DOT/FAA/AR-xx/xx

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

UAS Parameters, Exceedances, Recording Rates for ASIAS

June 30, 2020

Final Report

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

THIRD PARTY RESEARCH. PENDING FAA REVIEW.

Technical Report Documentation Page

1. Report No. DOT/FAA/AR-xx/xx	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle STATE-OF-THE-ART UNMANNED FLIGHT DATA MONITORING AND PARAMTERS, EXCEEDANCES AND RECORDING RATES		5. Report Date June 30, 2020
		6. Performing Organization Code
7. Author(s) Jim Higgins, UND; Dave Esser, ERAU, Ryan Wallace, ERAU, Li Zhang, MSU, Stephen France, MSU		8. Performing Organization Report No.
9. Performing Organization Name and Address ASSURE		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address		13. Type of Report and Period Covered Draft Report
		14. Sponsoring Agency Code
15. Supplementary Notes The Federal Aviation Administration Aviation William J. Hughes Technical Center Research Division COR was		

16. Abstract

Currently, state-of-the-art Unmanned Flight Data Monitoring (UFDM) includes facets that are in the formation stage. The basic building blocks of Flight Data Monitoring (FDM) are present even though there are no formally established nationwide UFDM standards, data repositories or governance structures. UAS flight data is relatively easily accessible to operators through either stand-alone recorders or flight telemetry sent back to the Ground Control Station (GCS). A larger FDM program could likely utilize this basic data, even though it varies widely in terms of usefulness and robustness. Several different data types and parameters generated by UAS telemetry and recorders demonstrate usefulness toward a fully-functioning UFDM solution. Where FDM programs are utilized, there are well-described economic and safety benefits. The Unmanned Air Safety Team (UAST) has formed a data team tasked with examining available flight data. The elements needed to develop a fully-functioning nationwide UFDM include the creation of a database, development of analytical tools and governance. UAS accident categories need to be developed similar to other types of Aviation Safety Information Analysis and Sharing (ASIAS) processes in other industries (commercial aviation, rotorcraft, etc.). Some categories identified in this study include: loss of battery, loss of command and control, rotor separation, loss of control, hard landing, collision on ground and collision in air. A UFDM data standard was developed using these categories. The standard includes type of parameter, refresh rate and whether the parameter can be derived. Comparisons were drawn across commercial, general and rotorcraft FDM data standards. Similar to other ASIAS efforts, the next steps toward implementing a nationwide UAS FDM program involve the creation of a database and establishing formal governance. Both of these future efforts could easily be developed along a similar path that was used in the creation of the National General Aviation Flight Information Database (NGAFID). Operators and the government would likely experience similar benefits found in other ASIAS endeavors, such as commercial aviation and rotorcraft. Given time and effort, a fully implemented and robust UFDM program can be successfully designed and deployed.

17. Key Words		18. Distribution Statement This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages	22. Price

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

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AC	Advisory Circular
AGL	Above Ground Level
AI	Artificial Intelligence
ASIAS	Aviation Safety Information Analysis and Sharing
ASRS	Aviation Safety and Reporting System
ATC	Air Traffic Control
CAST	Commercial Air Safety Team
CUSUM	Cumulative SUM
EWMA	Exponential Weighted Moving Average
FAA	Federal Aviation Administration
FDM	Flight Data Monitoring
FDR	Flight Data Recorders
FOQA	Flight Operations Quality Assurance
GA	General Aviation
GAJSC	General Aviation Joint Steering Committee
GCS	Ground Control Station
GPS	Global Positioning System
GUI	Graphics User Interface
HAT	Height Above Touchdown
ICAO	International Civil Aviation Organization
ICI	Imperial Chemical Industries
MOQA	Maintenance Operations Quality Assurance
MSL	Mean Sea Level
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NGAFID	National General Aviation Flight Information Database
PI	Principal Investigator
QA	Quality Assurance
QAR	Quick Access Recorder
RPM	Revolutions Per Minute
RTH	Return to Home
SME	Subject Matter Expert
SMM	Safety Management Manual
SMS	Safety Management System
SOM	Self-Organizing Maps
SQL	Structured Query Language
sUAS	Small unmanned aircraft systems
TQM	Total Quality Management
UAS	Unmanned Aircraft System
UFD	Unmanned Flight Data
UFDM	Unmanned Flight Data Monitoring
UAST	Unmanned Air Safety Team

EXECUTIVE SUMMARY

Currently, state-of-the-art Unmanned Flight Data Monitoring (UFDM) includes facets that are in the formation stage. The basic building blocks of Flight Data Monitoring (FDM) are present even though there are no formally established nationwide UFDM standards, data repositories or governance structures. UAS flight data is relatively easily accessible to operators through either stand-alone recorders or flight telemetry sent back to the Ground Control Station (GCS). A larger FDM program could likely utilize this basic data, even though it varies widely in terms of usefulness and robustness. Several different data types and parameters generated by UAS telemetry and recorders demonstrate usefulness toward a fully-functioning UFDM solution. Where FDM programs are utilized, there are well-described economic and safety benefits. The Unmanned Air Safety Team (UAST) has formed a data team tasked with examining available flight data. The elements needed to develop a fully-functioning nationwide UFDM include the creation of a database, development of analytical tools and governance. UAS accident categories need to be developed similar to other types of Aviation Safety Information Analysis and Sharing (ASIAS) processes in other industries (commercial aviation, rotorcraft, etc.). Some categories identified in this study include: loss of battery, loss of command and control, rotor separation, loss of control, hard landing, collision on ground and collision in air. A UFDM data standard was developed using these categories. The standard includes type of parameter, refresh rate and whether the parameter can be derived. Comparisons were drawn across commercial, general and rotorcraft FDM data standards. Similar to other ASIAS efforts, the next steps toward implementing a nationwide UAS FDM program involve the creation of a database and establishing formal governance. Both of these future efforts could easily be developed along a similar path that was used in the creation of the National General Aviation Flight Information Database (NGAFID). Operators and the government would likely experience similar benefits found in other ASIAS endeavors, such as commercial aviation and rotorcraft. Given time and effort, a fully implemented and robust UFDM program can be successfully designed and deployed.

1. INTRODUCTION.

This report is the fifth and final required deliverable submitted for ASSURE COE project A11L.UAS.43 UAS Parameters, Exceedances, Recording Rates for ASIAS. The project has several goals, but the overarching purpose of this research is to enable safe integration of Unmanned Aircraft Systems (UASs) in the National Airspace System (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.

This project designed and evaluated Flight Data Monitoring (FDM) for unmanned operations and integrate that data into the ASIAS system. In addition, this project will identify current Unmanned Aircraft Systems FDM (UFDM) capabilities and practices, including refresh/recording rate and robustness, and develop guidance for a UAS FDM standard. The UAS community has specific and disparate needs relative to manned aviation, such as the need for strong cyber-security measures regarding telemetry streams and the storage of sensitive UAS flight data. This project will also seek to identify the best governance practices regarding the use and research involved with UAS flight data. The proposed 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.

This project will also identify UFDM events, including event definitions and exceedances, using the normal ASIAS techniques. Future phases of this project will include the actual deployment of a UAS database which interfaces with ASIAS similar to other safety reporting programs.

This report describes the current state-of-the-art practices and protocols as it pertains the UFDM. Five aspects of UFDM will be described. These include: flight data recording and retrieval; long-term data storage; data analysis and hazard identification; and feedback to operators. This report will identify data types that are common across commercial, general, and unmanned aircraft and will also identify data types unique to unmanned aircraft. Further, this report will provide analysis for each unique data type as to why it is important, and will describe any additional events that can be detected at minimum collection rates both for generic and unique UAS mission segments.

II. BACKGROUND

FDM, also known as Flight Operations Quality Assurance (FOQA), is a process whereby an aviation organization systematically collects flight data generated by onboard flight recording devices; aggregates all of that flight data into a central repository; rigorously analyzes that data in an attempt to proactively identify latent safety hazards and hazardous trends; and, ultimately adopts new mitigations in an effort to manage the risks associated with the latent hazards. A key element of FDM is the ability to proactively predict hazards rather than reacting to accidents or incidents.

FDM is one of several proactive and predictive safety Quality Assurance (QA) techniques included in Safety Management Systems (SMSs). FDM has been widely adopted among most airlines throughout the world. The overwhelming consensus among those organizations using FDM is that such endeavors harbor great efficacy in reducing operational risk.

The most data-intensive underpinning of SMS involves safety risk management and safety assurance. The International Civil Aviation Organization's (ICAO) Safety Management Manual (SMM) classifies safety assurance initiatives into three broad categories: reactive, proactive, and predictive (ICAO, 2009). Examples of traditional reactive safety assurance measures include accident and incident investigations. Certainly, after an organizational accident or other serious incident occurs, the organization should seek to identify the root causes that ultimately led to the breakdown in the hope that such conditions, known as accident precursors, could be avoided in the future. While necessary, the investigation is classified as reactive because the organization is clearly only acting after the event occurred. In essence, the organization has to pay a significant price in terms of accidents or incidents to ascertain the accident precursors.

Proactive SMS program measures include reporting and monitoring. With reporting, an organization seeks out safety reports generated by the users and operators within their system. In order to encourage maximum participation, a hallmark of reporting systems includes the concepts of de-identification, anonymity, and/or immunity. Moreover, in exchange for an individual user's participation in submitting a safety report, the organization promises not to take sanction that individual. More contemporary safety reporting systems have also included protections from the applicable regulatory agency (in addition to protections for the operator). Unfortunately, reporting still relies solely on human involvement and the willingness of an individual to honestly and accurately relay information. Accordingly, it is likely that some accident precursors go unreported and are often only discovered with a reactive investigation.

One of the most utilized reporting systems is the Aviation Safety and Reporting System (ASRS) administered by the National Aeronautics and Space Administration (NASA). Created in 1975, the purpose of the ASRS program is to obtain incident reports "from pilots, air traffic controllers, dispatchers, cabin crew, maintenance technicians, and others…to identify deficiencies and discrepancies in the National Airspace System and provide data for planning and improvements…" (ASRS, 2018, pp. 4-5). The FAA uses aggregated data obtained from the ASRS program "to take corrective action and remedy defects or deficiencies within the NAS" as well as facilitate NAS planning and improvements (FAA, 2011, p. 1). In 2017, aviation stakeholders submitted a total of 94,302 self-reports to the Aviation Safety Reporting System (ASRS, 2018).

To encourage stakeholder reporting, the ASRS ensures confidentiality and anonymity to reporters and other involved parties (FAA, 2011). Moreover, the FAA has codified its commitment to protecting ASRS reporters, stating in 14 CFR:

> The Administrator of the FAA will not use reports submitted to the National Aeronautics and Space Administration under the Aviation Safety Reporting Program (or information derived therefrom) in any enforcement action except information concerning accidents or criminal offenses which are wholly excluded from the program. (§91.25)

Furthermore, the FAA (2011) considers the submission of ASRS reports to represent a constructive attitude of the reporter that will ultimately prevent future violations. As such, the agency commits to remit punitive enforcement action or civil penalties, provided the violation:

- Was inadvertent and not deliberate,
- Did not constitute a criminal offense or discloses a lack of qualification or competency,
- The individual has not committed a violation within the previous 5 years of the occurrence, and

- The individual provides evidence that a[n] ASRS report submission was made to NASA within 10 days of the violation (or when the individual became aware of the violation). (p. 9)

Although reactive measures such as accident and incident investigations and safety reporting are vital components to safety risk management and safety assurance, an additional source of data that generates objective reports can increase reliability to identify hazards prior to an incident or accident occurrence. These initiatives, widely known as FOQA programs, have been in existence in various formats since British Airways established a flight data program in 1962 (Fernandez, 2002). FOQA reports are typically data intensive and allow substantial analysis and aggregation with the hope of discovering accident and incident precursors.

