Automated Reporting for Flight Tests: Enhancing Efficiency with AI and Digital Twin Technology

Martin von Depka Prondzinski, Emy Arts, Christina Pätzold, Hendrik Meyer
German Aerospace Center (DLR)
Braunschweig, 38108, Germany
martin.vondepkaprondzinski@dlr.de

Modern flight tests generate vast amounts of data at a pace that surpasses human interpretability. As a consequence, the need for structured reports addressing various stakeholders, ranging from sustainability and safety to management is rapidly increasing.
An essential part of every flight test is the creation of a flight test report. Although all necessary information is available digitally, flight test engineers must manually compile and verify it, a process that can take several days or even weeks depending on the complexity. In an effort to reduce unnecessary manual labor, this work investigates the automated creation of parts of such reports. Initial executive reports are created using sensor data, flight documentation, and other external data sources such as ADS-B traffic and METAR, which are analyzed with traditional as well as AI-powered tools. The contents comprise information about the time and date at which the flight took place, the flight trajectory, flown maneuvers, fuel consumption, METAR reports at departure and arrival, specific sensor values, measurement volume verification and the presence of nearby traffic. The report is automatically created in less than a minute and only requires quick human verification, leading to a significant decrease of manual labor required.

1. INTRODUCTION

With the introduction of twinstash [1] at the German Aerospace Center (DLR), the process of providing research data has been significantly optimized—from several days to just a few minutes. However, flight test engineers are still responsible for manually compiling reports, verifying data integrity, and tailoring information to different target audiences. To enhance this workflow, the next development step for twinstash will include an automated reporting system that generates an initial executive summary.
This system will not only incorporate flight trajectories and selected flight parameters but will also enrich the data with external sources such as Automatic Dependent Surveillance Broadcast (ADS-B) traffic information [2] and METeorological Aerodrome Report (METAR) reports [3] as well as an AI-powered analysis tool for flight test cards.
This paper outlines the key features of the reporting tool and presents the structure of the automated executive summary for flight test specialists. In the outlook, further enhancements, including additional automated reports and extended functionalities, will be discussed. By integrating AI and digital twin technology, we aim to significantly accelerate and improve the reporting process, ultimately enabling more efficient and data-driven flight testing

1.1 Problem statement

The DLR operates the largest civilian fleet of its kind in Europe, with 13 specialized research aircraft and helicopters ranging from an Airbus A320 (ATRA) and business jets (e.g. HALO, Falcon 2000LX ISTAR) to helicopters, turboprops and gliders. Each of these aircraft is equipped with extensive measurement and sensor technology. Some aircraft at DLR record up to 3,000 parameters at different intervals, while simultaneously producing additional files such as flight test cards and modification documents. The data obtained is of great scientific interest to a wide range of disciplines from aerodynamics and aircraft system development to atmospheric research and climate science [4]. In the past, the processing and provision of this flight data has been time consuming and delayed. The measurement data was stored in a decentralized manner on a wide variety of systems, often on a project-specific basis and without a consistent link to metadata.
This meant that the data could hardly be found, accessed or reused beyond the original project context. In addition, the conversion and merging of the raw data from many subsystems required numerous manual steps and was correspondingly time consuming. Faulty sensor signals were often only discovered weeks after a flight, which delayed analyses and limited the usability of the data. This lack of efficient data management contradicts modern open science principles. In particular, the FAIR principles were not adhered to by these old processes, as the findability, accessibility, interoperability and reusability of research data could hardly be guaranteed [4].
As the demand for structured reports that address various stakeholders continues to grow the DLR has started to develop a centralized data management system for flight research data.

