Real-Time Data Visualization in Telemetry Systems

M. Ufelek[1], O. Yoruk[2], M.E. Cakici[3]
Turkish Aerospace Inc., Fethiye mah. Havacilik Blv. No:16 Kahramankazan/Ankara, Turkey
[1]mustafa.ufelek@tai.com.tr, [2]oyoruk@tai.com.tr,[3]muhammedemir.cakici@tai.com.tr

Abstract

It's highly critical to acquire, store, evaluate and analyze accurate real-time telemetry data during a test flight. Flight Test Management Centers are designed and used for this purpose. However, the capacity of Flight Test Management Centers are limited and majority of design engineers could not attend the test flight. The need of an application that lets design engineers to monitor the ongoing flight test from their workspace has occurred. For that reason, a software has been developed for real-time data monitoring and extracting.

On kick off of this project, two main objectives are aimed. The first one was to develop a smart system, which helps design engineers to monitor critical flight parameters in real-time. The second one was to prevent human faults with pre-defined caution/error indicators and make predictions by using data-driven algorithms.

The first phase of the project has determined as to gather data from flowing telemetry stream simultaneously in real-time and to store critical parameters in a time series database. The second phase has determined as to visualize data in comprehensible and user-friendly dashboards by selecting proper widgets. The last phase includes developing data models by using machine learning algorithms to make predictions regarding to previously stored flight data.

Key words: Data Acquisition, Real-Time Data Visualization, Data Analysis, Flight Test Database, Big Data

The Idea Behind

As Mustafa Kemal Ataturk told 100 years ago, 'The Future is in the skies'. Aviation and Aerospace Industry has boomed in the last 100 years and it is still growing day by day. Requirements are changing and companies are designing new products to meet that requirements. Time is more important than anything else today. No matter how good the design is, the product could not succeed if it has not been served to the market on time. For that reason, the design should be transformed into the product very quickly.Testing activities are vital in aviation and aerospace industry, significant time is allocated to flight testing activities in the product development process. These flight tests should be accomplished very carefully, in a very short time period. There is a need to collect as much data as possible from the experimental aircraft and analyze this data accurately. Rapid data analysis allows project Flight Test Team to be ready for the next flight test. Flight Test Management Centers are designed to visualize real-time data received from the aircraft and manage the flight test according to this data. However, the capacities of flight test management centers are limited and designers who are not directly related with the ongoing flight test does not attend to the test. In many tests, design engineers also need to monitor real-time data; so the data should be available under authentication control outside of the test management centers. First aim is to extend the capacity with an in-house web-based software that allows engineers to monitor the data directly from their workspaces. Second aim is to store real-time data that is collected during the flight in a time-series database and use the stored data for further analysis. Thus, while the data is visualized in real time, a flight test database will be created.

Telemetry data includes parameters that are critical for analyzing the design. The data must be stored in a database for design engineers, product teams and flight test engineers. The Flight test team should be able query this data with different criterias. For example, comparison of values for the same parameter on test flights in various weather conditions or test points is very important.

After obtaining a sufficient amount of data, predictions can be made by using machine learning algorithms. As the size of the flight test database grows, the data model built on it will work with higher accuracy. Therefore, storing and labeling parameter data in each new flight test is critical.

Fig. 1. A Flight Test Management Center, Turkish Aerospace Industries Inc.

Current Design of Telemetry Network

The flight test management center owns a seperated network called ‘Telemetry Network’. Desktop applications (such as IADS) runs on client computers which are called test monitoring consoles. Flight test team can monitor the flight ontest monitoring consoles. Every test monitoring console runs a customized flight test monitoring display which has designed uniquely according to the flight test team members requests.

