatedAirspeed[j] > 20) {
    /*
    System.out.println("\t\t\tother time[" + j + "]: " + otherInfo.epochTime[j] + " == flight time[" + i + "]: " + flightInfo.epochTime[i]
        + ", flight lat/lon: " + flightInfo.latitude[i] + " " + flightInfo.longitude[i] + ", other lat/lon: " + otherInfo.latitude[j] + " " + otherInfo.longitude[j]
        + " -- distance: " + distanceFt
        );
    */

    //System.out.println("\t\t\t\t\tflight alt AGL: " + flightInfo.altitudeAGL[i] + ", other alt AGL: " + otherInfo.altitudeAGL[j] + ", final distance: " + distanceFt);

    //startTime is null if an exceedence is not being tracked
    if (startTime == null) {
        //start tracking a new exceedence
        startTime = flightInfo.dateSeries.get(i) + " " + flightInfo.timeSeries.get(i);
        otherStartTime = otherInfo.dateSeries.get(j) + " " + otherInfo.timeSeries.get(j);

        startLine = i;
        otherStartLine = j;
        severity = distanceFt;
```

```
            //System.out.println("start date time: " + startTime + ", start line number: " + startLine);
        }
        endLine = i;
        otherEndLine = j;
        endTime = flightInfo.dateSeries.get(i) + " " + flightInfo.timeSeries.get(i);
        otherEndTime = otherInfo.dateSeries.get(j) + " " + otherInfo.timeSeries.get(j);

        if (distanceFt < severity) {
            //this time was even closer than the last closest proximity
            //for this event, update the severity
            severity = distanceFt;
        }

        //increment the startCount, reset the endCount
        startCount++;
        stopCount = 0;

    } else {
        //this time didn't trigger proximity

        if (startTime != null) {
            //we're already tracking a proximity event, so increment
            //the stop count
            stopCount++;

            if (stopCount == stopBuffer) {
                //System.err.println("Stop count (" + stopCount + ") reached the stop buffer (" + stopBuffer + "), new event created!");

                if (startCount < startBuffer) {
                    //we didn't have enough triggers to reach the start count so don't create
```

```
                //the event
            } else {
                //we had enough triggers to reach the start count so create the event
                Event event = new Event (startTime, endTime, startLine, endLine,
severity, otherFlight.getId());
                eventList.add(event);

                //add in an event for the other flight as well so we don't need to recalculate
this
                otherInfo.updateWithEvent(connection, new Event(otherStartTime,
otherEndTime, otherStartLine, otherEndLine, severity, flightId), otherFlight.getStartDateTime());
            }

            //reset the event values
            startTime = null;
            otherStartTime = null;
            endTime = null;
            otherEndTime = null;
            startLine = -1;
            otherEndLine = -1;
            endLine = -1;
            otherEndLine = -1;

            //reset the start and stop counts
            startCount = 0;
            stopCount = 0;
        }
    }
}

//iterate both as they had matching times
i++;
j++;
```

```
                totalMatches++;
            }
            //System.out.println("\t\tseries matched time on " + totalMatches + " rows");


            //if there was an event still going when one flight ended, create it and add it to the list
            if (startTime != null) {
                Event event = new Event(startTime, endTime, startLine, endLine, severity, otherFlight.getId());
                eventList.add( event );

                //add in an event for the other flight as well so we don't need to recalculate this
                otherInfo.updateWithEvent(connection, new Event(otherStartTime, otherEndTime, otherStartLine, otherEndLine, severity, flightId), otherFlight.getStartDateTime());
            }
        }
        //end the loop processing a particular flight
    }
    //end the loop processing all flights

    for (Event event : eventList) {
        System.out.println("\t" + event.toString());
        eventsFound++;
        uploadProcessedEmail.addProximity(flightFilename, "flight " + flightId + ", '" + flightFilename + "' - had a proximity event with flight " + event.getOtherFlightId() + " from " + event.getStartTime() + " to " + event.getEndTime());
    }