1.2 DLR Digital Twins / Twinstash

Twinstash (short for "Digital Twins Storage and Application Service Hub") is a platform developed by the DLR for storage, visualization, analysis and automated upload of flight data, which serves as the basis for digital twins of DLR research aircraft [1]. Twinstash has been continuously developed into the current prototype system since 2018 as part of the DigTwin [5] (2018 – 2021) and DigECAT [4] (since 2022) projects. The system supports the upload, download and search of large flight test data sets, including metadata such as take-off/landing time, aircraft identification and trajectory, via both a Python client and a web interface. The web interface offers intuitive navigation through the flight test data, grouping information by project, flight or aircraft, as well as a wide range of visualization tools, such as the display of flight trajectories in 2D and 3D and charts for analyzing recorded sensor signals. Twinstash creates a standardized organizational platform that makes all flight data available within the DLR shortly after landing [1]. This saves large amounts of time as initial measurement data can already be analyzed and discussed during the debriefing.

2. METHODS

As outlined above, the automated reporting process involves multiple data sources and a substantial volume of information. The user provides the system with the name of the flight and, if relevant, specific sensors of interest and the system returns an initial executive report summarizing the flight with information from these different sources. To effectively extract the

relevant data, dedicated methods are implemented to retrieve and process information from diverse origins, including twinstash, flight test cards, and other external services. This section describes the procedures and techniques used to access, extract and process the required data from each of these sources.

2.1 Twinstash

Twinstash provides essential information for generating automated flight reports. This includes measurement data, the configuration of the measurement system, and project-related metadata. Specifically, it offers details such as the project name, flight identifier, and operational metadata. Access to this information is facilitated through the python client provided by twinstash, which enables seamless integration of structured data into the report generation pipeline.

Using the twinstash python client, the relevant flight is identified using the flight name provided by the user. For the creation of the report the flight's metadata and sensor data are retrieved.
From the metadata operational details are extracted, these include: block-on and block-off times, fuel levels, measurement start and stop times, total measurement duration, the date of the flight, as well as comments and Flight Test Instrumentation (FTI) remarks. Using twinstash's special trajectory call, which ensures consistency and synchronization across all recorded trajectory dimensions, the two-dimensional trajectory is visualized using on a flight map using of the Folium[1] python package. The user can define which other sensor data is relevant for the report, to showcase the report the chosen sensors were: true airspeed, barometric altitude and gross weight. The data of these sensors together with their full name and unit are visualized as line charts using Plotly[2].

Additionally, the system verifies whether the measurement system has recorded all the data configured for the flight and whether the volume of collected data appears plausible. This validation requires multiple steps and the cross-checking of different data sources. Firstly, the configuration file, which specifies what sensors were configured for the flight, is retrieved and converted into a structured format that facilitates comparison. Next, twinstash is queried for all sensors recorded for the flight in question. The configured sensors from the configuration file are then compared to the actual sensors recorded in the measurement system, in order to detect any discrepancies between expected and observed sensor data. In addition to checking sensor availability, a plausibility check on the data volume is performed. An estimate of the expected data volume per hour of flight is used to compute the expected total volume for the recorded flight duration. This estimate is then compared to the actual data volume stored in twinstash. When sensor data is uploaded to twinstash for a specific flight, the raw file of this sensor data is uploaded and connected to the flight in binary format as well, the size of this file is then stored in its metadata. To compute the total volume of data for the flight, the file sizes of all sensors are summed up. If the measured data falls within predefined tolerance margins, the data volume is classified as plausible. Otherwise, a warning is issued to indicate a potential anomaly in the data recording process.

[1] https://python-visualization.github.io/folium accessed on 23/05/2025
[2] https://plotly.com/ accessed on 23/05/2025

2.2 Flight Test Cards

The extraction of flight test card data is divided into three main components, each motivated by the specific type of content to be extracted. In this work, we focus on maneuver data, which is typically stored in tabular form within the flight test cards. Consequently, the first step in the extraction pipeline involves detecting the presence and location of tables within the flight test cards. The second step is to recognize the structure of the detected tables, which is essential for interpreting the information they contain. The final step involves understanding and converting the handwritten content within the tables into machine-readable text.