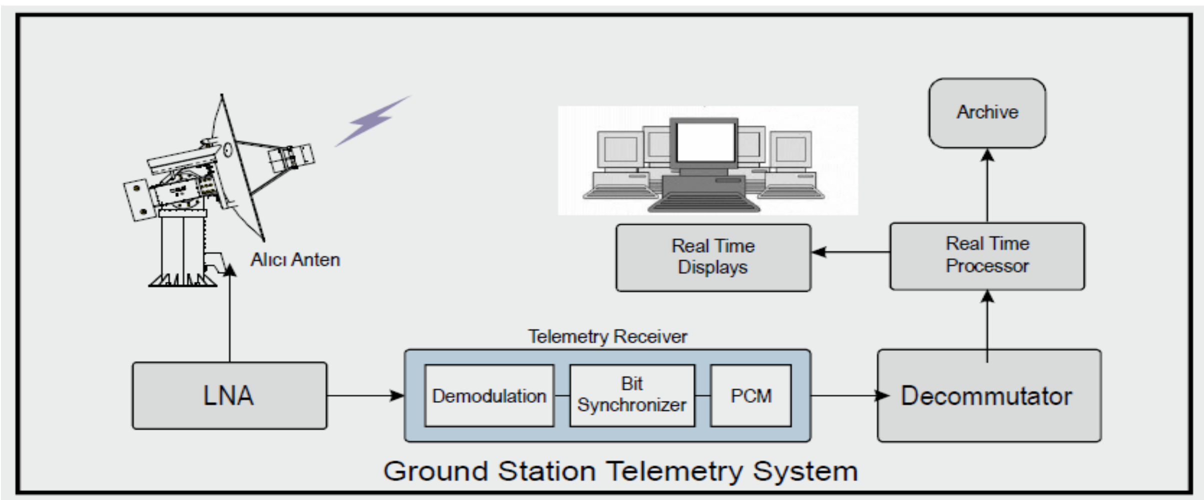

Fig. 2. Telemetry System Design

Extended Design

Turkish Aerospace has an internal network of approximately 10000 clients. This network is isolated from the Internet and telemetry network. Employees use computers on this network to carry out their jobs. Therefore, they need to access telemetry data through this network. The design which has shown on Fig. 2 has extended as shown in Fig. 3. Conversion of signals from telemetry is done by Decommutator. A software that fetchs data after this transformation and inserts into database, has been developed. As shown in the Fig. 3, this software receives parameter data from the source in the telemetry network and inserts into a database in the company network where the employees' workspaces are located. The database is a time-series database and data is stored with its sampling rate. There is an analytics platform running in the server. This analytical platform provides real-time visualization of data recorded in the database.