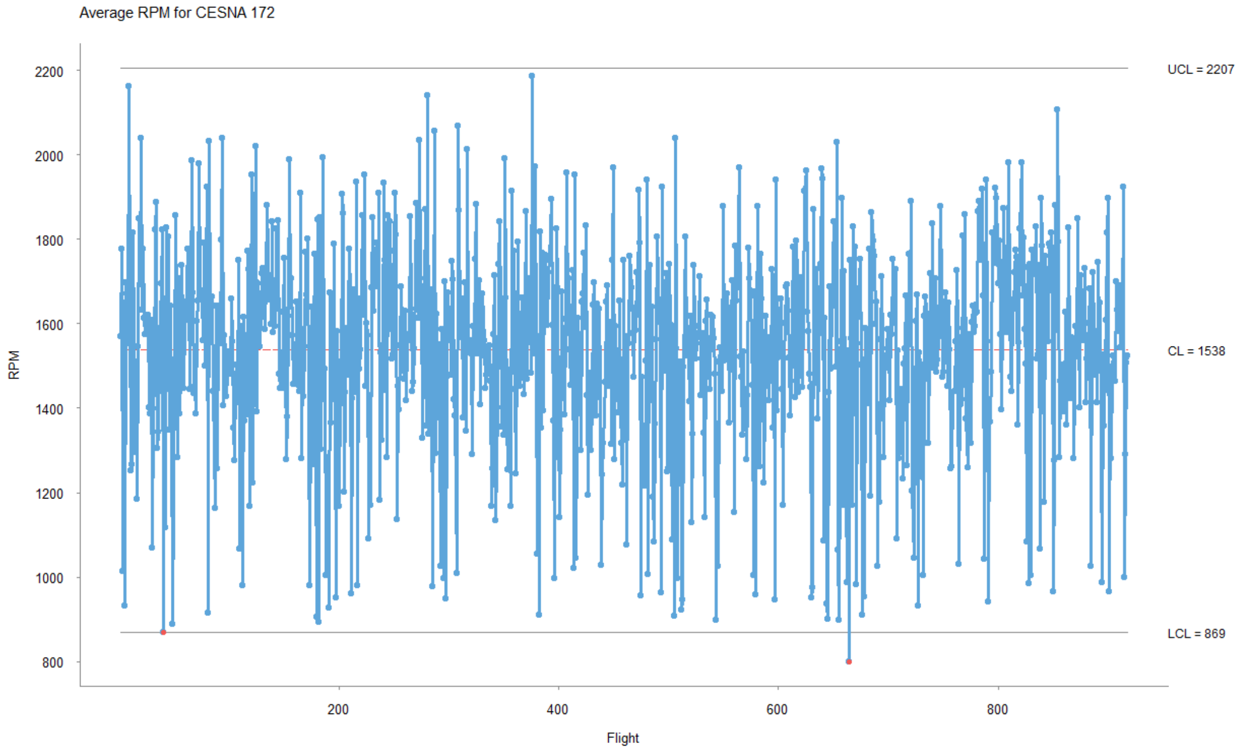
III. BENEFITS OF UFDM

Safety programs like FDM and FOQA have roots in the ideas surrounding quality and reliability theory. Any set of guidelines for controlling UAS parameters must fall under some overall quality-based framework. This section outlines some basic terminology and best practice for managing quality and reliability in industry and will describe how these ideas can be utilized when designing a control system for UAS parameters. A brief historical overview of quality and reliability management over the last 100 years will be followed by summaries of some of the major concepts, techniques, and definitions for different aspects of quality management. An overview of quality management in the aircraft industry will describe specific aspects of quality management applied to the design of aircraft and which are relevant to the design of UASs.

While the idea of setting standards for quality dates back many centuries (Saleh & Marais, 2006), quality and reliability management as a modern discipline was developed from the early part of the 20th century. The birth of modern mass production, combined with increasingly sophisticated empirical and statistical methods led to the creation of new techniques to help improve product quality and reliability. In addition, the two world wars in the first half of the 20th century led to military demand for structured methods to increase manufacturing and operational efficiencies (Barlow, 1984). For example, in World War II, the United States military, when utilizing increasingly sophisticated methods of electronic warfare, found that failed vacuum tubes were the most common source of equipment failure (Coppola, 1984), resulting in a drive to increase production quality and reliability for these items.

Statistical Process Control

Early statistical work on reliability, included pioneering work by Shewhart (1931), who developed the "Shewart Chart", which determines if parameters deviate from a set or hypothesized standard, relative to underlying statistical variability. This type of "process control" is used for monitoring parameters over time and warning of significant variations from the standard, which lay outside of control limits, defined by standard deviations (usually 2) from the process standard. It can be combined with corrective systems during "process regulation" (Box, 1993). For example, in a UAS setting, a flight stability parameter may deviate from its required value and some mechanical process could be employed to bring the parameter back in line. The concepts of statistical process control have been introduced into general management practice via the Six Sigma Method (Klefsjö, Wiklund, & Edgeman), where processes are controlled so that the number of process defects are limited to those outside 3 standard deviations on either side of the process (roughly 3.4 per million). An example of a flight parameter control chart is given below, which monitors the average Revolutions Per Minute (RPM) for flights of a Cesna 172 over time. Here, several flight points lie outside the control limits and these could be investigated further.



In addition to the basic Shewhart chart, many more control chart variants have been introduced over the last 50 years. For example the Cumulative SUM (CUSUM) chart tests the sum of deviations from the process standard. Exponential weighted moving average (EWMA) charts, calculate the process mean, weighting recent observations more strongly than less recent observations. Both of these types of charts can be used to calculate small shifts in process means. Most of these methods can be found in standard references on statistical quality control (Montgomery, 2012; Oakland, 2007). Specific adaptions to process control techniques have been made for aircraft control systems. For example, Samara, Fouskitakis, Sakellariou, and Fassios (2008) describe a method that detects sensor faults, by testing if variation in process data is significantly above expected norms.

III. A. Experimental Design and Statistical Quality Control

Statistical experimental design methods are utilized to help determine optimal parameter settings for a system. In contrast to process control methods, which are used to evaluate deployed, working systems, experimental design methods are used to help determine optimal parameter settings during the system design phase. For example, experimental design may be used with the stability of a UAS vehicle as the dependent variable and used to find the best combination of values of different feature variables (e.g., wing length, thickness, shape) in order to optimize stability. These methods are often referred to as “off-line” quality control methods. Both statistical process control and experimental design can be combined in a feedback loop, with experimental design used to optimize parameter settings and process control used to test the performance in a real environment, where results are fed back to the design stage to further optimize the design.

Statistical experimental design methods for quality purposes grew out of the general work on experimental design, pioneered by Fisher (1935). They became widely used in industry in the post-war period. Imperial Chemical Industries (ICI) released a handbook on experimental design for industry, which incorporated "response surface methodology", designed to help optimize parameters to maximize system performance (Stewart, Mullins, & Drew, 1996). Taguchi, developed a set of designs that could test performance with a much smaller number of experiments than with a traditional factorial (every combination of every parameter) design (Taguchi, 1985). These designs are fit into an overall quality framework, which includes parameter design stages, designed to minimize performance variation (e.g., maximize the flight stability for UASs) and tolerance design, where parameter tolerances are set to maximize the trade-off between performance tolerance and cost (Kackar, 1985). For example, an over-engineered product with low error tolerances may cost too much to make, while a product with error tolerances set too high could incur long term costs due to errors and unreliability. Here, the use of statistical quality control can be used in aviation design, to maximize affordability, which is defined in as the ratio of operational effectiveness to the cost of achieving this effectiveness, where operational effectiveness is defined as a weighted sum of capability, survivability, readiness, and dependability (Mavris, DeLaurentis, Bandte, & Hale, 1998).

Experimental design has been used in the aircraft design wind tunnel force balance tests, where components of force such as normal, axial, side force, roll, pitch and yaw are applied to an aircraft (Parker & Finley, 2007). Such tests can be applied in a similar manner for UASs. In general, the use of statistical methods in aircraft instrument calibration, can result in significant savings in both calibration time and costs and is core to the development of the fault tolerant control systems (Zhang & Jiang, 2008) that are required for modern aviation design.

A wide range of experimental design methods and procedures have been developed over time. These include designs ranging from full-factorial designs to designs for gaining insight into parameters from only a few experimental runs. Increasingly sophisticated response surface methods have been developed to optimize parameters. Several modern textbooks (Lawson, 2014; Montgomery, 2017; Myers, Montgomery, & Anderson-Cook, 2016) cover the full breadth of the field.

III. B. Quality Management and Total Quality Management

Much of the impetus for modern quality control and reliability theory was provided by Edwards Deming, who took the statistical quality control work of Shewhart and built a managerial approach to quality control that encompassed statistical methods and more general management practices, such as the fact that quality control should be the responsibility of all workers. Deming's work was particularly well received in Japan and helped to inform the post war Japanese push for quality control in manufacturing (Tsutsui, 1996). This work was built upon by quality experts such as Juran (Juran, 1974) and Ishikawa (Ishikawa, 1985), who created a range of tools for quality management. Ishikawa's "fishbone" diagram is a visual representation of the specific "causes" of a particular quality problem, ordered by category. For example, if a UAS was prone to stalling, the possible contributors (ex, wing shape, lift, stability, etc.) would be organized by category to give insight into the problem. Pareto diagrams aim to separate the important causes of a quality problem from the unimportant causes. For example, a wing design issue may have a major impact on stability of a UAS, while the type of mounted camera, would only have a minor impact.

In combination, these tools contribute to the broad management philosophy of TQM (total quality management) (Hackman & Wageman, 1995). TQM is both a set of tools and an overall managerial philosophy. TQM can incorporate more managerial constructs, such as management commitment, customer focus, design quality management, information usage, and employee empowerment and involvement (Ahire, Golhar, & Waller, 1996). Additionally, TQM can incorporate more engineering based constructs such as safety management (Kontogiannis, Leva, & Balle, 2016), which includes process monitoring, risk analysis, and reliability engineering (Zio, 2009) activities. Given the variety of techniques that fall under the banner of TQM, there are a wide variety of TQM philosophies and implementations. However, there is empirical evidence that firms with at least some commitment to TQM outperform firms who do not have such a commitment (Powell, 1995).

III. C. Observational Data Analysis

The methods of "statistical experimental design", as per the name, are experimental. There are some situations where experimental methods have limited usefulness. For example, flight datasets on commercially operated UAS contain a wealth of information on UAS performance and experimental control of parameters are not possible. Similar constraints apply to any third-party flight dataset. In these instances, an analyst may still wish to examine the relationship between different parameters and performance. Here, the analyst may perform an "observational analysis", using a variety of statistical methods, to examine the relationship between different observed variables. A summary of a few of the most popular of these methods provided in the Table 1.

To demonstrate how observational UAS data can be analyzed, an observational analysis was carried out on the Cesna 172 data described previously. In the dataset, the turning error of the plane was evaluated for each flight. Each flight was classified as "Aligned", "Large Overshoot", "Large Undershoot", "Small Overshoot", or "Small Undershoot". A multinomial logistic regression model was built with the turning error as the dependent variable and the time of the turn, the light conditions (day or night), the wind component, the mean airspeed, the heat differential, the runway (from five possible runways), and the mean RPM as the independent variables. The analysis was run using the "multinom" function from the "nnet" package in the statistical software package R. A summary of the variable significances is given Figure 1.