The table detection and structure recognition, are performed using Microsoft's Table Transformer library[3]. This module is integrated into the extraction pipeline as well as into the recognition pipeline. Due to the limited availability of annotated flight test cards, transfer learning is used to enhance performance. Specifically, models that are pretrained on publicly available datasets are fine-tuned using domain-specific data representing typical flight test card layouts and structures. This approach enables the model to generalize better and improves detection accuracy, as the convolutional neural network (CNN) benefits from a richer and more relevant training dataset.

The final component of the pipeline is Optical Character Recognition (OCR), for which we utilize OCR-D, an open-source framework built upon the widely adopted Tesseract OCR engine [6]. OCR-D enhances Tesseract by improving code stability, performance, and maintainability, and by offering seamless integration with other modules within the OCR-D ecosystem. The underlying OCR model is based on a Long Short-Term Memory (LSTM) network, which is well-suited for handling the variability inherent in handwritten text—particularly that of Flight Test Engineers (FTEs). However, in this work, we deliberately avoid training the model on handwritten data due to the limited availability of labeled samples. Instead, our approach relies on a recognition model trained on machine-printed text, which benefits from a much larger and more reliable ground truth. To address the domain-specific handwriting variability, we apply post-correction techniques that leverage aerospace-specific terminology and contextual knowledge. This strategy provides a more flexible and generalizable solution while maintaining high recognition performance.

2.3 External Data

Alongside the information from twinstash and the flight test cards, also external data sources are used to provide information for the reports. Namely, METAR [2] weather reports and OpenSky [6] for ADS-B traffic data.

Using the trajectory data retrieved from twinstash as described previously, the departure and arrival coordinates are obtained. These coordinates are compared to those of airports in the 'ourairports' database[4] to find the nearest airport. The flight departure and arrival time are retrieved from twinstash using the flights metadata, and the METAR message closest in time is provided in the report.

[3] https://github.com/microsoft/table-transformer
[4] https://ourairports.com/world.html accessed on 23/05/2025

The ICAO ID of the aircraft, obtained through its asset metadata and the departure and arrival time from the flights metadata are used to find the flight of interest in the OpenSky historical database. This database provides information regarding for the aircraft's trajectory based on its ADS-B transmissions. The database is also queried for any other aircraft that comes into a range of 37 km laterally or 600 m vertically of the flight for which the report is created. These ranges were chosen based on the ICAO guidelines for separation methods when using ADS-B surveillance [8]. Nearby traffic is found by parsing over the aircraft's trajectory every 15 s and running a second search within the above-named bounds for any other aircraft's presence. If nearby traffic is identified their ICAO IDs are stored and their registration is looked up in the OpenSky aircraft database. For each nearby aircraft their trajectory is analyzed to identify the nearest distance it flew to our flight, the time this distance was smallest, the altitude at the time of smallest distance, when the nearby aircraft was first and last seen in range, as well as the total time it was in range. The trajectory, color coded by altitude, of the user's chosen flight together with those of nearby traffic in the time it was in range is then shown on a map.

2.4 Report Creation

This section provides an overview of how the data obtained from the previously described modules is integrated, how this information is represented in the automated flight report, and how the final PDF report is generated. As previously described, the automated flight report integrates data from multiple sources. The outputs of the three main processing modules— twinstash data retrieval and sensor data processing, flight test card extraction, and processing of information from external sources—serve as inputs for the report generation pipeline. To generate the PDF report, a configuration file is used to define the layout, including positional information for each data element. This file specifies where each piece of content should appear in the report, and the corresponding data is automatically inserted into those positions during the creation process. The configuration file is divided into two main sections. The first defines global elements such as headers or logos, which are repeated on every page. The second contains page-specific content, including trajectory plots, flight metadata, and other flight-specific information, which typically appear only on the first page of the report. For the actual generation of the PDF document, the Python library ReportLab[5] is used, as it provides a flexible and efficient way to programmatically create structured, high-quality PDF reports.