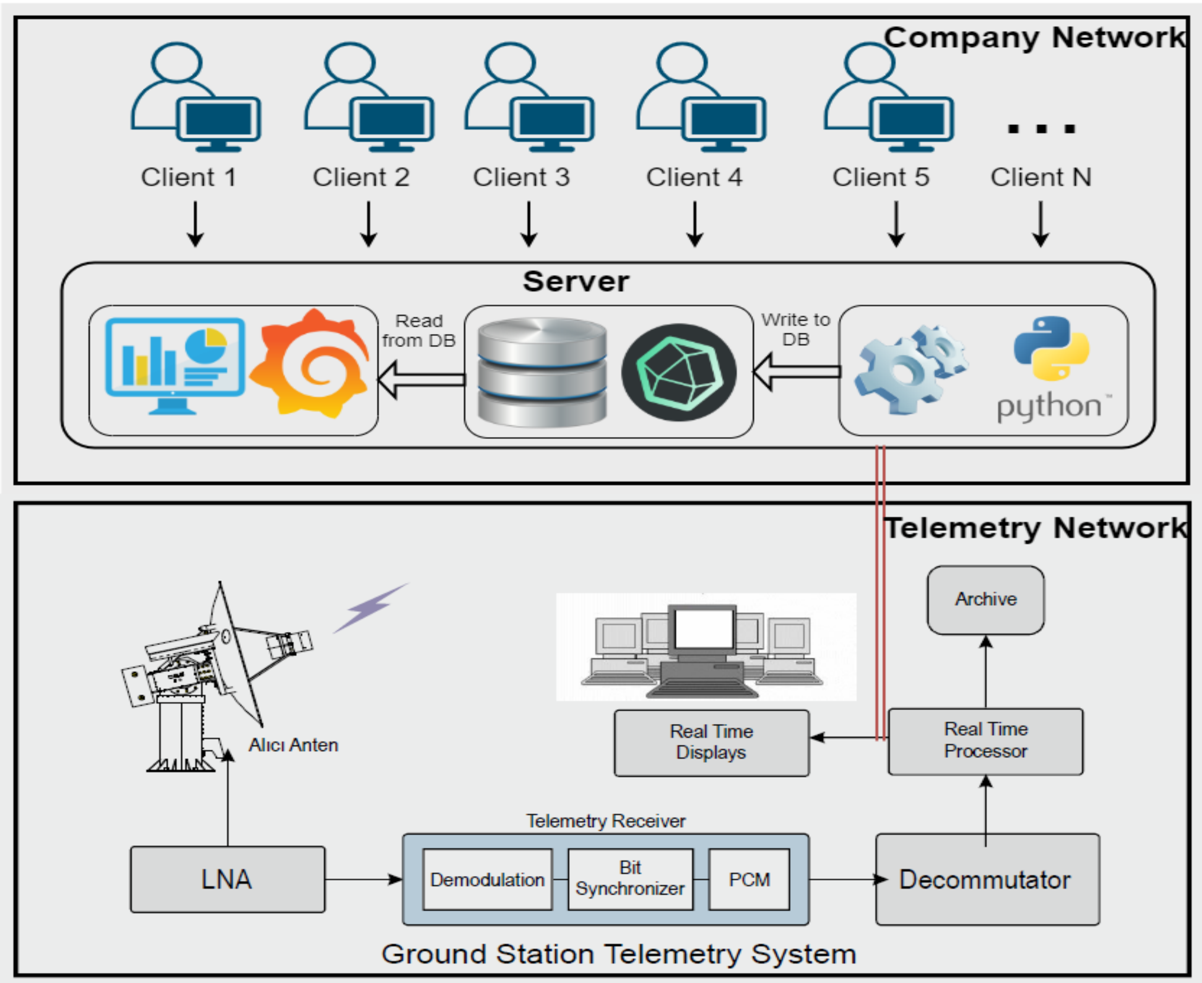

Fig. 3. Access data through Company Network

Phases of Solution and Techologies Used

Three stages had to be applied to implement the solution. In order to implement these processes, technologies commonly used in software world such as Python, Grafana, InfluxDB and Docker have been selected. Docker has been used to run container structures. Python programming language has been used to develop software that fetches real-time telemetry data and inserts into the database. Influxdb has been used to store data, and Grafana has been used to visualize stored data.

Fig. 4 Technologies used

DOI 10.5162/ettc2020/3.3

Phase 1: Data Acquisition and Storage

The first phase was to develop an application that reads telemetry data in real time and stores into a time-series database. In telemetry network IADS CDS (Cache Data Server) is used to serve data to client desktops. In addition to this, IADS provides an API to get this data programmatically. This interface has been used to read real-time data. Through this interface, it is possible to get not only parameter data but also many additional data such as test point logs, event logs, parameter list, parameter calibration definitions, and mission attributes tables.

The software has been developed with the Python programming language. It is a desktop application that runs in a server in the telemetry network. Since the data rate is very high, the software has been developed to operate at maximum speed and it works with a large number of threads to carry out the operations. It is a desktop application without any visual components and works as a service. Parameter names can be configurable for different type of aircrafts. Information about which parameters will be written to the database in real-time is recorded in a configuration file and the software performs data reading and writing operations in accordance with this information.

For the selection of the database, the most important criteria was the speed of writing and reading. Planned writing and reading speed was designed to be one tenth of a second. This means that all selected parameters are both be written and read within 100ms. Even delays in the level of seconds can cause data to be misinterpreted. Since the data is processed on time based, this database must be a time-series. One of the most used time-series databases in the software community, Influxdb was chosen to meet that requirement. In addition to parameter data, the database contains metadata about the flights. Test point and mission attribute data are also stored in a relation in the database. Storing these data together makes it easier for flight test engineers to make detailed analyzes with accumulated data. At the end of the test flights, a telemetry-specific flight test database has been created and this database is accesible for all users to run complex queries.