Method	Description
Linear Regression	Linear regression is used when there is a dependent variable (e.g., fuel economy) which can be explained by a number of independent factors (e.g., average flight speed, altitude, number of turns, etc.) Regression analysis is a mature topic and a range of modelling tools can be used to deal with problems such as correlated variables, unobserved variables and missing data. A classic text is Draper and Smith (1998). A modern book focussing on computational implementation is Faraway (2016).
Structural Equation Modeling	Structural equation modelling can be used for modelling more complex causal relationships than those modelled by standard regression. For example, an analyst may wish to understand the impact of average speed on fuel economy. However this relationship is influenced by wind speed ,the fuel grade, and the UAS aerodynamics, which all need to be incorporated into the model. Kline (2015) gives a practical overview of SEM modelling.
Logistic Regression	Logistic regression is used when there is a dichotomous dependent variable. For example, UASs have been used to drop medical supplies in disasters (Thiels, Aho, Zietlow, & Jenkins (2015). Here the success of dropping a medical package in a specified drop zone could be the dependent variable and the flight characteristics could be independent variables. Logistic regression is described in Hosmer, Lemeshow, & Sturdivant (2013). Multinomial logistic regression expands logistic regression to dependent variables with multiple categories. For example, UAS drops could be classified as {correct, undershoot, overshoot, drift to left, drift to right}.
Cluster Analysis	Cluster analysis is used to group data entities based on the similarity/dissimilarity of observations. For example, cluster analysis could be used to group UAS operators by flying characteristics or group UASs by performance characteristics. Cluster analysis is often used in combination with visualization to help analysts understand patterns in the data. A classic text on the issue is Kaufman & Rousseeuw (1990) and a range of modern implementations of clustering analysis algorithms are listed in James, Witten, Hastie, & Tibshirani, R. (2013).