2 RESULTS

Through an automated pipeline, described in its steps in the previous section, a post-flight report can be created in under a minute, this report merely requires the user to name the flight name and specific sensor values they would like to be shown. In the Appendix an example report is provided, of the "F89_20220701_ConMo_6" flight which was flown on July 1st 2022 with the DLR aircraft ISTAR (D-BDLR), a Falcon 2000LX. The first page shows the trajectory as well as its timing, fuel consumption, METAR reports at takeoff and landing, and space for additional comments (in this case the objective of the flight). The second page shows the graphics of specific sensor values

[5] https://www.reportlab.com/ accessed on 23/05/2025

chosen by the user, both the trajectory on the first page and the sensor charts are annotated with events that the flight test engineer marked during the flight by pushing the event button. The third page reports on the maneuvers flown based, which are retrieved from the flight test cards (see section 2.2). The fourth page shows two sections, the first contains the measurement data volume feasibility check (see section 2.1), the green color indicates that the data value is plausible and no sensors were identified as missing. The second section addresses nearby traffic and shows the trajectory of the flight, color coded with its altitude, together with the trajectories of other aircraft in its range (see section 2.3), in this case only one other aircraft was identified at the arrival airport.

3 CONCLUSION

In this study we have shown that the proposed system can significantly reduce the amount of manual labor required to create flight test reports by using an automated pipeline. The system generates a report in less than a minute using a combination of measurement data, the flight test card, and external services and only requires a human to verify the information rather than creating the report from scratch. However, several aspects of the system still require improvement before it can be deployed in a production environment. For instance, the OCR-D component currently lacks robust post-correction mechanisms to automatically resolve recognition errors. Additionally, a verification step involving human oversight is necessary to ensure the correctness of the generated report. This means the process is not yet fully autonomous, as human confirmation is still required. Overall, the proposed solution already covers many essential elements of the initial executive post-flight report, such as timing, fuel consumption, and recorded events. This information and its format, and as a result the example which can be found in the appendix, addresses flight test specialist. To ensure completeness and compliance with operational requirements, this example shall be showcased to a group of flight test specialists to identify missing information or components. Furthermore, the system shall be expanded to create a variety of reports addressing different stakeholders. A manager report, for instance, should include less sensor and maneuver specific information and describe weather information as one would for a layman. On the other hand, other expert-specific reports might need additional functionality, such as analyzing information that spans over the course of several flights. A sustainability report, might address a single aircraft over the course of a year and provide data regarding total fuel consumption and kilometers flown. A research data management report may rather describe how many hours specific sensors were recording for and how much data they generated. A maintenance report on the other hand, should list all shop visits, recorded errors and FTI issues. The reporting system shall thus be adapted to allow the user to select a predefined type of report of or create a personalized report based on a selection of available functionalities.

REFERENCES

[1] S. Haufe, M. Bäßler, C. Pätzold, M. Tchorzewsk, H. Meyer, E. Arts and A. Kamtsiuris, "Digital twins storage and application service hub (TWINSTASH)," in *DLRK*, Dresden, 2022.

[2] RTCA Inc., "Minimum Aviation System Performance Standards for Automatic Dependent Surveillance-Broadcast (ADS-B)," 2002.

[3] Federal Aviation Administration,, "Aeronautical Information Manual Chapter 7. Safety of Flights—Section 1. Meteorology," Washington, 2021.

[4] E. Arts, M. Bäßler, S. Haufe, A. Kamtsiuris, H.Meyer, C. Pätzold, R. Schültzky and M. Tchorzewski, "Digital Twin for Research Aircraft," in *DLRK*, Dresden, 2022.

[5] H. Meyer, J. Zimdahl, A. Kamtsiuris, R. Meissner, F. Raddatz, S. Haufe and M. Bäßler, "Development of a digital twin for aviation research," in *DLRK 2020*, online, 2020.

[6] R. Smith, "An Overview of the Tesseract OCR Engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, Curitiba, 2007.

[7] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic and M. Wilhelm, "Bringing up OpenSky: A Large-scale ADS-B Sensor Network for Research," in *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (IPSN '14)*, Berlin, 2014.

