

Design and Development of Flight Data Visualization Tool for Post Flight Analysis

Shashi Ranjan^{}, Anand G, Manju Nanda, Jyothi Lakshmi S*

Specialist Design Engineer, CSIR-National Aerospace Laboratories,
Kodihalli Bengaluru, Karnataka, India, 560008

Abstract. Data Analysis and Visualization play a significant role today. The data represented graphically are more attractive and easily understandable. The paper aims to describe the design and development cycle of a flight data visualization tool. This tool facilitates a graphical analysis of critical flight parameters, which aids in deciding the upcoming operation during the testing stage. Data has always been a constant element in every field of work. Data analysis has evolved from manual to computational methods in recent years and is widely used for pre and post-research purposes. To achieve high accuracy and fast computation, high-level programming languages produce graphical data within a smaller framework of time, enabling quick analysis.

Keywords: Data Analysis tool, Python, GUI, Data visualization.

1. Introduction

The process of examining, transforming, modeling data, and presenting it into helpful information that provides context for the data in a visual representation enables the viewer to gain knowledge about the internal structure of the data and relationships in it is our goal. The graphical representation includes graphs, plots, and charts. Examining raw data to study patterns or trends can be tiring; here is where information visualization comes into the picture. This tool aims to provide a swift and accurate real-time evaluation of flight data in the field with minimal hardware requirements. This paper focuses on briefly describing the development life cycle of the tool. Many tools have been developed to visualize flight data. However, it is challenging to meet all the requirements of the target users. Certain professionals, like the aircrew, would like to learn about the flying environment and the aircraft itself. On the other hand, engineers and scientists must analyze and process the pre and post-flight data to understand its behavior, which will aid in developing and upgrading the development and upgradation of hardware and software components. The application can be used for appending the files in .CSV format in minimal time without error from the desired path and to plot the graph further.

* Corresponding Author: Shashirnjn14@gmail.com

The user can fetch the data, and the selection of parameters is up to the users to support the decision-making and comparisons of the data.

2. Related works

Several tools have been developed to visualize flight data. The data analyzing tools are not target specific to the users; the accuracy of the critical data analysis for safety is catastrophic. Previously the graphing and analysis have been done manually. The data obtained during the aircraft testing is then given to an analyst for analysis. The size of the data would dictate the length of time it takes. This process generally takes a great deal of time. Another downside of the conventional method is paperwork and storage. The quantity of paperwork generated would be massive, and the probability of human error is high. That makes the conventional method less desirable.

3. Proposed work

A model-based approach to design and development has the potential to overcome the limitations of conventional methods. The possibilities of automation are expanding with the development of technologies and programming. In these modern times, researchers increasingly use software to help them analyze data. Since the tool has been developed using python, its fastest-growing uses are data science and academic research. Python is an excellent tool for data visualization due to its simplicity and ease of use. With the help of python libraries such as matplotlib, this tool generates plots and subplots of the raw data fed into the system. It also facilitates a few essential functions that can be performed on the plots. In the conventional method, designing mathematical models is a labor-intensive process. The probability of human error is also reasonably high. The pitfalls of conventional methods include generating vast amounts of documents, high dependency on human resources, and non-automated accumulating, arranging, and sorting of data and storage trouble. The model-based approach, in contrast, performs better in all aspects, whereas the conventional method has a flaw. Fig 1 shows the different stages involved in the development cycle of the manual method for analyzing data, which makes the proposed method efficient to a greater extent.

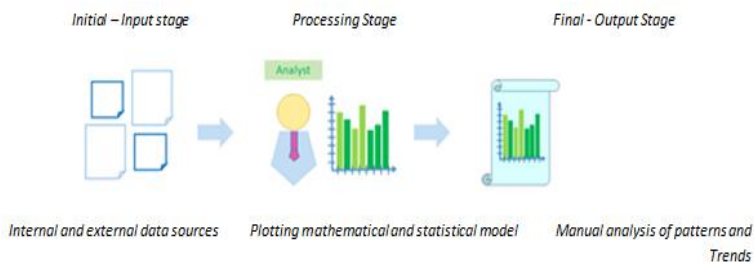


Fig. 1. Manual method of Analyzing Data

In manual methodology, if changes are to be made, the entire process must be repeated. Accommodating changes is quite a task and is highly time-consuming. If the analyst commits an error, the entire process must be executed again, which can be tedious and expensive. In contrast, introducing changes and modifications in an automated method is relatively easy. The model is refined to such an extent that error rates are lower than manually crafted analyses. Fig 2 shows the different stages involved in the development cycle of the automated method for analyzing data.