Table 1. Statistical and Analytical Methods.

```
Analysis of Deviance Table (Type III tests)

Response: Turn.Error
                     LR Chisq Df Pr(>Chisq)
Time.in.Turn           46.309  4  2.124e-09 ***
Light.Conditions        6.776  4    0.14823
Wind.Component         37.954  4  1.145e-07 ***
Mean.Airspeed          25.288  4  4.404e-05 ***
Height.Differential     9.634  4    0.04706 *
Runway                 47.204 16  6.330e-05 ***
Mean.RPM               13.213  4    0.01028 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 1. Analysis of Deviance.

In this example, the results follow standard statistical notation. Chi-Squared statistical tests based on likelihood ratios are used to test the significance of the independent variables. Lower p-values (Pr(>Chisq)) give more evidence that the independent variables are significantly related to the dependent variable, as opposed to the null hypothesis of the variables being unrelated. Taking a standard Type I error cutoff of 0.05, all the variables are significant except for the light conditions. While it may be intuitive that light conditions should have a strong effect on flight maneuver accuracy, there were very few night flights in the dataset, making it difficult to obtain evidence of significance. Figure 2 below shows the predicted turn probabilities across the values of the wind component and turn time independent variables (for each graph, the other independent variables are set to their mean values). The graphs in Figure 2 show that as the wind component and turn time increase, there is a much stronger chance of a large undershoot or overshoot.

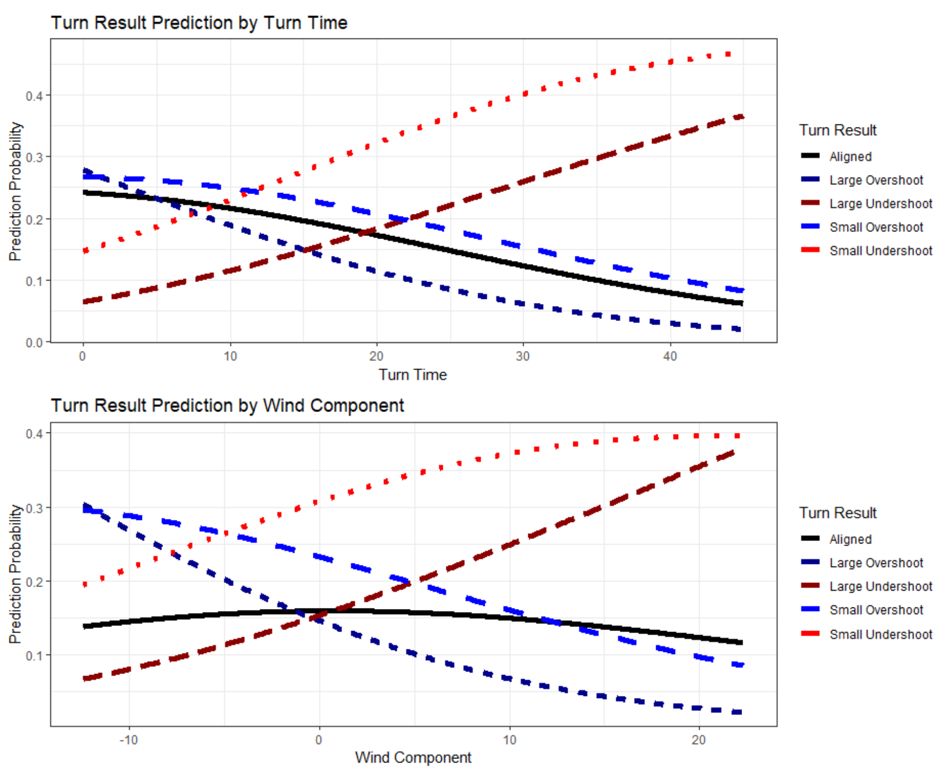


Figure 2. Predicted Turn Probabilities.

III. D. Summary of Quality Methodology

Statistical process control is used for online monitoring of flight parameters. Experimental design can be used to help optimize UAS design parameters with respect to safety and economic performance. With increasing amounts of UAS data becoming available, there is a need to analyze these data with respect to flight performance (stability, flight path accuracy, etc.) and economic performance (fuel economy). Observation analysis using statistical methods provides the tools to interpret and gain knowledge from these data. TQM provides an overall set of managerial processes for controlling the UAS design and implementation processes. It is proposed that these tools should be used together, with built in feedback loops. For example, experimental design could be used for UAS design, then on-line statistical process control and off-line observational methods could be used to analyze flight data. The results of these analyses could be fed back into the UAS design process.

III. E. SAFETY BENEFITS

Although FDM has been widely accepted as providing benefits to its operators (see Lacagnina, 2007; Callantine, 2001; Holtom, 2000; and Larder, 1999; among others), there are many obstacles that operators have to overcome prior to obtaining a successful program (Mitchell, Sholy & Stolzer, 2007). Several of the identified obstacles include cost of equipment, storage and software processing, lack of analytical capabilities, fear of public disclosure, fear of liability and operator/pilot privacy concerns. In manned commercial aviation, these obstacles have generally been overcome, while manned general aviation is also seeing some FDM growth (Lowe, Pfleiderer and Chidester, 2012).

III. F. FDM/FOQA Benefits

The FAA (2004) indicated in Advisory Circular (AC) No: 120-82 that, "…wide implementation of FOQA programs could have significant potential to reduce air carrier accident rates below current levels….The value of FOQA programs is the early identification of adverse safety trends that, if uncorrected, could lead to accidents."

Similar to FOQA in manned aircraft, as outlined in the FAA's AC, UASs will benefit from UFDM by reducing the accident rate with manned flights, which could be fatal and catastrophic. In particular, UFDM will identify near missed collisions between manned and unmanned aircraft as well as help monitor for other safety hazards. A key element in FDM/FOQA is the application of corrective action and follow-up to assure that unsafe conditions are effectively remediated.

For example, there have been anecdotal reports of crashes and many near missed crashes between UAS and maned flights. A thorough search of an FDM/FOQA database like the National General Aviation Flight Information Database (NGAFID) could be conducted by the project team to find out crashes and near crashes between UAS and manned flights, if UAS data were included in the database. The Airport Safety Database developed by Deborah Carstens (2018) could also provide information about manned flights with UAS near airports.

The second type of safety benefit is the crash and accident rate reduction between UASs. Although it may not be possible to determine the exact safety benefit in terms of the accident rate reduction at present, when used as a part of TQM described above, the safety benefit should increase.

III. G. Data Driven Safety Approaches

There are examples in manned flights where flight data are used to evaluate and improve the safety performance. For example, Marais (2018) developed safety risk performance algorithms to take general aviation flight data as input and identify hazardous phases during the flight. Helicopter Association International (2014) completed a project for the FAA to provide data driven safety analysis of rotorcraft FDM. Li, etc. (2016) presented the Gaussian Mixture Model, a data mining approach, to identify latent risks from FDM without specifying what to look for in advance. Chang (2014) developed a fuzzy logic model to monitor the exhaust gas temperature to diagnose the potential problem and abnormal conditions of turbojet engines. Clachar (2015) evaluated an Artificial Intelligence (AI) approach (Kohonen self-organizing maps, or SOM) and analytical techniques asynchronously to address big data in FDM database. She concluded "SOM identified hard landings and unsafe low-level maneuvers and that some approaches that were high, fast, and steep would be harder to detect by using traditional flight safety [analytical techniques]".

A data driven safety approach will enable researchers to utilize statistical methods, data mining techniques, and/or artificial intelligent algorithms to find out the risk and safety factors. This should hold true even among different types of UASs, manufactures, models with the same manufacturer (and even outliers of the same manufacturers and models), or different UAS operators (operated beyond the line of sight).

In the future, there will likely be an ever-increasing amount of UFDM data with data elements such as UAS flight 3D trajectories (using satellite-based tracking), speed, acceleration rate, deceleration rate, engine/battery temperature etc. Statistical analysis and AI algorithms could be utilized to identify the outliners of this data. Those outliers can then be forwarded to the corresponding manufacturer, operator and safety authorities. With the domain expertise, the corrective actions can be taken in future UAS operations.

One promising benefit from data driven endeavors is to develop new safety knowledge, which has previously not been readily available from UAS flight data. Oehling and Barry (2019) applied machine learning towards this direction with success. Their approach discovered flight abnormalities missed by the traditional system, detecting arrival phases for which no exceedance event existed.

III.H. Potential Real Time Safety Benefit

UAS fleets are diversified in terms of weight, engines, power sources, manufacturers, and communications. With a potential real time cloud implementation of a database, UAS flight data could be amalgamated into air traffic management systems and could monitor telemetry of all participating UAS platforms continuously. That information could prove critical to the safety of UAS and other aircraft and help avoid collisions.

Some UAS platforms are equipped with sense and avoid (SAA) technologies. The technology will "sense" the UASs and other surrounding targets and take proactive actions to avoid collision. For example, if a UAS will accelerate/decelerate at a constant rate of x, the x can be used to disseminate flight data to other UASs projected to be in the UAS's path. This will allow for a more accurate projection of future paths. The projected path information will be critical to collision avoidance systems.

Although peer-to-peer commination might be better in reducing communication latency for collision avoidance system, sometimes, that would not be possible (e.g., for privacy and security reasons, or incompatible peer-to-peer communications etc.). Real time UFDM can also feed the ground-based control systems. For example, the Ground Based Sense and Avoidance developed by MIT (2018) and Volpe (2018) provides remote UAS pilot with the manned traffic surrounding the UAS. The pilot has the information to take "sense and avoid" actions to avoid collision with manned flight. With real time UFDM, data from other UASs could also be provided, in addition to the manned flight information, improving the safety of remote operations.

III. I. Promote Safety through Training

To become a commercial drone pilot for UAS over 55lbs, the FAA (2019) requires a pilot license. To operate UAS beyond Visual line of sight (BVLOS), there is a waiver required by FAA (2019). When flying drone in controlled airspace (e.g., near airport), the flight must be pre-authorized by FAA (2018).

All these regulations and rules have been established to ensure a safe UAS operation. Similar to a new driver-trainee with an automobile on highways, it is reasonable to train the commercial UAS pilot to fly safely in the NAS while monitoring their performance, at least at the beginning of their commercial pilot career. In manned pilot training, Ladenburg (2011) reported Embry-Riddle and the University of North Dakota has used FDM in training their manned pilots.

Trajectories of the flights and surrounding areas in UFDM will provide an ideal data set to analyze pilot training and help correct learning gaps to avoid the hazardous activity, or coach operators to perform additional maneuvers to reduce unnecessary exposure to potential safety hazards.

III. J. Framework of Economic Benefits

While the safety benefits are tangible and understandable, the value of increasing safety is difficult to quantify. The expense of an organizational accident is extreme when one considers the potential loss of life and destruction of property. The economic benefit is also associated with the particular make and model of UAS and/or aircraft involved. In one example, Cavka and Cokorilo (2012) detailed Airbus A320 accidents, severity and fatalities. The costs are both direct and indirect. As data becomes more available, quantifying these costs for UAS operations will become more exact.

III. K. Economic Benefit of FOQA in Manned Flights

As discussed above, when an aircraft experiences a crash, an accident investigation team will normally respond. As information is learned from the accident, a feedback loop can be formed which will help identify whether UFDM could have found latent hazards within the database. This could help avoid similar accidents in the future, and once there is any avoidable crash, economic benefits can be realized.

In addition to accident avoidance, more economic benefits might come from the realization of saving the cost from damages to aircraft, especially with a high-profile event like a manned-unmanned collision. There are similarities between this type of accident and the near-missed incidents. There could also be data fusion with wildlife abatement databases, which in turn could drive down the potential for these types of accidents as well.

III. L. Preventive Maintenance Function

The preventive maintenance function that would come with a Maintenance Operations Quality Assurance (MOQA) program has potential for cost savings. As a part of total quality control described earlier, system outputs and other parameters can be statistically analyzed and outliers can be easily identified. With advances in AI, there are more prospective opportunities to identify UAS for mechanical irregularities.

For example, when a UAS's acceleration at takeoff is significantly slower than those with similar models, it is likely there is engine/battery problems. If the preventive maintenance confirms the malfunctions and repairs the UAS, a potential system failure and ensuing accident could be avoided.

III. M. Insurance Adjustments

Almost all aircraft owners and operators utilize insurance as a method of managing their risks. Insurance is generally a data-driven industry, relying upon metrics and other actuarial assessment in order to quantify a client's risk and consequently their costs. It is conceivable that an organization employing a UFDM program could use their data as empirical proof of operational practices; and, if acceptable to the insurance vendor, could further use this data to obtain a premium reduction. This has been demonstrated in the commercial manned flight industry, but due to the lack of a UFDM database, the authors are unaware of any existing insurance premium adjustments within the UAS industry for UFDM at the present time.

III. N. Conclusion of UFDM Benefits

UFDM provides many benefits for its participants. From a safety perspective, it offers a chance to proactively and predictively learn about potential accident precursors before an actual accident occurs. Even with unmanned flight, there is potential for loss of life from other manned aircraft operating near UAS flight paths. Economically, the loss of platforms and sensors can be substantial.

UFDM and similar safety initiatives have their roots in TQM and consequently, SMS programs. In high-consequence industries like aviation, the principles and analytical capabilities of TQM, SMS and UFDM should be utilized to become more fully informed of potential safety hazards. Risk reduction should ensue from these activities.