[8] International Civil Aviation Organisation, "Doc 4444 Air Traffic Management Chapter 5. Separation methods and minima," 2016.

BIOGRAPHIES

Martin von Depka Prondzinski is a software developer and researcher at the German Aerospace Center (DLR), where he works as a researcher assistant specializing in software development and data processing. He holds a Bachelor's degree in Computer Science with a focus on Computer Graphics and a minor in Aerospace Engineering from the Technical University of Braunschweig. With prior experience as an intern at Microsoft Germany and a researcher assistant at DLR, he has worked on software prototyping, data analysis, and computational modeling. He has a strong background in software development, algorithm design, and various programming technologies.

Emy Arts is a research associate at the German Aerospace Center (DLR) specializing in AI applications for maintenance. Her 5 years of experience at the DLR, combined with a Bachelor's degree in Informatics at the University of Udine and a Master's degree in Intelligent Adaptive Systems at the University of Hamburg and gave her a niche skillset of a machine learning engineer in the field of aviation maintenance. This skillset is used to tackle data engineering and data science tasks to develop concepts to improve the aviation digitization and aircraft maintenance.

Christina Pätzold is a researcher at the German Aerospace Center (DLR) specialized in flight testing, sensor instrumentation and measurement data handling. She holds a master degree in aerospace engineering from the Technical University Braunschweig. With prior experience as researcher at the Institute of flight guidance at TU Braunschweig und as flight inspector at Flight Calibration Services GmbH, her work focuses on supporting scientists in the preparation, execution and evaluation of flight tests, the development of a FAIR flight test data infrastructure, digital twins, installation of measurement data acquisition systems up to a system for intervention in the flight control system of an aircraft.

Hendrik Meyer is a researcher at the German Aerospace Center (DLR) specializing in maintenance and modification processes in aviation. He holds a degree in Aircraft Engineering from Hamburg University of Applied Sciences and a Master's in Business Administration from AKAD University Stuttgart. With prior experience at Lufthansa Technik, his research focuses on digital twins, predictive maintenance, and Integrated Vehicle Health Management (IVHM). He has contributed to projects such as CleanSky II and DEMETER, aiming to enhance maintenance efficiency in aviation. Meyer is also involved in organizing the CBM Academy, fostering knowledge exchange in condition-based maintenance.