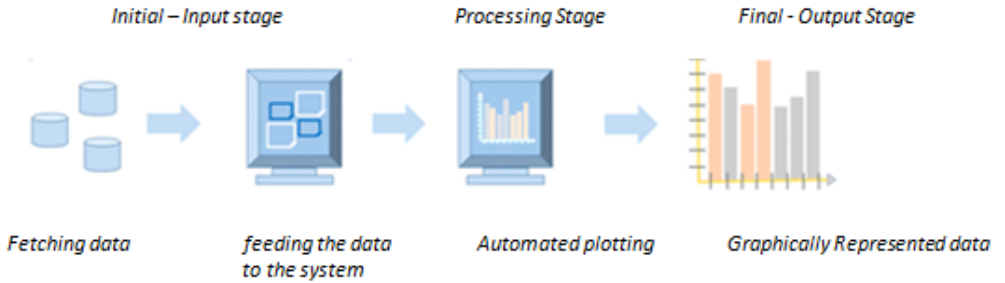


Fig. 2. Automated method of f Analyzing Data

The advantage of the software is that it allows for more efficient and effective work processes. It saves time, gives more profound insights into the data, and allows users to customize the tool according to their requirements. The software attracts non-technical users with no technical knowledge required to use the application, similar to the GUI of mobile phones, games, etc. Technical operators can use the apps in many industries without any prior knowledge

4. Implementation

The App developed is used in Data Analysing. The flight parameter is analyzed and visualized by the graphically plotted graph. Python has been a great tool in data science and machine learning. The GUI developed the Pycharm IDE 2021.2.1 version, which is an open-source tool. Python 3.10 is an interpreter, a general-purpose, high-level language used in many applications. Using python libraries such as GUI tkinter, numpy, pandas, matplotlib, visualizing data has become more accessible and can be customized according to the user's requirements. The GUI developed here supports the .csv files due to less latency during extracting flight Data. The GUI reads the labels of the columns and displays them in the Lists, which can be manually selected to know the individualities of the flight parameters, and options to append the .csv files are incorporated. The application is capable of frequency selection, or the required frequency at the data is extracted from the flight. The data from the aircraft operations at 256Hz, 16Hz, and 32 Hz frequencies can be plotted.

The software development and integration are explained with the V-model in SDLC, with verification and validation since there is a directly associated testing phase for every single

phase in the development cycle, which is a highly disciplined model, and the next phase starts only after the completion of the previous phase.

4.1 Verification Phase

Requirement Analysis is the first phase in the development cycle where the initial requirements of the product are discussed and noted. The initial requirement of this tool was to graphically represent flight parameters in four subplots, which can later vary to either a single plot or eight different subplots. The parameter selection entirely depends on the user.

System Design is the complete design of the system. The hardware requirements of this tool are pretty basic. The executable file should execute in all the systems. Any system that supports python scripting is sufficient since the result will be generated in an executable file. A .exe file can run on any system, irrespective of having a python compiler or not.

Architectural Design includes architectural specifications and designs. Generally, a few technical approaches are proposed and documented that fulfill the technical criteria. Architectural design is divided into various 'modules' with different functionalities.