Operators have to overcome their very real objections to participating in UFDM. These objections could include cost, lack of technical expertise, privacy and fear of the unknown. While these represent concerns, other facets of the aviation industry have shown that these can be overcome; and consequently, everyone benefits from such programs.

IV. 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). This data is known as telemetry, and so long as a digital recording of that telemetry is made and is available for retrieval, that data can be used for 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 an UFDM system.

Small unmanned aircraft systems (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 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 A).

Figure 3 illustrates the telemetry recording capability using a publicly-available, manufacturer-created data reader called CsvView (CsvView/DatCon, n.d.; CsvView Manual, n.d.). The CsvView program decodes and displays telemetry datasets in a graphical user interface, including GoogleEarth and color-coded data graphs. In the telemetry depicted in Figure 1, the red line indicates the sUAS flight path, and green line indicates position of the remote controller. The sUAS home point (launch point) is indicated by the "H" symbol and the unmanned aircraft is represented by the "A" symbol. Flight telemetry is time synced to other data signals, and the operator can drag the aircraft along its flight path to see correlated system status or other time-indexed signals at the respective point during the flight.

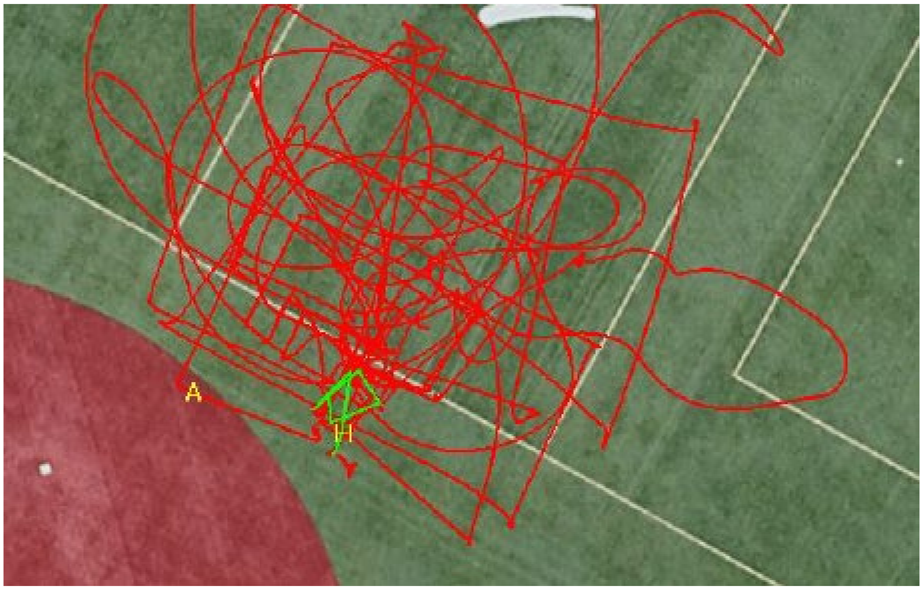


Figure 3. Sample sUAS flight telemetry derived from a DJI sUAS.

Figure 4 shows a sample time-indexed signal set for motor speed. The time index time is located on the bottom x-axis, with the zero point representing the aircraft launch time, with data to the left of the zero point representing pre-launch information. Note the vertical bar labeled 1,069.94—this indicates the current selected time and is correlated with the aircraft telemetry data. Specific collected data points can be overlaid on one or more *signal player* charts to evaluate flight information. Additionally, this data can be exported into other formats, such readable by common tabular programs, such as MS Excel. Other online resources, such as Airdata UAV (2018) are also capable of reading and interpreting DJI's formatted (.DAT) files.

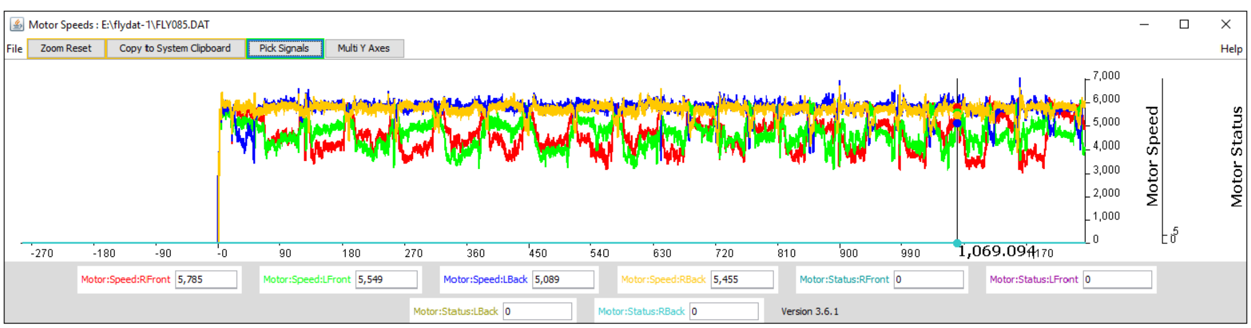


Figure 4. Sample time-indexed signal set for individual motor speed.

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 application 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). Data architecture and descriptions for Yuneec platforms are presented in detail in Appendix A.

In some cases, platform data may be extracted from common command and control sources, such as the Piccolo Command Center Ground Station. 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.

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[1], that data are unable to be shared either in this report or publicly as it has been determined to be Controlled Unclassified Information (CUI). Although these data cannot be shared in this report, several characteristics of that 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 (FDR) or Quick Access Recorder (QAR) 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.

While the platform census revealed widespread availability of UAS flight data recording sources, the research team also encountered several challenges. It is notable that data accessibility from the UAS was at times problematic. In most cases, data extraction required the use of proprietary software and yielded data files that required decoding or formatting conversion to access usable flight data. Data extraction methods were generally inconsistent between manufacturers and in many cases the process was somewhat cumbersome. These impediments may detract UAS users—particularly those operating small UAS—from furnishing datasets to a safety database. Additionally, the recording capabilities and captured parametrics of small UAS varied widely between manufacturers, and in some cases between platforms, as well. While almost all platforms contained some level of basic telemetry (including GPS location, altitude, speed, heading, etc), the remainder of recorded flight data varied considerably. These inconsistencies may make data comparison between platforms particularly problematic. It is also notable that in most cases flight recording capability for small UAS platforms is generally implemented for engineering product troubleshooting and fault detection—not safety or FOQA purposes. Perhaps not surprisingly, there is limited source documentation that codifies recorded parametrics, recording rates, measurement units, and the methodology of which various data points are derived.

[1] This data was generated while the aircraft were on corporate training flights with the location information of the aircraft completely redacted and removed for security reasons.

Appendix A also includes a column denoting whether the data type is specific to UAS platforms. Although both commercial and general aviation have robust and sizeable flight data types available, there are some data that are specific to UAS; and, in some cases, the amount of information available for UAS data types is much more robust and encompassing compared to commercial and general aviation.

One example of available data unique to UAS is battery metrics. With commercial and general FDM, battery metrics are generally limited to capacity and temperature. Given that the battery for a UAS platform is the de facto powerplant for the aircraft, UFDM battery metrics are more robust and numerous. For example, individual battery cell monitoring is common. Current and dissipation is also commonly monitored. Maximum and minimum capacity are also sometimes measured in UFDM

As UAS platforms increase in complexity, they sometimes have the ability to amalgamate battery health with other safety concepts. For example, some UASs have the ability to store their "home" position, or geometric position from where they took flight. When the battery reaches a critical low point, a Return to Home (RTH) function may trigger, which causes the UAS platform to return to the exact place it took off. All of this requires unique data recordings which very well may be germane to UFDM functions.

Another data capability unique to some UAS platforms is the ability to monitor senor packages placed on the aircraft. The most common sensor package found on UASs is a camera. Some UAS telemetry datastreams and data recordings will monitor sensor status, gimballed platform status (if installed), and possible sensor malfunction or loss of power status messages. From a UFDM safety perspective, monitoring sensor status should be included in the database if possible as sensor malfunction could bring about hazards to flight. One example of this would be a sensor overheat or fire situation. This type of situation could easily cause or help cause an accident. Accordingly, sensor data should be included in any UFDM database.

Appendix B depicts over 200 flight data events commonly found across commercial and general aviation as part of a FDM program. Many of these data events are not topical to UFDM. Appendix D denotes a basic list of parameters needed for a UFDM solution. It should be noted that given the data available as indicated in Appendix A, most UAS platforms will likely be able to produce data that far exceeds the number of data types depicted in Appendix D. The parameters listed in Appendix D however indicate the minimum data types needed to develop most FOQA/FDM tools and analytical techniques. Any future database or data collection repository should be built to accommodate any and all data that a UAS platform may generate, including that data which goes beyond those listed in Appendix D.

The basic level of data needing to be collected to provide some UFDM functionality includes time, position and altitude. Almost any UAS platform that utilizes GPS will be able to provide this data. GPS systems are capable of highly accurate measurements with GPS time being particularly accurate to less than a 40 nanosecond (billionth of a second) error (GPS.gov, 2019).

From this basic level of data, many other parameters could be derived. For example, ground speed could be calculated by an algorithm that measures the time it takes the UAS platform to travel between two geospatial positions. Likewise, direction of flight could also be calculated. There are other important data types that can also be derived from this basic level of data.

The refresh rate listed in Appendix D also depicts the minimum level needed for fully-functional UFDM analysis. With many parameters, updates of at least six times per second (6 Hz) are the minimally needed rates for FDM analysis. Other parameters don't need to update as often;

however, any future UFDM database should be able to log data asynchronously so that any level of refresh rates can be accommodated.

V. LONG-TERM DATA STORAGE

Currently, there is no national or internationally accepted data standard regarding UFDM flight data. In addition, long term storage for UAS flight data is not available to aggregate across all platforms. Most data that are currently stored is done so by individual companies, and is generally not shared with other operators at this time.

For manned aviation, there are databases that are used to aggregate flight data. For example, many commercial aviation operators send their flight data to MITRE, who on behalf of ASIAS, will collect and analyze the data for industry-wide trends (MITRE, 2013). For general aviation, the National General Aviation Flight Information database (NGAFID, pronounced "N-G-A-FID") allows public sharing of flight data from operators throughout the world (NGAFID, 2019).

There are several issues that may be problematic when envisioning how a national data repository could be created. One issue that seems to be very specific to UFDM is import and export regulations and laws. Because UASs were initially developed for military use, the civilian industry has had to take extra precautions to ensure any UAS activity is in compliance with these rules, including the telemetric data generated by UAS flights. Although this may impact military operations far greater than civilian operations, forecasting into both the short- and long-term periods, UAS military use will continue to be a large contributor of flight activity within the National Airspace System (NAS).

Another point that is generally debated revolves around the question of whether all of the data that can be recorded should be stored for UFDM purposes. The authors of this report would strongly recommend that all data should be recorded and housed. Regarding flight data, new analytical techniques are continually being developed, and data that may not be important at present could prove invaluable as technology improves.[2]

VI. DATA ANALYSIS AND HAZARD IDENTIFICATION

There are flight data analysis software platforms available for UASs. The majority of these software packages offer mapping, reanimation and some maintenance prediction. Traditional FOQA software platforms, such as General Electric/Austin Digital's eFOQA (General Electric Aviation, 2019) or Aerobytes' FDM (Aerobytes, 2019), will help analysts identify safety trends, track exceedances, and conduct a deep analysis into specific safety issues. The classic techniques analyzed from within an FDM system include: exceedance monitoring, trend analysis, benchmarking, policy and procedures assurance, research studies, accident investigation and maintenance quality assurance. These capabilities are not yet found within UFDM.

The NGAFID allows for the importing and analysis of any manned flight data. Although it has not yet been used to analyze unmanned flight data, it would be relatively easy to accommodate such data within the database. The analytical tools would likely have to be

[2] One example of this is predictive maintenance using flight data. In recent years, many commercial operators have learned to use onboard sensor data to make predictions about the health of the aircraft. The data used for this would not have been as important in previous uses of FDM analysis.

developed specific to unmanned as the manned tools may or may not be appropriate. Ideally, similar to the Commercial Aviation Safety Team (CAST) and General Aviation Joint Steering Committee (GAJSC) methodology, the Unmanned Aircraft safety Team (UAST) or some other governing body could direct the efforts into unmanned tool development. Figure 5 is a screenshot of the approach analysis tool calculated by the NGAFID.

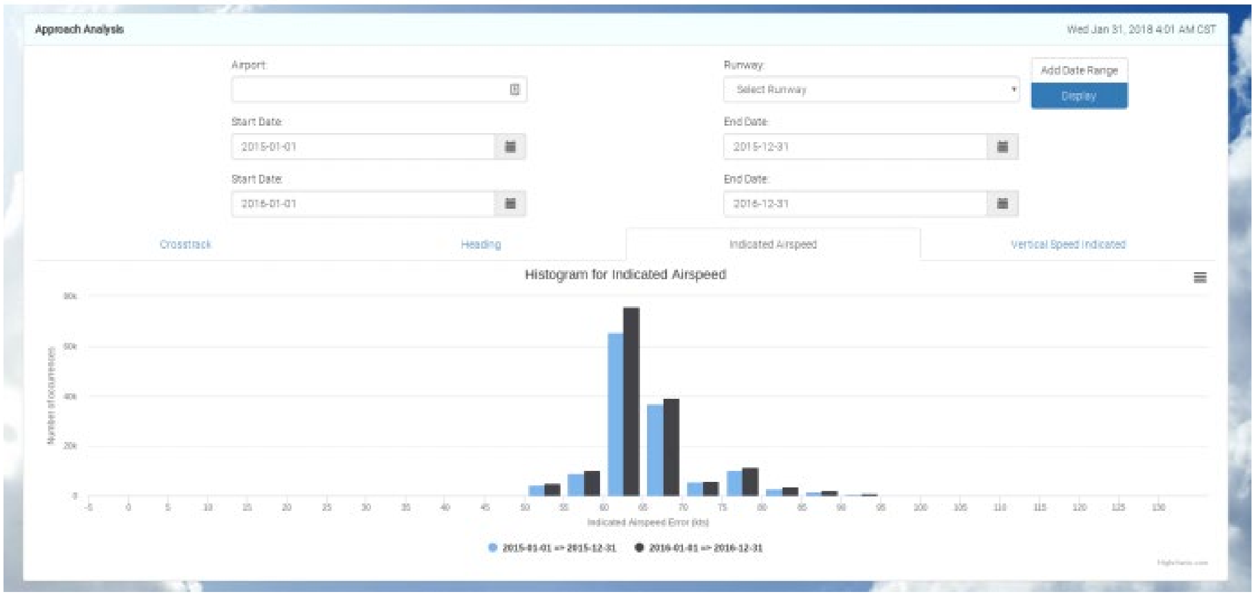


Figure 5. Approach analysis screenshot from the NGAFID.

VII. EXCEEDANCES AND PARAMETER RANKING

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 B 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.

VIII. FEEDBACK TO OPERATORS

Although it is imperative to have useful safety data collection capabilities coupled with the ability to analyze data, it could prove futile if providing safety data feedback into the system is unavailable. Since there are no nationwide storage capabilities at present, there is no data feedback to operators. However, it is very possible that individual organizations and operators do utilize their own data to review and validate previous operations. While this may prove useful, it still does not rise to the requirements of a robust and working FDM program.

IX. UNMANNED AIRCRAFT SAFETY TEAM DATA GAP ANALYSIS

In collaboration with MITRE, UAST was formed to study safety aspects related to UAS flight. This team was formed in a similar fashion as the Commercial Aircraft Safety Team (CAST) and the General Aviation Joint Steering Committee (GAJSC). All of these groups serve the purpose of lessening the aircraft accident and fatality rates associated with their sector of study. In the case of the UAST, that group focuses on UAS safety.

One initiative undertaken by the UAST is the formation of a Data Committee. This committee "examines all sources of available UAS data to be used for safety analysis. The Data Committee will consider which data might be important, how to normalize the data, and how to begin making it available for analysis. Focusing on establishing a relationship with UAS manufactures to develop a secure viable method for them to share data with the UAST as to fill the data void within the UAS industry is the primary initiative for the UAST's Data Committee" (UAST, 2019, p. 1).