APPENDIX – EXAMPLE REPORT

Flight Report: F89_20220701_ConMo_6

Aircraft: D-BDLR	Project Name: ConMo	twinstash ID: 65723d93c9cbc807e97fc82a

DATE UTC
2022-07-01

TIME
Block OFF UTC
09:10

TAKEOFF UTC
09:15:37

LANDING UTC
10:52:43

Block ON UTC
10:55

Flight Time HH:MM:SS
01:45:00

FUEL
Block OFF
7900 lbs

Block ON
4700 lbs

Fuel Used
3200 lbs

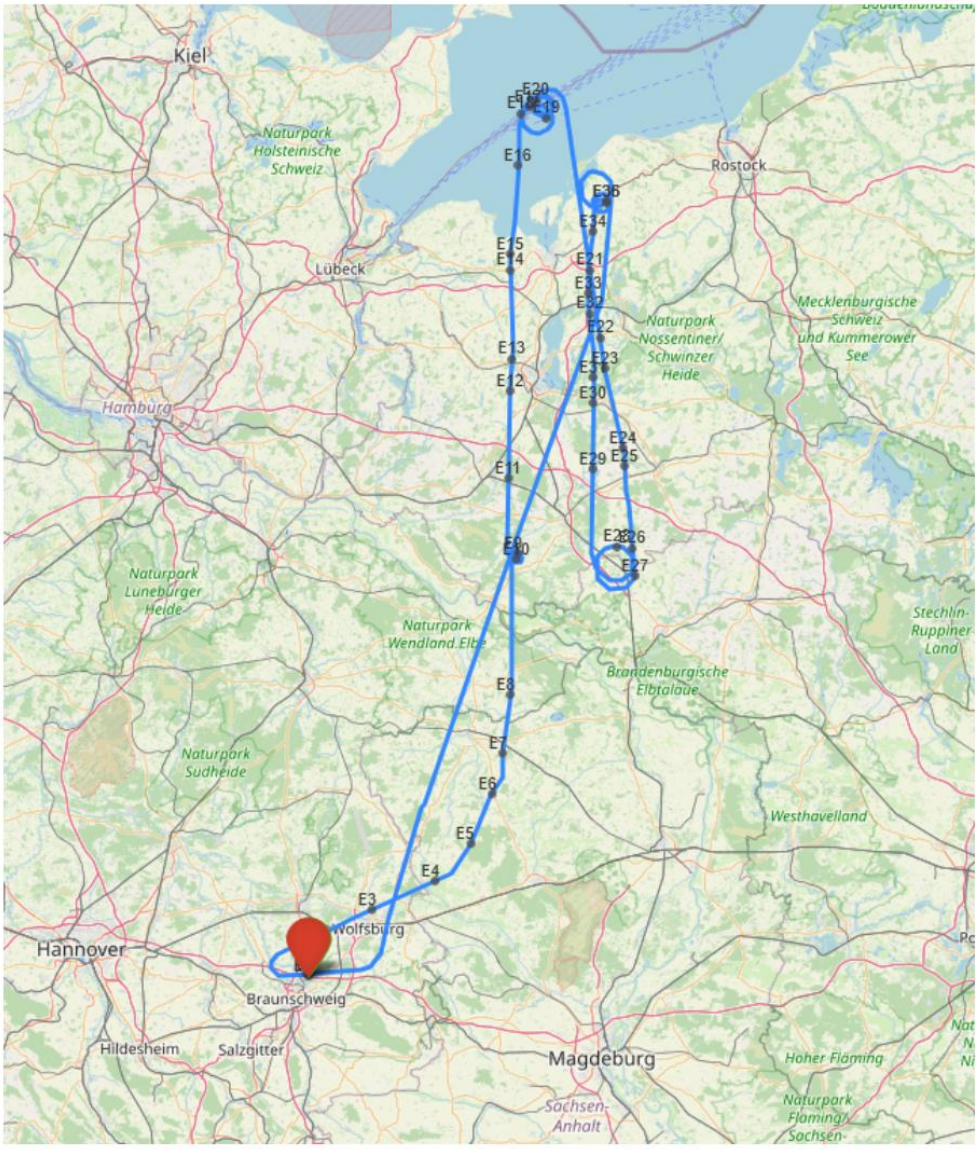
MEASUREMENT
Start UTC
09:10:00

Stop UTC
10:56:45.4

Duaration HH:MM:SS
1:46:45.40

METAR
Takeoff
EDVE 011120Z 28009KT 9999 SCT010 BKN017 OVC047 15/13 Q1017

Landing
EDVE 011250Z 27007KT 9999 FEW019 SCT028 OVC033 16/13 Q1018



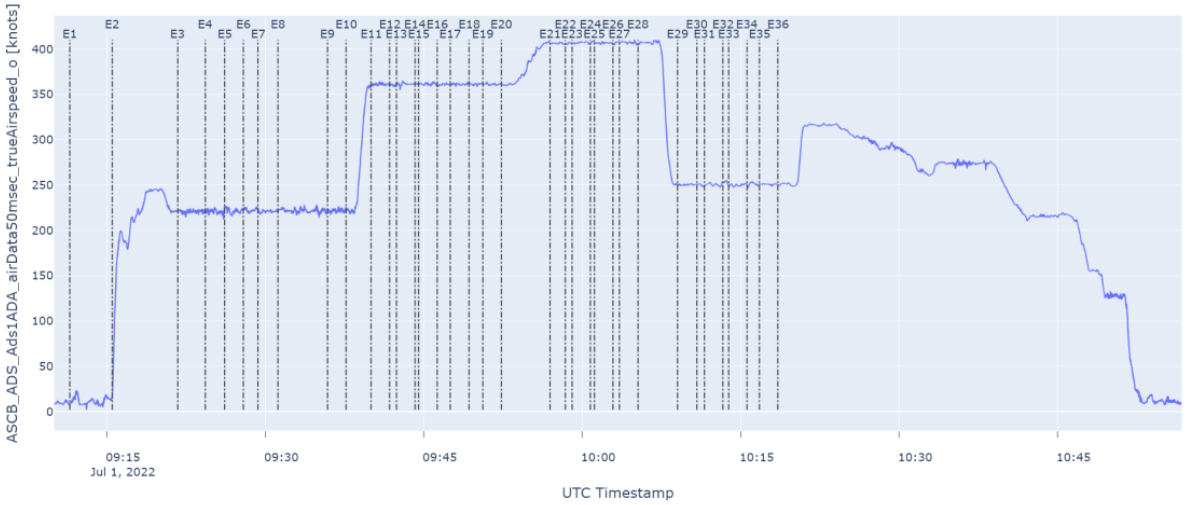
Comments / Flight Objectives
Evaluation of the SHM System in-flight