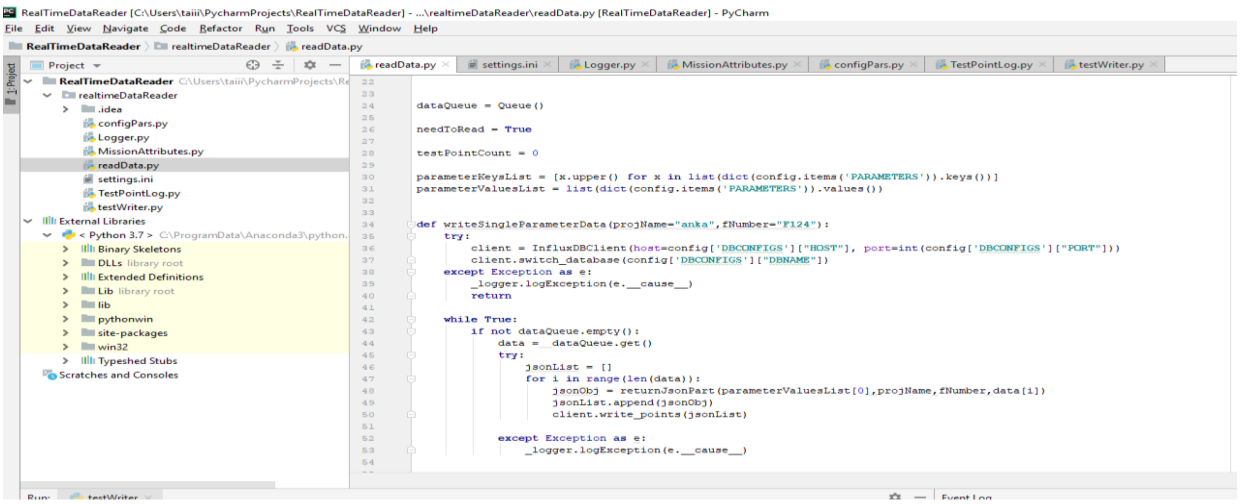

Fig. 5. Real-time Data Reader and Writer Application

time	ALTITUDE	AOA	CAS	COR...	HEADING	HYBRIDPITCH...	HYBRIDPITCH...	HYBRIDROLL...	HYBRIDROLL...	HYBRIDYAWR...
27.03.2020 06:...	2745,22607421...	-4,4252333641...	2,10891342163...	0	120	-4,4494915008...	-2,4971110820...	-0,2485307306...	0,17808306217...	-0,3792332410...
27.03.2020 06:...	2745,14550781...	-4,8252334594...	1,87440657615...	0	120	-5,0582151412...	-2,8497934341...	-0,1246691495...	0,84451377391...	-0,0677318871...
27.03.2020 06:...	2745,14550781...	-4,8252334594...	1,87440657615...	0	120	-5,0582151412...	-2,8497934341...	-0,1246691495...	0,84451377391...	-0,0677318871...
27.03.2020 06:...	2745,08569335...	-5,2252335548...	1,74128425121...	0	120	-5,3413114547...	-0,4293925464...	-0,0016370285...	0,08481176942...	0,07580327242...
27.03.2020 06:...	2745,08569335...	-5,2252335548...	1,74128425121...	0	120	-5,3413114547...	-0,4293925464...	-0,0016370285...	0,08481176942...	0,07580327242...
27.03.2020 06:...	2745,09375	-5,1718058586...	1,66713643074...	0	120	-5,1718058586...	1,50386404991...	-0,1040408164...	-0,5141221284...	-0,1829194873...
27.03.2020 06:...	2745,09375	-5,1718058586...	1,66713643074...	0	120	-5,1718058586...	1,50386404991...	-0,1040408164...	-0,5141221284...	-0,1829194873...
27.03.2020 06:...	2745,13403320...	-4,7729296684...	1,87976527214...	0	120	-4,5896682739...	3,78622317314...	-0,0553448908...	1,22339463233...	-0,4451417624...
27.03.2020 06:...	2745,13403320...	-4,7729296684...	1,87976527214...	0	120	-4,5896682739...	3,78622317314...	-0,0553448908...	1,22339463233...	-0,4451417624...
27.03.2020 06:...	2745,24194335...	-4,3729300498...	2,09696102142...	0	119	-3,7587034702...	3,90318799018...	0,28218060731...	1,50220096111...	-0,4891535341...
27.03.2020 06:...	2745,24194335...	-4,3729300498...	2,09696102142...	0	119	-3,7587034702...	3,90318799018...	0,28218060731...	1,50220096111...	-0,4891535341...
27.03.2020 06:...	2745,251953125	-3,9729297161...	2,24832463264...	0	119	-3,2896273136...	0,95765155553...	0,36638915538...	-0,4756725728...	-0,2474727183...
27.03.2020 06:...	2745,251953125	-3,9729297161...	2,24832463264...	0	119	-3,2896273136...	0,95765155553...	0,36638915538...	-0,4756725728...	-0,2474727183...
27.03.2020 06:...	2745,20703125	-3,5729298591...	2,32653975486...	0	119	-3,5330836772...	-2,4283320903...	0,18199421465...	-0,8328031301...	-0,1916854083...
27.03.2020 06:...	2745,20703125	-3,5729298591...	2,32653975486...	0	119	-3,5330836772...	-2,4283320903...	0,18199421465...	-0,8328031301...	-0,1916854083...
27.03.2020 06:...	2745,15087890...	-3,9479556083...	2,39358377456...	0	119	-4,1031551361...	-2,7248795032...	0,11225530505...	0,08456897735...	-0,3732891678...
27.03.2020 06:...	2745,15087890...	-3,9479556083...	2,39358377456...	0	119	-4,1031551361...	-2,7248795032...	0,11225530505...	0,08456897735...	-0,3732891678...
27.03.2020 06:...	2745,13354492...	-4,3474926948...	2,43820834159...	0	119	-4,3474926948...	-0,3635543286...	0,15531104803...	0,10412547737...	-0,1769526004...
27.03.2020 06:...	2745,13354492...	-4,3474926948...	2,43820834159...	0	119	-4,3474926948...	-0,3635543286...	0,15531104803...	0,10412547737...	-0,1769526004...
27.03.2020 06:...	2745,17944335...	-4,2308955192...	2,42384910583...	0	119	-4,2308955192...	0,90229403972...	0,17909003794...	0,23640976846...	-0,1927882581...
27.03.2020 06:...	2745,17944335...	-4,2308955192...	2,42384910583...	0	119	-4,2308955192...	0,90229403972...	0,17909003794...	0,23640976846...	-0,1927882581...
27.03.2020 06:...	2745,15942382...	-4,0896863937...	2,32270193099...	0	119	-4,0896863937...	0,19924426078...	0,18185110390...	0,00061969406...	-0,0375448577...
27.03.2020 06:...	2745,15942382...	-4,0896863937...	2,32270193099...	0	119	-4,0896863937...	0,19924426078...	0,18185110390...	0,00061969406...	-0,0375448577...
27.03.2020 06:...	2745,15112304...	-4,1943354606...	2,06723594665...	0	119	-4,1943354606...	-1,2581508159...	0,12352048605...	-0,4274314045...	-0,0105015132...
27.03.2020 06:...	2745,15112304...	-4,1943354606...	2,06723594665...	0	119	-4,1943354606...	-1,2581508159...	0,12352048605...	-0,4274314045...	-0,0105015132...