One product produced by the UAST Data Team was a gap analysis progress report (Walsh & Feerrar, March 20, 2018). As part of this report, five different UAS data categories were identified. They included Digital Flight, Flight Mission and Performance, Safety Reporting, Sensor and Environmental. Further investigation ensued regarding the digital flight data and safety reporting data categories.

The gap analysis report also analyzed what data and data types would be important to collect. This was primarily accomplished by looking at end-state failures (e.g., throttle failure, attitude reference failure, human error). From these end states, a data map of critical information was described, which identified gaps in current unmanned data capabilities.

X. TEST SITE DATA COLLECTION

One area that may prove useful for future UFDM initiatives involves flight data collection at the FAA authorized test sites. All test sites are required to collect data, including flight data if available, of flights operating within their boundaries. This data is generally used for the purposes of validating the mission and maintaining adequate safety margins; however, the data collection is not yet used in the analysis of classic FDM facets (e.g., exceedances, trend analysis, benchmarking). This current ASSURE project will coordinate with test site data collection efforts where possible.

XI. APPLYING ASIAS METHODOLOGY

ASIAS is a system that uses processes and protocols to assess risks affecting different aspects of the aviation community and industries throughout the United States. It is governed by an amalgam of government and industry representatives. The most mature ASIAS constituency currently involves commercial aviation, but the general aviation and helicopter communities also have burgeoning ASIAS initiatives.

By almost any measure, the commercial ASIAS program, currently overseen by the Commercial Aviation Safety Team (CAST), has been very successful. National Transportation Safety Board (NTSB) metrics indicate only one 14 CFR 121 fatality among United States airlines since 2008. While there may be additional reasons for this low fatality rate within commercial aviation, many have noted the accident and fatality rates have dropped along a corresponding timeline to align with CAST recommended mitigations.

At the heart of ASIAS and CAST, accident categories have been used to determine which accident causes have resulted in fatalities.[3] These categories are determined using data from NTSB accident reports. These reports are analyzed for frequency and severity. As an example, Loss of Control (LOC) events are part of an NTSB-defined accident category. The number of times this accident category appears relative to other accident categories can be used to determined the frequency of events. Severity can be determined by using fatality data. A compilation of accident categories rank ordered by frequency and severity can be visually arranged in a Pareto diagram to help the CAST determine where they should focus mitigations.

In unmanned operations, there have been no documented or officially reported fatalities. Additionally, a central database or repository of serious UAS incidents or events has not yet been established. So, in order to build a similar CAST-like methodology with UASs, another accident-risk category identification and selection technique must be utilized. For the purposes of this project and report, since identification and selection of these UAS categories is beyond the scope of this project, several quasi-categories were generated and used to demonstrate how a future CAST-like methodology might interact between UASs and UFDM. These categories are labeled as quasi because based upon the knowledge of this research team as well as some seasoned researchers in the industry, their inclusion in an ultimate UAS Pareto diagram could reasonably be expected to occur even though there has not yet been a formalized process for category identification.

The categories and their associated definitions selected for this exercise 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

[3] More recently, due to the reduction in fatalities, CAST now focuses more on incident categories.

person on the ground; and Collison in Air – a collision between the UAS and another object while airborne.

In Tables 2 through 8, essential FDM data parameters are listed for each of these categories. These tables depict the data that should be collected in order to help track potential hazards and track any CAST-like mitigations. The type of parameter, refresh rate, the ability to be derived and whether or not that parameter would be required for the associated category are also listed. The listed refresh rate is the minimum number of times per second that particular data parameter should be recorded. Some parameters may be derivable using a combination of required parameters and outside data sources. One example of a derived parameter would be height above terrain. If the altitude above sea level is known (perhaps through GPS altitude measurements), a derived height above terrain could be calculated by measuring the difference between the height above sea level and the known terrain from an outside terrain database. This type of derived parameter is used widely in FDM databases such as the NGAFID.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Battery Voltage	1	No	Yes
Battery Percent Remaining	1	Yes	Yes
Battery Dissipation Rate	1	Yes	Yes
Altitude (MSL)	6	No	No
Height Above Terrain	6	Yes	No
Battery Low Return to Base	1	Yes	No
Battery Temperature	1	No	No
Flight Time Elapsed	1	Yes	No
Loss of Telemetry	1	Possibly	No
Cell Data:			
Voltage	1	Yes	No
Remaining Voltage (Gap)	1	Yes	No
Temperature	1	No	No

Table 2. Loss of Battery FDM Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes

Table 3. Loss of Command and Control Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Motor RPM	6	No	Yes
Motor Volts	6	No	No
Motor Current	6	No	No
Motor Watts	6	No	No
Speed	1	Yes	No
Vertical Speed	1	Yes	No
Roll	6	No	No
Yaw	6	No	No
Pitch	6	No	No
Acceleration (3-axis)	6	No	No
Battery Voltage	1	No	No
Altitude (MSL)	6	No	No
Height Above Terrain	6	Yes	No

Table 4. Rotor Separation FDM Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Roll	6	No	Yes
Yaw	6	No	Yes
Pitch	6	No	Yes
Acceleration (3-axis)	6	No	No
Altitude (MSL)	6	No	No
Motor RPM	6	No	No

Table 5. Loss of Control FDM Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Acceleration (3-axis)	64 or higher	No	Yes
Altitude (MSL)	6	No	No
Height Above Terrain	6	Yes	No
Roll	6	No	No
Yaw	6	No	No
Pitch	6	No	No
Motor RPM	6	No	No

Table 6. Hard Landing FDM Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Acceleration (3-axis)	64 or higher	No	Yes
Altitude (MSL)	6	No	No
Height Above Terrain	6	Yes	No
Roll	6	No	No
Yaw	6	No	No
Pitch	6	No	No
Motor RPM	6	No	No

Table 7. Collision on Ground FDM Data Parameters.

Type of Parameter	Minimum Refresh Rate (Hz)	Derivable?	Required?
Time	6	No	Yes
Latitude Position	6	No	Yes
Longitude Position	6	No	Yes
Acceleration (3-axis)	64 or higher	No	Yes
Altitude (MSL)	6	No	No
Height Above Terrain	6	Yes	No
Roll	6	No	No
Yaw	6	No	No
Pitch	6	No	No
Motor RPM	6	No	No
ADSB Traffic Data	1	No	No

Table 8. Collision in Air FDM Data Parameters.

Drawing from the above categories, a basic UFDM parameter set can be defined. At the data-poor end, time and location (latitude and longitude) would be the minimum required data needed to be included in a nationwide UFDM program. With only these parameters, many of the above accident categories – and, presumably, most categories – would not be able to be very robustly evaluated. The lack of inclusion of additional parameters would not allow for a complete risk mitigation system as seen in other CAST-like endeavors, but they would allow for some basic analysis and as such should be accepted into any UFDM database. So, although more data would be needed for a more robust and complete UFDM solution, at least the above two parameters would be required at a minimum.

To properly create analytical tools useful to safety analysts, Table 9 depicts the necessary data that should be included in any UAS flight recording system. This will allow for proper risk assessment for most accident categories, and will also help guide any future CAST-like group as they examine data and create mitigations. The refresh rate includes the minimum recommended for meaningful data analysis, with acceleration having a minimum of 64 Hertz. Regarding acceleration, the rationale behind requiring such a high refresh rate is for the allowance of trying to capture instantaneous G-loading, which can occur will with a one-second time interval. Some parameters that can be derived are not included in Table 9.

Type of Parameter	Minimum Refresh Rate (Hz)
Time	6
Latitude Position	6
Longitude Position	6
Altitude (MSL)	6
Roll	6
Yaw	6
Pitch	6
Acceleration (3-axis)	64
Motor RPM	6
Battery Voltage	1

Table 9. Minimum Required Data for UFDM Recorders.

XII. CONCLUSION

It is clear that the current state-of-the-art UFDM primarily involves the ability to collect flight data. Long term data storage and analysis, as well as feedback to the operators, are not really existent outside of individual organizations. It should be noted however that the unmanned industry is ahead of where the general aviation industry was in its onset into FDM, primarily in terms of flight data availability. Within general aviation, flight data was not readily available to operators until the late 2000s. That is clearly not the issue with unmanned platforms.

Figure 6 depicts a fully functional overview of a UFDM implementation. At present, only the data logging capabilities are found within the UAS community. From the figure, in order to develop a robust and active CAST/ASIAS-like FDM program, database storage, data analysis and feedback to the operators need to be developed. In addition, the issue of governance will have to be agreed upon by participating operators. This project recommends a basic level of data and data types for UFDM analysis.

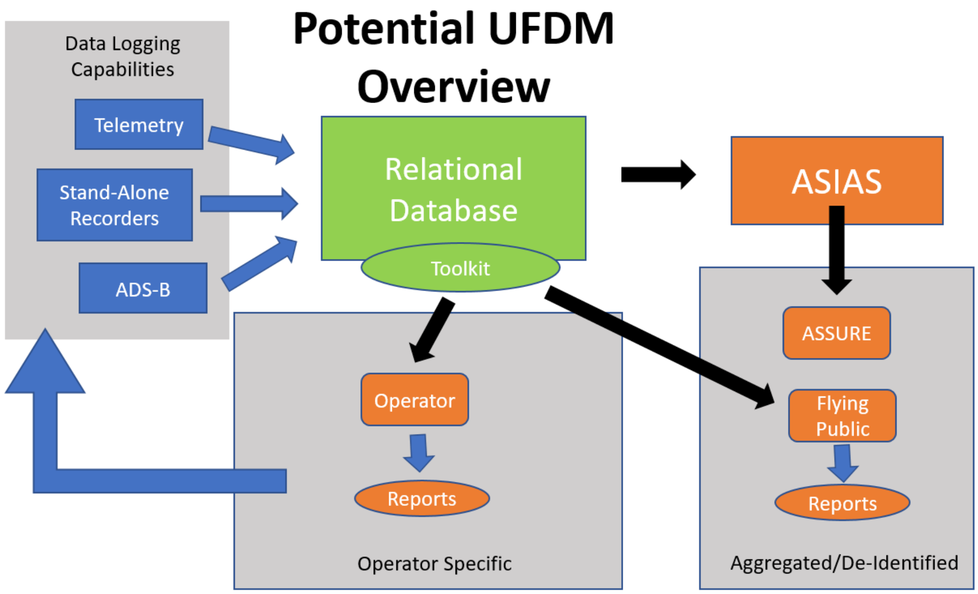


Figure 6. Potential UFDM Overview.

As this ASSURE study progresses, the focus will turn directly onto the flight data. Data availability, standardization and usefulness will be evaluated in terms of usability within a CAST/GAJSC safety paradigm. The good news is data is available and relatively easy to recover. There is however little uniformity to data formats and types of data collected varies widely. Although this will be a challenge to overcome in creating a nationwide UFDM program, the basic building blocks are in place. The remaining tasks include building a data repository and establishing governance.

References

Aerobytes. (2019). FDM/FOQA solutions. Retrieved from https://www.aerobytes.co.uk

Ahire, S. L., Golhar, D. Y., & Waller, M. A. (1996). Development and validation of TQM implementation constructs. Decision Sciences, 27(1), 23-56. doi:10.1111/j.1540-5915.1996.tb00842.x

R. E. Barlow. (1984). Mathematical theory of reliability: A historical perspective. IEEE Transactions on Reliability, R-33(1), 16-20. doi:10.1109/TR.1984.6448269

Box, G. (1993). Quality improvement: The new industrial revolution. International Statistical Review / Revue Internationale De Statistique, 61(1), 3-19. doi:10.2307/1403590

Callantine, T. J. (2001). *Analysis of flight operational quality assurance data using model-based activity tracking* (No. 2001-01-2640). SAE Technical Paper.

Carstens, Deborah. (2018). Project 22: Airport Safety Database and Analysis. https://www.pegasas.aero/projects/airport-safety-database-and-analysis

Cavka, 1Ivana and Cokorilo Olja. 2012. Cost-Benift Assessment of Aircraft Safety. International Journal for Tra_c and Transport Engineering, 2012, 2(4): 359 – 371

Clachar, Sophine A. (2015). Identifying and Analyzing Atypical Flights by Using Supervised and Unsupervised Approaches. Transportation Research Record: Journal of the Transportation Research Board, No. 2471, 10–18.

Chang, Ray. (2014). Performance Diagnosis for Turbojet Engines Based on Flight Data. Journal of Aerospace Engineering, Volume: 27 (1). 9-15.

A. Coppola. (1984). Reliability engineering of electronic equipment a historical perspective. IEEE Transactions on Reliability, R-33(1), 29-35. doi:10.1109/TR.1984.6448271

CsvView/DatCon. (n.d.). Retrieved from https://datfile.net/

CsvView Manual. (n.d.). Retrieved from https://datfile.net/Doc/CsvViewManual.pdf

Draper, N. R., & Smith, H. (1998). *Applied regression analysis* (3rd ed.). Hoboken,NJ: John Wiley & Sons. FAA (2004) AC 120-82 - Flight Operational Quality Assurance Document Informationhttps://www.faa.gov/regulations_policies/advisory_circulars/ index.cfm/go/document.information/documentID/23227

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. (2019) Become A Drone Pilot. https://www.faa.gov/uas/commercial_operators/become_a_drone_pilot/

FAA (2018). Flying Drones Near Airports (Controlled Airspace) – Part 107 https://www.faa.gov/uas/commercial_operators/part_107/

FAA (May, 2017). Safety Risk Management Policy. Retrieved from https://www.faa.gov/ documentLibrary/media/Order/FAA_Order_8040.4B.pdf

FAA (2019) Part 107 Waivers https://www.faa.gov/uas/commercial_operators/part_107_waivers/

Faraway, J. J. (2016). *Linear models with R* (2nd ed.). Boca Raton,FL: Chapman and Hall/CRC.

Fernandez, R. V. (September, 2002). An Analysis of the Potential Benefits to Airlines of Flight Data Monitoring Programmes. Unpublished Master's thesis, Cranfield University, Cranfield, Bedfordshire, U.K.

Fisher, R. A. (1935). The design of experiments. London, UK: Oliver And Boyd.

General Electric. (2019). eFOQA Solutions. Retrieved from https://www.geaviation.com /digital/flight-risk-management-commercial-aviation

GPS.gov. (2019). GPS Accuracy. Retrieved from https://www.gps.gov/systems/gps/performance/accuracy/

Hackman, J. R., & Wageman, R. (1995). Total quality management: Empirical, conceptual, and practical issues. Administrative Science Quarterly, 40(2), 309-342. doi:10.2307/2393640