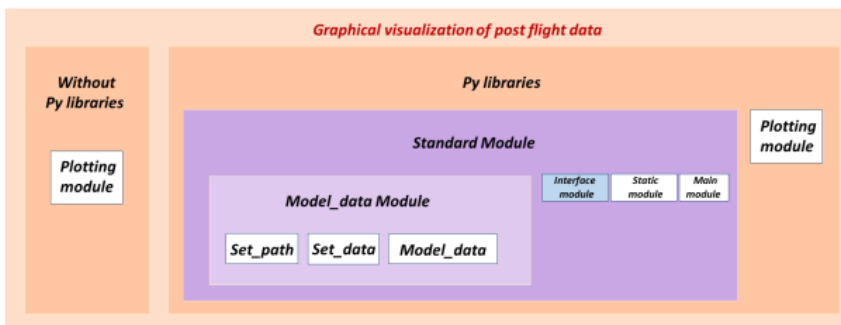


Fig. 3. System Architecture

Module Design is an in detail internal design for all system modules. It is referred to as low-level design, whereas architectural design is also called high-level design.

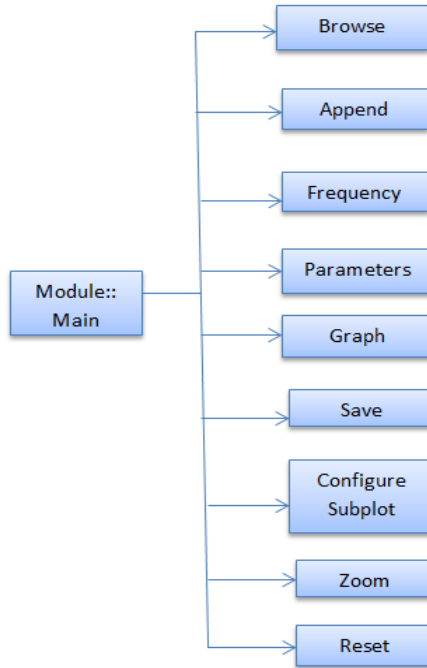


Fig. 4. Modules diagram

4.2 Validation Phases

Unit testing takes place during the validation phase. At the initial level, to eliminate bugs, but on the other hand, faults cannot be discovered. Integration testing happens in the architectural design phase to test whether the internal communication and coexistence of the internal modules are taking place harmoniously. System testing happens in the system design phase. Its main task is to check the entire system's functionality. It validates how the system communicates with the external environment. Software and hardware compatibility issues are detected in this stage, and to run the App in different systems. Acceptance testing takes place in the requirement analysis stage. It checks whether the product is compatible with all systems, such as screen resolution, orientation, etc. Performance defect(s) in an actual user environment is identified here.

5. Methodology

Data analysis is analyzing and visualizing the data to discover useful information, support decision-making, and inform conclusions. The tool generates real-time graphically visualized data. The App accepts data files in '.CSV' format. The generated output has a few operations that can be performed, including saving and zooming in and out of all subplots simultaneously.