    System.out.println("\n");
```

19 APPENDIX I

Low Battery Level Event Definition

A Low Battery Level event occurs when (Battery Level < 25%) is triggered at least 1 time within 30 seconds, and ends when no trigger occurs for 30 seconds.

The parameter used for Battery Level is dependent on aircraft make and model, per what is recorded in the flight data recording. Note, for aircraft that have charging capabilities, such as a generator or alternator, a separate event has been created to detect "battery not charging." This event definition can be found in Appendix B. Known parameters for make and model are provided in Table 8.

Table 8. Known Parameters for Make and Model (Low Battery Level Event).

Make	Model	Parameter	Limit	Source
DJI	ALL	Battery(0):battery%	<25	User Manual, p20
Parrot	ALL	Battery_level	<25	Flight Data

20 APPENDIX J

Battery Not Charging Event Definition

A Battery Not Charging event occurs when (Battery Charging < 1) is triggered for a continuous duration of 60 seconds, and ends when no trigger occurs for 30 seconds.

The parameter used for Battery Charging is dependent on aircraft make and model, per what is recorded in the flight data recording. In the case of the Insitu Scan Eagle, a discrete parameter records when the battery is charging (1 = charging). When the discrete parameter records a value of "0", the battery is not charging. Known parameters for make and model are provided in Table 9.

Table 9. Known Parameters for Make and Model (Battery not charging event.

Make	Model	Parameter	Limit	Source
Insitu	Scan Eagle	DID_ACPARAM_BATT_CHARGING	<1, >60s	Flight Data

21 APPENDIX K

Low Battery Temperature Event Definition

A Low Battery Temperature event occurs when (Battery Temperature < 5) is triggered for a continuous duration of 10 seconds, and ends when no trigger occurs for 30 seconds.

The manufacturer's limit for Low Battery Temperature is dependent on battery make and model, with limits varying based on the capabilities of the battery itself. For example, in the case of the DJI Matrice 600, a parameter is recorded for Battery Temperature, with a published low temperature limit of 5 degrees. Known parameters for make and model are provided in Table 10.

Table 10. Known Parameters for Make and Model (Low Battery Temperature Event).

Make	Model	Parameter	Limit	Source
DJI	Matrice 600	Battery(0):Temp	<5	User Manual, p19

22 APPENDIX L

High Altitude Limit Exceeded Event Definition

A High Altitude Limit Exceeded event occurs when (AltAGL > 400) is triggered for a continuous duration of 5 seconds, and ends when no trigger occurs for 30 seconds.

The event monitors the aircraft's altitude AGL and compares it to the regulatory requirement (FAR Part 107) to be at, or below, 400 feet AGL. This event uses the aircraft's recorded altitude (in barometric or GPS altitude) and compares it against publicly-available terrain data to calculate an AGL height at any given point along the flight path. In most cases, GPS resolution is sufficient to be confident in the aircraft's geospatial position above the ground, which improves the accuracy of the AGL calculation in the NGAFID.

Table 11. Known Parameters for Make and Model (High Altitude Limit Exceeded Event).

Make	Model	Parameter	Limit	Source
ALL	AIRCRAFT	AltAGL (computed)	>400	FAR 107

23 APPENDIX M

Low Positional Accuracy Event Definition

A Low Positional Accuracy event occurs when (GPS Health is low and/or GPS Error is high) is triggered for a continuous duration of 5 seconds, and ends when no trigger occurs for 30 seconds.

The event monitors the health of the GPS position being received by the aircraft, which directly impacts the operational capabilities of the UAS. This event is highly dependent on the source and type of data recorded by the UAS, along with the associated impact it has on the operational capabilities of the UAS in controlled and autonomous flight. In the case of DJI, a parameter for GPS Health is recorded on a scale of 0 to 5. In situations where the GPS Health is 2 or less, the UAS is unable to use GPS for stabilization features, which degrades the controllability of the UAS. For Parrot UAS, the aircraft records GPS Position Error, which is an increasing value when there is a degraded GPS solution detected. In this case, any value greater than 0 is an indication of a low positional accuracy that affects the controllability of the UAS. Known parameters for make and model are provided in Table 12.

Table 12. Known Parameters for Make and Model (Low Positional Accuracy Event).

Make	Model	Parameter	Limit	Source
DJI	ALL	gpsHealth	<2	DJI
Parrot	ALL	product_gps_position_error	>0	Parrot

24 REFERENCES

Elsner, H. (2017). Q500log2kml User Manual: FlightLog Data Evaluation for Yuneec Quad or Hexa Copter Ver. 2.8. Retrieved from http://b-drone.nl/wp-content/uploads/2017/02/Q500log2kml-manual.pdf

FAA Safety Team (2018). *Aircraft Performance Monitoring.* Retrieved from: https://www.faa.gov/news/safety_briefing/2018/media/SE_Topic_18-12.pdf

The National General Aviation Flight Information Database (NGAFID). (n.d.). https://ngafid.org/.