Helicopter Association International. (2014). Collect, Aggregate, and Disseminate Rotorcraft Flight Data Monitoring Data to Provide Data Driven Safety Analysis. https://trid.trb.org/View/1367521

Holtom, M. (2000). Properly managed FOQA programme represents an important safety tool for airlines. *ICAO Journal*, *55*(1), 7-11.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Hoboken, NJ: John Wiley & Sons.

International Civil Aviation Administration (ICAO). (2009). Safety Management Manual (SMM) – Second Edition. (ICAO Doc 9859 AN/474). Montreal, QC, Canada: Author.

Ishikawa, K. (1985). What is total quality control? The Japanese way Prentice Hall.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (1st ed.). New York, NY: Springer.

Juran, J. M., Gryna, F. M., & Bingham, R. S. (1974). Quality control handbook (3rd ed.). New York,NY: McGraw-Hill.

Kackar, R. N. (1985). Off-line quality control, parameter design, and the Taguchi method. Journal of Quality Technology, 17(4), 176-188. doi:10.1080/00224065.1985.11978964

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding groups in data: An introduction to cluster analysis* (1st ed.). Hoboken, NJ: John Wiley & Sons.Klefsjö, B., Edgeman, R. L., &

Wiklund, H. (2001). Six sigma seen as a methodology for total quality management. Measuring Business Excellence, 5(1), 31-35. doi:10.1108/13683040110385809

Kline, R. B. (2015). *Principles and practice of structural equation modeling* (4th ed.). New York, NY: Guilford publications. Kontogiannis, T., Leva, M. C., & Balfe, N. (2017). Total safety management: Principles, processes and methods. Safety Science, 100(B), 128-142. doi:https://doi.org/10.1016/j.ssci.2016.09.015

Lacagnina, M. (2007). C-FOQA takes root. *AeroSafety World*, *2*, 11-15.

Landsberg, Bruce. (2011). Safety Pilot: How Much Data Should We Collect? https://www.aopa.org/news-and-media/all-news/2011/april/01/safety-pilot

Li, Lishuai, Hansman, R John, Palacios, Rafael and Welsch, Roy (2016). Abnormally detection via a Gaussian Mixture Model for flight operation and safety monitoring. Transportation Research Part C: Emerging Technologies.. Volume: 64, 45-57

Larder, A. D. (1999, May). Helicopter HUM/FDR: benefits and developments. In *ANNUAL FORUM PROCEEDINGS-AMERICAN HELICOPTER SOCIETY* (Vol. 55, pp. 1839-1846). American Helicopter Society.

Lawson, J. (2014). Design and analysis of experiments with R (1st ed.). Boca Raton,FL: Chapman and Hall/CRC.

Lowe, S. E., Pfleiderer, E. E., & Chidester, T. R. (2012). *Perceptions and efficacy of flight operational quality assurance (FOQA) programs among small-scale operators* (No. DOT-FAA-AM-12/1). FEDERAL AVIATION ADMINISTRATION OKLAHOMA CITY OK CIVIL AEROSPACE MEDICAL INST.

Marias, Karen. 2018. Project 5: Safety Analysis for general Aviation. https://www.pegasas.aero/projects/project-5-safety-analysis-for-general-aviation

Mavris, D., DeLaurentis, D., Bandte, O., & Hale, M. (1998). A stochastic approach to multi-disciplinary aircraft analysis and design. *36th AIAA Aerospace Sciences Meeting and Exhibit*. Retrieved from https://arc.aiaa.org/doi/abs/10.2514/6.1998-912.

MIT Lincoln Laboratory Ground-based system helps UAVs avoid collision https://www.ll.mit.edu/news/ground-based-system-helps-uavs-avoid-collisions

Mitchell, K., Sholy, B., & Stolzer, A. J. (2007). General aviation aircraft flight operations quality assurance: overcoming the obstacles. *IEEE Aerospace and Electronic Systems Magazine*, *22*(6), 9-15.

MITRE. (August, 2013). Government and Industry Collaborate to Improve Safety through Data Sharing. Retrieved from https://www.mitre.org/publications/project-stories/government-and-industry-collaborate-to-improve-safety-through-data-sharing

Montgomery, D. C. (2012). Statistical quality control (7th ed.). Hoboken, NJ: John Wiley & Sons.

Montgomery, D. C. (2017). Design and analysis of experiments(9th ed.). Hoboken,NJ: John Wiley & Sons.

Myers, R. H., Montgomery , D. C., & Anderson-Cook, C. M. (2016). Response surface methodology: Process and product optimization using designed experiments (4th ed.). Hoboken,NJ: John Wiley & Sons.

NGAID. (2019). National General Aviation Flight Information Database. Retrieved from https://ngafid.org/

Oakland, J. S. (2007). Statistical process control (6th ed.). London, UK: Routledge.

Julian Oehling, David J. Barry. (2019). Using Machine Learning Methods in Airline Flight Data Monitoring to Generate New Operational Safety Knowledge From Existing Data. Safety Science 114 (2019) 89–104.

Parker, P. A., & Finley, T. D. (2007). Advancements in aircraft model force and attitude instrumentation by integrating statistical methods. Journal of Aircraft, 44(2), 436-443.Powell, T. C. (1995). Total quality management as competitive advantage: A review and empirical study. Strategic Management Journal, 16(1), 15-37. doi:10.1002/smj.4250160105

Saleh, J. H., & Marais, K. (2006). Highlights from the early (and pre-) history of reliability engineering. Reliability Engineering & System Safety, 91(2), 249-256.

P. A. Samara, G. N. Fouskitakis, J. S. Sakellariou, & S. D. Fassois. (2008). A statistical method for the detection of sensor abrupt faults in aircraft control systems. *IEEE Transactions on Control Systems Technology, 16*(4), 789-798.

Skylogic Research. (2017). 2017 drone market sector prospectus: Opportunities and challenges in key market segments. *Author.* Retrieved from http://droneanalyst.com/research/ research-studies/2017-drone-marketsector-report

Stuart, M., Mullins, E., & Drew, E. (1996). Statistical quality control and improvement. European Journal of Operational Research, 88(2), 203-214. doi:https://doi.org/10.1016/0377-2217(95)00069-0

Taguchi, G. (1985). Quality engineering in Japan. Communications in Statistics - Theory and Methods, 14(11), 2785-2801. doi:10.1080/03610928508829076

Thiels, C. A., Aho, J. M., Zietlow, S. P., & Jenkins, D. H. (2015). Use of unmanned aerial vehicles for medical product transport. *Air Medical Journal*, *34*(2), 104-108.

Tsutsui, W. M. (1996). W. Edwards Deming and the origins of quality control in japan. Journal of Japanese Studies, 22(2), 295-325. doi:10.2307/132975.\

UAST. (2019). Preventing bullying. Retrieved from https://www.unmannedaircraftsafetyteam.org/is-an-industry-government-partnership-committed-to-ensuring-the-safe-operations-of-unmanned-aircraft-systems/

Volpe National Transportation Center. (2019) System Enables Drone Operators to Detect and Avoid Other Aircraft https://www.volpe.dot.gov/air-traffic-systems-operations/ air-traffic-management-systems/ground-based-sense-and-avoid

Walsh, E. B., & Feerrar, W. N. (March 20, 2018). Unmanned Aircraft Safety Team Data Gap Analysis Progress Report. Washington, DC: MITRE.

Wikipedia (2019) List of UAV Related Incidents. https://en.wikipedia.org/wiki/List_of_UAV-related_incidents

Yuneec. (n.d.). Yuneec H520 Owner's Guide. Retrieved from http://commercial.yuneec.com/files/downloads/H520/Yuneec%20H520%20Manual+Datapilot%20Operations%20Guide_EN.pdf

Zhang, Y., & Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, *32*(2), 229-252.

Zio, E. (2009). Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety, 94(2), 125-141. doi:https://doi.org/10.1016/j.ress.2008.06.002

XII. APPENDIX A

Table 1

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,			Yes

	FollowFunctionClose, IOCOpen, IOCClose, DropCalibration, PackMode, UnPackMode, EnterManaualMode, StopDeform), DownDeform, UpDeform, ForceLanding, ForceLanding2, OTHER			
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 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	10	No	
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,	200	Yes

		pitch, and yaw values coming from Flight Controller.			
	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 2

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),	10	No

	GPS_Atti_Limited(24), Follow_Me(25),OTHER(100);		
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), Joystick(200), Atti_Limited(300), GPS_Atti_Limited(400), Follow_Me(500),OTHER(600);	10	Yes
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 3

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	*	supposed	
yaw	Gier	*	supposed	
pitch	Nick	*	supposed	
motor_status	Motor Status, bitwise. Motor numbers according to the picture in the GUI		supposed	
imu_status	IMU Status (intertial meassurement unit)	bits	supposed	
gps_status	GPS unit status	bits	supposed	
cgps_used	C-GPS (unknown meaning)	?		
press_compass_status	Sensor Status (Barometer/Magnetometer)	bits	supposed	
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	suppos ed	
	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).

XIII. APPENDIX B

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

XV. Appendix C

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			

86	**Field:**	selected_decision_height	**Type:**	smallint(5)	**Not Required**
	Units:	feet	**Range:**	-99,999 - 99,999	
	Description:	Selected decision height in autopilot			
	Example:	200			

87	**Field:**	efis_display_format*	**Type:**	tinyint(3)	**Not Required**
	Units:	NA	**Range:**	0 - 256	
	Description:	Selected EFIS format/mode. *Note: separate table will be available to describe various modes. efis_display_format will be foreign keyed into the other table.			
	Example:	3			

88	**Field:**	mfd_display_format*	**Type:**	tinyint(3)	**Not Required**
	Units:	NA	**Range:**	0 - 256	
	Description:	Selected MFD format/mode. *Note: separate table will be available to describe various modes. mfd_display_format will be foreign keyed into the other table.			
	Example:	3			

89	**Field:**	thrust_command	**Type:**	varchar(8)	**Not Required**
	Units:	percent	**Range:**	Undefined	
	Description:	Description of commanded thrust for auto-throttle equipped aircraft, will be aircraft specific.			
	Example:	92.6			

90	**Field:**	thrust_target	**Type:**	varchar(8)	**Not Required**
	Units:	percent	**Range:**	Undefined	
	Description:	Description of thrust target set, will be aircraft specific.			
	Example:	92.6			

91	**Field:**	fuel_quantity_total	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Total fuel quantity as recorded (not dervived).			
	Example:	288.761			

92	**Field:**	fuel_quantity_left_main	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity left main tank as recorded.			
	Example:	145.412			

93	**Field:**	fuel_quantity_right_main	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity right main tank as recorded.			
	Example:	145.412			

94	**Field:**	fuel_quantity_aux_1	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity auxiliary tank # 1 as recorded.			
	Example:	91.765			

95	**Field:**	fuel_quantity_aux_2	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity auxiliary tank # 2 as recorded.			
	Example:	91.765			

96	**Field:**	fuel_quantity_aux_3	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity auxiliary tank # 3 as recorded.			
	Example:	91.765			

97	**Field:**	fuel_quantity_cg_trim_tank	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs.	**Range:**	0 - 99,999.999	
	Description:	Fuel quantity Center of Gravity (CG) trim tank as recorded.			
	Example:	91.765			

98	**Field:**	eng_1_fuel_flow	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs. per hour	**Range:**	0 - 99,999.999	
	Description:	Fuel flow engine #1			
	Example:	128.311			

99	**Field:**	eng_2_fuel_flow	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs. per hour	**Range:**	0 - 99,999.999	
	Description:	Fuel flow engine #2			
	Example:	128.311			

100	**Field:**	eng_3_fuel_flow	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs. per hour	**Range:**	0 - 99,999.999	
	Description:	Fuel flow engine #3			
	Example:	128.311			

101	**Field:**	eng_4_fuel_flow	**Type:**	float(8,3)	**Not Required**
	Units:	Lbs. per hour	**Range:**	0 - 99,999.999	
	Description:	Fuel flow engine #4			
	Example:	128.311			

102	**Field:**	primary_nav_system_reference*	**Type:**	tinyint(3)	**Not Required**
	Units:	NA	**Range:**	0 - 256	
	Description:	Primary navigation used for system reference. *Note: separate table will be available to describe various modes. primary_nav_system_reference format will be foreign keyed into the other table.			
	Example:	3			

103	**Field:**	icing	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Ice detection system status.			
	Example:	no			

104	**Field:**	eng_1_vibration_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #1 vibration indication.			
	Example:	no			

105	**Field:**	eng_2_vibration_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #2 vibration indication.			
	Example:	no			

106	**Field:**	eng_3_vibration_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #3 vibration indication.			
	Example:	no			

107	**Field:**	eng_4_vibration_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #4 vibration indication.			
	Example:	no			

108	**Field:**	eng_1_overtemp_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #1 overtemp warning indication.			
	Example:	no			

109	**Field:**	eng_2_overtemp_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #2 overtemp warning indication.			
	Example:	no			

110	**Field:**	eng_3_overtemp_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #3 overtemp warning indication.			
	Example:	no			

111	**Field:**	eng_4_overtemp_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #4 overtemp warning indication.			
	Example:	no			

112	**Field:**	eng_1_oil_press	**Type:**	float(5,2)	**Not Required**
	Units:	psi	**Range:**	-999.99 - 999.99	
	Description:	Oil pressure engine #1			
	Example:	87.22			

113	**Field:**	eng_2_oil_press	**Type:**	float(5,2)	**Not Required**
	Units:	psi	**Range:**	-999.99 - 999.99	
	Description:	Oil pressure engine #2			
	Example:	87.22			

114	**Field:**	eng_3_oil_press	**Type:**	float(5,2)	**Not Required**
	Units:	psi	**Range:**	-999.99 - 999.99	
	Description:	Oil pressure engine #3			
	Example:	87.22			

115	**Field:**	eng_4_oil_press	**Type:**	float(5,2)	**Not Required**
	Units:	psi	**Range:**	-999.99 - 999.99	
	Description:	Oil pressure engine #4			
	Example:	87.22			

116	**Field:**	eng_1_oil_press_low_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #1 low oil pressure warning.			
	Example:	no			

117	**Field:**	eng_2_oil_press_low_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #2 low oil pressure warning.			
	Example:	no			