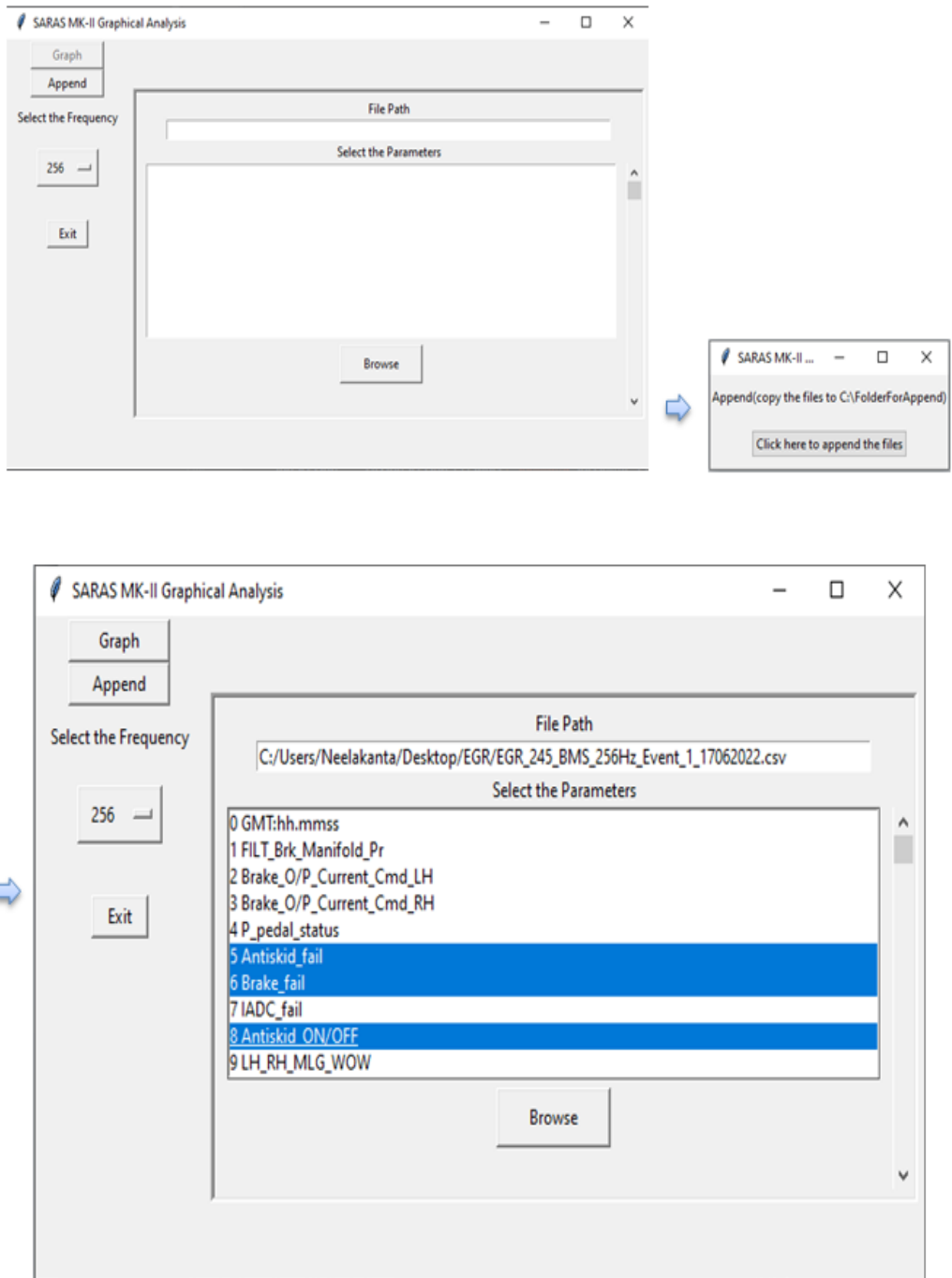


Fig. 5. Execution flow

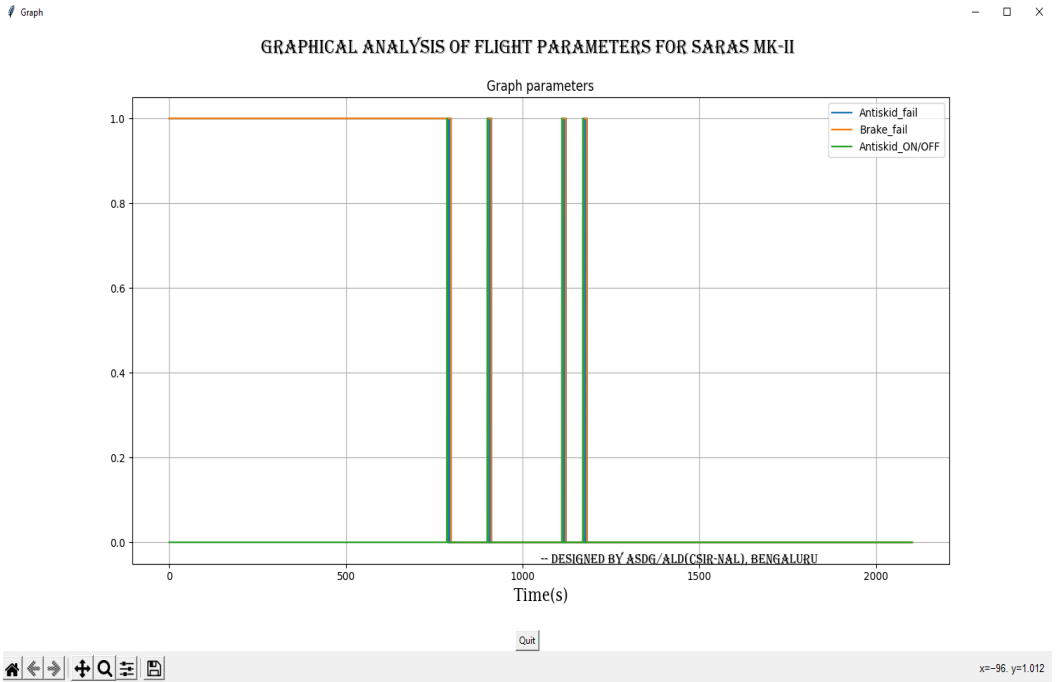


Fig. 6. Output Graph

- A file in .CSV format is fetched as an input.
- If multiple files exist, an option to append is provided to link them into a single file.
- After browsing for files and feeding them into the application, the next step is plotting graphs.
- The frequency selection is optional, with 16hz, 32hz, and 256hz options based on which the output will be generated on the chosen frequency.
- The plotted graph is then displayed where a few operations can be performed.
- Save a copy of the graph in a PNG format for future reference.
- Finally, the process can either be repeated or concluded.

6. Conclusion and Future Works

An efficient way to analyze the data of flight, graphically represent the data for comparisons and conclusions. This tool has produced three variations: a single plot, four subplots, and eight subplots. The user has the freedom to choose input parameters according to the requirements. Frequencies of higher ranges can also be added if desired with hardcoding. The drawback of the software is only .CSV files can be chosen as of now and will be upgraded for other types of files in the future. The App can still be user-friendly with additional features like toggles of legends with the graph lines.

References

1. Lovin K. Jose, Manju Nanda, S. Poonam Patil, and G. Anand. Model-Based Code Generation for Avionics Display Graphics. Computational Algorithm for AI Technology, Proceedings of ICAIECES, 2020.
2. Reshma Rajkumar, Manju Nanda, J. Jayanthi, and G. Anand. Model-Based Test Approach on EICAS Ground Testpages Using SCAD Display Software. Computational Algorithm for AI Technology, Proceedings of ICAIECES, 2020.