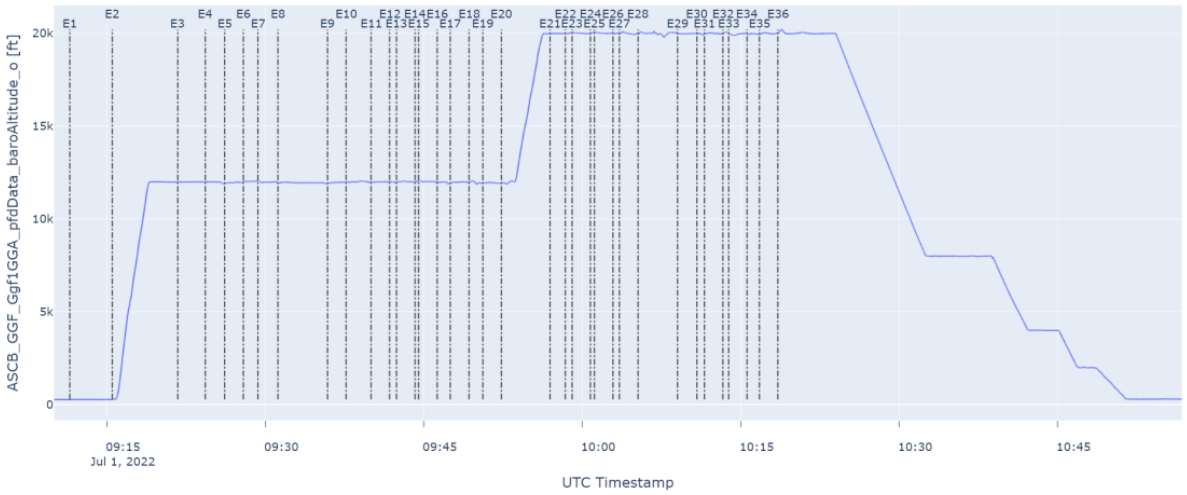
FTI Issues
-

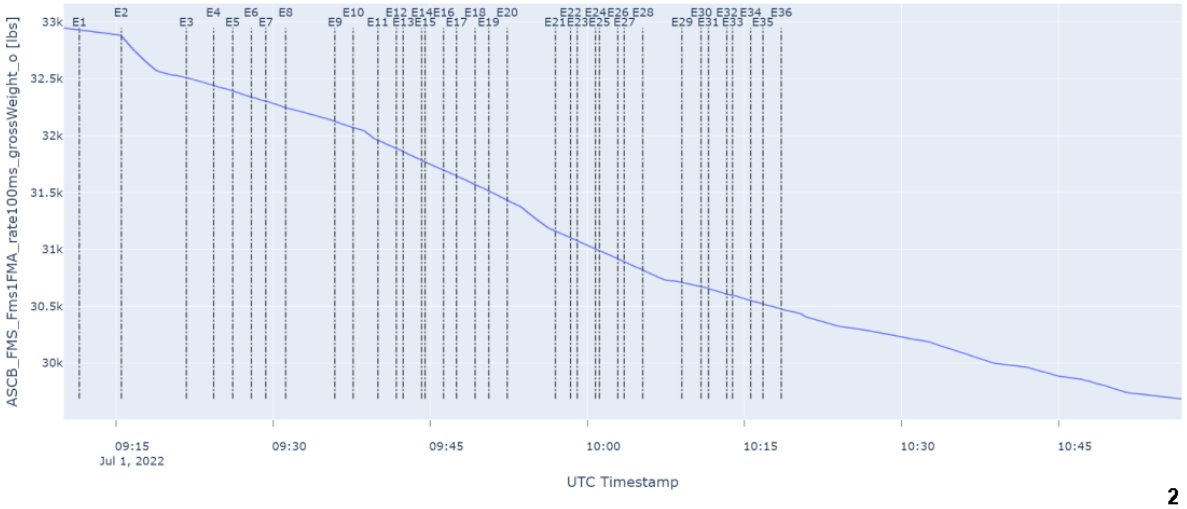
Flight Report: F89_20220701_ConMo_6

Aircraft: D-BDLR	Project Name: ConMo	twinstash ID: 65723d93c9cbc807e97fc82a

Sensor Overview







Flight Report: F89_20220701_ConMo_6

Aircraft: D-BDLR	Project Name: ConMo	twinstash ID: 65723d93c9cbc807e97fc82a

Maneuver from Flight Test Card

Man.	Start [UTC]	PA [FL]	SPD [KIAS]	Rudder [%]	AOB [deg]	Stop [UTC]	Comments / Events
S/L	09:21:40	120	180	/	/	09:23:40	#3 #4
SHSS RH	09:26:08	120	180	+100	-12	09:27:53	#5 #6
SHSS LH	09:29:18	120	180	-100	+13	09:31:00	#7 #8
G-Turn	09:35:50	120	180	/	+45	09:37:37	#9 #10

Man.	Start [UTC]	PA [FL]	SPD [KIAS]	Rudder [%]	AOB [deg]	Stop [UTC]	Comments / Events
S/L	09:39:58	120	300	/	/	09:41:42	#11 #12
SHSS RH	09:42:24	120	300	+30	-13	09:44:08	#13 #14
SHSS LH	09:44:30	120	300	-30	+13	09:46:14	#15 #16
G-Turn	09:47:30	119	300	/	+45	09:49:15	#17 #18