118	**Field:**	eng_3_oil_press_low_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #3 low oil pressure warning.			
	Example:	no			

119	**Field:**	eng_4_oil_press_low_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #4 low oil pressure warning.			
	Example:	no			

120	**Field:**	eng_1_oil_temp	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Engine #1 oil temperature			
	Example:	107.218			

121	**Field:**	eng_2_oil_temp	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Engine #2 oil temperature			
	Example:	107.218			

122	**Field:**	eng_3_oil_temp	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Engine #3 oil temperature			
	Example:	107.218			

123	**Field:**	eng_4_oil_temp	**Type:**	float(6,3)	**Not Required**
	Units:	Degrees	**Range:**	-999.999 - 999.999	
	Description:	Engine #4 oil temperature			
	Example:	107.218			

124	**Field:**	eng_1_overspeed_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #1 overspeed warning.			
	Example:	no			

125	**Field:**	eng_2_overspeed_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #2 overspeed warning.			
	Example:	no			

126	**Field:**	eng_3_overspeed_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #3 overspeed warning.			
	Example:	no			

127	**Field:**	eng_4_overspeed_warning	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Engine #4 overspeed warning.			
	Example:	no			

128	**Field:**	eng_1_cht_1	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #1			
	Example:	204.11			

129	**Field:**	eng_1_cht_2	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #2			
	Example:	204.11			

130	**Field:**	eng_1_cht_3	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #3			
	Example:	204.11			

131	**Field:**	eng_1_cht_4	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #4			
	Example:	204.11			

132	**Field:**	eng_1_cht_5	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #5			
	Example:	204.11			

133	**Field:**	eng_1_cht_6	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #1 Cylinder Head Temperature (CHT) for cylinder #6			
	Example:	204.11			

134	**Field:**	eng_2_cht_1	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #1			
	Example:	204.11			

135	**Field:**	eng_2_cht_2	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #2			
	Example:	204.11			

136	**Field:**	eng_2_cht_3	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #3			
	Example:	204.11			

137	**Field:**	eng_2_cht_4	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #4			
	Example:	204.11			

138	**Field:**	eng_2_cht_5	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #5			
	Example:	204.11			

139	**Field:**	eng_2_cht_6	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #2 Cylinder Head Temperature (CHT) for cylinder #6			
	Example:	204.11			

140	**Field:**	eng_3_cht_1	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #1			
	Example:	204.11			

141	**Field:**	eng_3_cht_2	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #2			
	Example:	204.11			

142	**Field:**	eng_3_cht_3	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #3			
	Example:	204.11			

143	**Field:**	eng_3_cht_4	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #4			
	Example:	204.11			

144	**Field:**	eng_3_cht_5	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #5			
	Example:	204.11			

145	**Field:**	eng_3_cht_6	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #3 Cylinder Head Temperature (CHT) for cylinder #6			
	Example:	204.11			

146	**Field:**	eng_4_cht_1	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #1			
	Example:	204.11			

147	**Field:**	eng_4_cht_2	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #2			
	Example:	204.11			

148	**Field:**	eng_4_cht_3	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #3			
	Example:	204.11			

149	**Field:**	eng_4_cht_4	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #4			
	Example:	204.11			

150	**Field:**	eng_4_cht_5	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #5			
	Example:	204.11			

151	**Field:**	eng_4_cht_6	**Type:**	float(5,2)	**Not Required**
	Units:	Degrees F	**Range:**	-999.99 - 999.99	
	Description:	Engine #4 Cylinder Head Temperature (CHT) for cylinder #6			
	Example:	204.11			

152	**Field:**	eng_1_egt_1	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #1			
	Example:	200.31			

153	**Field:**	eng_1_egt_2	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #2			
	Example:	200.31			

154	**Field:**	eng_1_egt_3	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #3			
	Example:	200.31			

155	**Field:**	eng_1_egt_4	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #4			
	Example:	200.31			

156	**Field:**	eng_1_egt_5	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #5			
	Example:	200.31			

157	**Field:**	eng_1_egt_6	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #1 Exhaust Gas Temperature (EGT) cylinder #6			
	Example:	200.31			

158	**Field:**	eng_2_egt_1	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #1			
	Example:	200.31			

159	**Field:**	eng_2_egt_2	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #2			
	Example:	200.31			

160	**Field:**	eng_2_egt_3	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #3			
	Example:	200.31			

161	**Field:**	eng_2_egt_4	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #4			
	Example:	200.31			

162	**Field:**	eng_2_egt_5	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #5			
	Example:	200.31			

163	**Field:**	eng_2_egt_6	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #2 Exhaust Gas Temperature (EGT) cylinder #6			
	Example:	200.31			

164	**Field:**	eng_3_egt_1	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #1			
	Example:	200.31			

165	**Field:**	eng_3_egt_2	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #2			
	Example:	200.31			

166	**Field:**	eng_3_egt_3	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #3			
	Example:	200.31			

167	**Field:**	eng_3_egt_4	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #4			
	Example:	200.31			

168	**Field:**	eng_3_egt_5	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #5			
	Example:	200.31			

169	**Field:**	eng_3_egt_6	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #3 Exhaust Gas Temperature (EGT) cylinder #6			
	Example:	200.31			

170	**Field:**	eng_4_egt_1	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #1			
	Example:	200.31			

171	**Field:**	eng_4_egt_2	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #2			
	Example:	200.31			

172	**Field:**	eng_4_egt_3	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #3			
	Example:	200.31			

173	**Field:**	eng_4_egt_4	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #4			
	Example:	200.31			

174	**Field:**	eng_4_egt_5	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #5			
	Example:	200.31			

175	**Field:**	eng_4_egt_6	**Type:**	float(6,2)	**Not Required**
	Units:	Degrees F	**Range:**	-9999.99 - 9999.99	
	Description:	Engine #4 Exhaust Gas Temperature (EGT) cylinder #6			
	Example:	200.31			

176	**Field:**	yaw_trim_surface_position	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Rudder trim surface position, negative denotes left, positive denotes right			
	Example:	-4.1092			

177	**Field:**	roll_trim_surface_position	**Type:**	float(6,4)	**Not Required**
	Units:	Degrees	**Range:**	-99.9999 - 99.9999	
	Description:	Roll trim surface position, negative denotes left, positive denotes right			
	Example:	6.7884			
178	**Field:**	brake_pressure_system_1	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Brake pressure system #1			
	Example:	74.2187			

179	**Field:**	brake_pressure_system_2	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Brake pressure system #2			
	Example:	74.2187			

180	**Field:**	brake_pressure_system_3	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Brake pressure system #3			
	Example:	74.2187			

181	**Field:**	brake_pedal_application_left	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Left side brake pedal application.			
	Example:	22.1237			

182	**Field:**	brake_pedal_application_right	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Right side brake pedal application.			
	Example:	22.1237			

183	**Field:**	sideslip_angle	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Angle of sideslip as recorded (not derived), negative denotes left, positive denotes right.			
	Example:	-2.1659			

184	**Field:**	eng_1_bleed_valve_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Amount of opening in bleed valve, engine 1. 0 denotes fully closed.			
	Example:	13.8772			

185	**Field:**	eng_2_bleed_valve_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Amount of opening in bleed valve, engine 2. 0 denotes fully closed.			
	Example:	13.8772			

186	**Field:**	eng_3_bleed_valve_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Amount of opening in bleed valve, engine 3. 0 denotes fully closed.			
	Example:	13.8772			

187	**Field:**	eng_4_bleed_valve_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Amount of opening in bleed valve, engine 4. 0 denotes fully closed.			
	Example:	13.8772			

188	**Field:**	deicing_system_selection*	**Type:**	tinyint(2)	**Not Required**
	Units:	NA	**Range:**	0 - 99	
	Description:	Selection of onboard de-icing. *Note: separate table will be available to describe various modes. deicing_system_selection format will be foreign keyed into the other table.			
	Example:	3			

189	**Field:**	computed_cg	**Type:**	float(9,4)	**Not Required**
	Units:	Inches aft of datum	**Range:**	-99,999.9999 - 99,999.9999	
	Description:	Computed CG as recorded (not derived).			
	Example:	64.9992			

190	**Field:**	ac_bus_1_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	AC Bus #1 status.			
	Example:	powered			

191	**Field:**	ac_bus_2_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	AC Bus #2 status.			
	Example:	powered			

192	**Field:**	ac_bus_3_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	AC Bus #3 status.			
	Example:	powered			

193	**Field:**	ac_bus_4_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	AC Bus #4 status.			
	Example:	powered			
194	**Field:**	dc_bus_1_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	DC Bus #1 status.			
	Example:	powered			

195	**Field:**	dc_bus_2_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	DC Bus #2 status.			
	Example:	powered			

196	**Field:**	dc_bus_3_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	DC Bus #3 status.			
	Example:	powered			

197	**Field:**	dc_bus_4_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"powered", "not powered"	
	Description:	DC Bus #4 status.			
	Example:	powered			

198	**Field:**	system_1_volts	**Type:**	float(5,2)	**Not Required**
	Units:	Volts	**Range:**	0 - 999.99	
	Description:	Electrical system #1 voltage.			
	Example:	27.21			

199	**Field:**	system_2_volts	**Type:**	float(5,2)	**Not Required**
	Units:	Volts	**Range:**	0 - 999.99	
	Description:	Electrical system #2 voltage.			
	Example:	27.21			

200	**Field:**	system_1_amps	**Type:**	float(5,2)	**Not Required**
	Units:	Amps	**Range:**	0 - 999.99	
	Description:	Electrical system #1 amperage.			
	Example:	27.21			

201	**Field:**	system_2_amps	**Type:**	float(5,2)	**Not Required**
	Units:	Amps	**Range:**	0 - 999.99	
	Description:	Electrical system 2 amperage.			
	Example:	27.21			

202	**Field:**	apu_bleed_valve_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Amount of opening in APU bleed valve. 0 denotes fully closed.			
	Example:	13.8772			

203	**Field:**	hydraulic_1_pressure	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Hydraulic pressure system #1.			
	Example:	82.1117			

204	**Field:**	hydraulic_2_pressure	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Hydraulic pressure system #2.			
	Example:	82.1117			

205	**Field:**	hydraulic_3_pressure	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Hydraulic pressure system #3.			
	Example:	82.1117			

206	**Field:**	hydraulic_4_pressure	**Type:**	float(8,4)	**Not Required**
	Units:	psi	**Range:**	0 - 9999.9999	
	Description:	Hydraulic pressure system #4.			
	Example:	82.1117			

207	**Field:**	loss_cabin_pressure	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Loss of cabin pressure indication.			
	Example:	no			

208	**Field:**	fms_failure	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	FMS failure detected.			
	Example:	no			

209	**Field:**	hud_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Heads-up display (HUD) status.			
	Example:	no			

210	**Field:**	synthetic_vision_display_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Synthetic Vision System (SVS) status.			
	Example:	no			

211	**Field:**	paravisual_display_status	**Type:**	enum	**Not Required**
	Units:	NA	**Range:**	"no", "yes"	
	Description:	Paravisual display status.			
	Example:	no			

212	**Field:**	pitch_trim_control_selection	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Pitch trim selection from flight deck, negative denotes down trim, positive denotes up trim.			
	Example:	-3.1117			

213	**Field:**	roll_trim_control_selection	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Roll trim selection from flight deck, negative denotes left trim, positive denotes right trim.			
	Example:	-1.2761			

214	**Field:**	yaw_trim_control_selection	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Yaw trim selection from flight deck, negative denotes left trim, positive denotes right trim.			
	Example:	6.1276			

215	**Field:**	trailing_edge_flap_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Trailing edge device actual control surface position.			
	Example:	6.1276			

216	**Field:**	leading_edge_flap_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Leading edge device actual control surface position.			
	Example:	6.1276			

217	**Field:**	spoiler_position	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	0 - 9999.9999	
	Description:	Spoiler actual control surface position.			
	Example:	15.0081			
218	**Field:**	spoiler_selection	**Type:**	float(8,4)	**Not Required**
	Units:	Degrees	**Range:**	-9999.9999 - 9999.9999	
	Description:	Spolier selection from flight deck.			
	Example:	14.9823			

XV. Appendix D

Minimum Required Data for UFDM Recorders.

Type of Parameter	Minimum Refresh Rate (Hz)
Time	6
Latitude Position	6
Longitude Position	6
Altitude (MSL)	6
Roll	6
Yaw	6
Pitch	6
Acceleration (3-axis)	64
Motor RPM	6
Battery Voltage	1