3. G. Anand, Manju Nanda, K. Nayana, and J. Jayanthi. Avionics Display for Two-Seated Aircraft Using OpenGL. Computational Algorithm for AI Technology, Proceedings of ICAIECES, 2020.
4. S. Kumar, M. Nanda, P.R. Rao, L.K. Jose, Model-based approach for design and development of avionics display application, in ICETE (2019)
5. Aghassibake, N., et al. (2020). Supporting data visualization services in academic libraries. *The Journal of Interactive Technology and Pedagogy*, 18.
6. Jong-Ik Shim, Gyu-Se Jo. Flight Data Visualization and Post-test Flight Data Analysis System by Using Database. *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS Vol:16 No:02*.
7. Dr. Jae Sug Ki. Study on Developing a Flight Data Visualization. *Journal of the society of Korea Industrial and Systems Engineering*. Vol 26, No 3, pp.74 -78, September 2003.
8. Ravi Shanker Yadav. Improvement in the V-Model. *International Journal of Scientific & Engineering Research*, Volume 3, Issue 2, February 2012.
9. Mahmoud Sherif Zakaria. Data visualization as a research support service in academic libraries: An investigation of world-class universities. *The Journal of Academic Librarianship* 47 (2021).
10. Qing Liu, Zheng Qiao, Yu Lv. PyVT: A python-based open-source software for visualization and graphic analysis of fluid dynamics datasets. *Aerospace Science and Technology* 117 (2021).
11. Michael Riesener, Christian Doelle, Stefan Perau, Philipp Lossie, Guenther Schuh. Methodology for iterative system modeling in agile product development. 31st CIRP Design Conference 2021.