Man.	Start [UTC]	PA [FL]	SPD [KIAS]	Rudder [%]	AOB [deg]	Stop [UTC]	Comments / Events
S/L	09:56:32	200	300	/	/	09:58:20	21 #22
SHSS RH	09:58:57	200	300	+30	-13	10:00:44	#23 #24
SHSS LH	10:01:08	200	300	-30	+13	10:02:52	#25 #26
G-Turn	10:03:20	200	300	/	+45	10:05:17	#27 #28

Man.	Start [UTC]	PA [FL]	SPD [KIAS]	Rudder [%]	AOB [deg]	Stop [UTC]	Comments / Events
S/L	10:09:00	200	180	/	/	10:10:46	#21 #22
SHSS RH	10:11:31	200	180	+100	-13	10:13:17	#23 #24
SHSS LH	10:13:48	200	180	-100	+12	10:15:32	#25 #26
G-Turn	10:16:45	200	180	/	+45	10:18:31	#27 #28

Flight Report: F89_20220701_ConMo_6

Aircraft: D-BDLR	Project Name: ConMo	twinstash ID: 65723d93c9cbc807e97fc82a

Measurement Verification

Measurement Data Amount
4150625926 Bytes

Missing Sensors
-

Proximity Report

Nearest 3D distance km	Time of nearest distance	Altitude at nearest distance in m	Enter Time	Exit Time	Seconds in Range	Registration
2.77	10:54:17	2.77	10:54:17	10:55:15	58.0	D-CFMF