Fig. 6. Sample Parameter Data

DOI 10.5162/ettc2020/3.3

Phase 2 : Visualization

Visualization of the data via using proper widgets is as important as acquiring it. Stored data should be converted to visuals at high speed and this visuals should be easy to understand and interpret. Considering the speed of writing data to the database, it should be displayed on the dashboard with a delay of up to one second. Since the data is stored in a time-series database, the visualization technology has to support this type of database.

As the data visualization application will be the gateway to data access and will operate on the same network with all internal users, authorization control is very important. Only authorized personnel should be able to access the dashboards to be developed. Therefore, a solution with a strong security structure and in accordance with the authorization policy used in the network should be chosen. It should also be customizable and extensible. Although there are some differences according to the technology used, generally there are standard visual sets. These visual sets may not meet the requirement when the working area is very specific. Since aviation and aerospace is a very specific field, it cannot be expected that the products on the market to contain the visual sets used in this field. Therefore, it should be open to custom development. Grafana is the right choice from this point of view. As many custom plugins can be found and used, it is not difficult to develop a new custom plugin. As a matter of fact, some of the visuals used in the dashboards are customized for custom use within the company. The infrastructures used in this phase were preferred to be open source. Community support is very important on these open source platforms. This criterion was also taken into consideration during the selection.

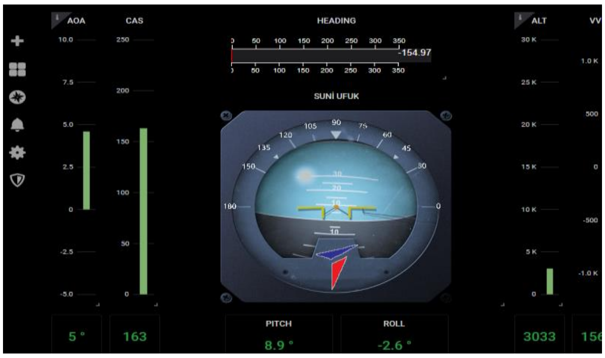

Fig. 7. Artificial Horizon-Community Plugin extended

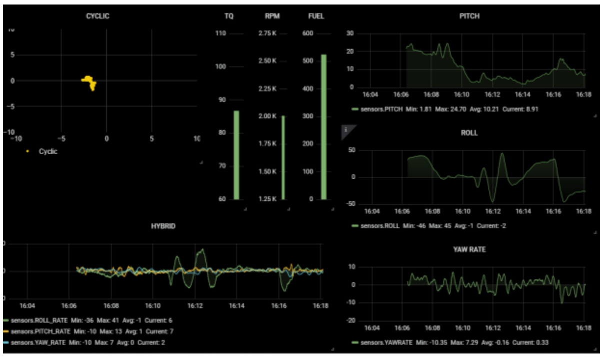

Fig.8 Sample Visuals from Flight Test Dashboard

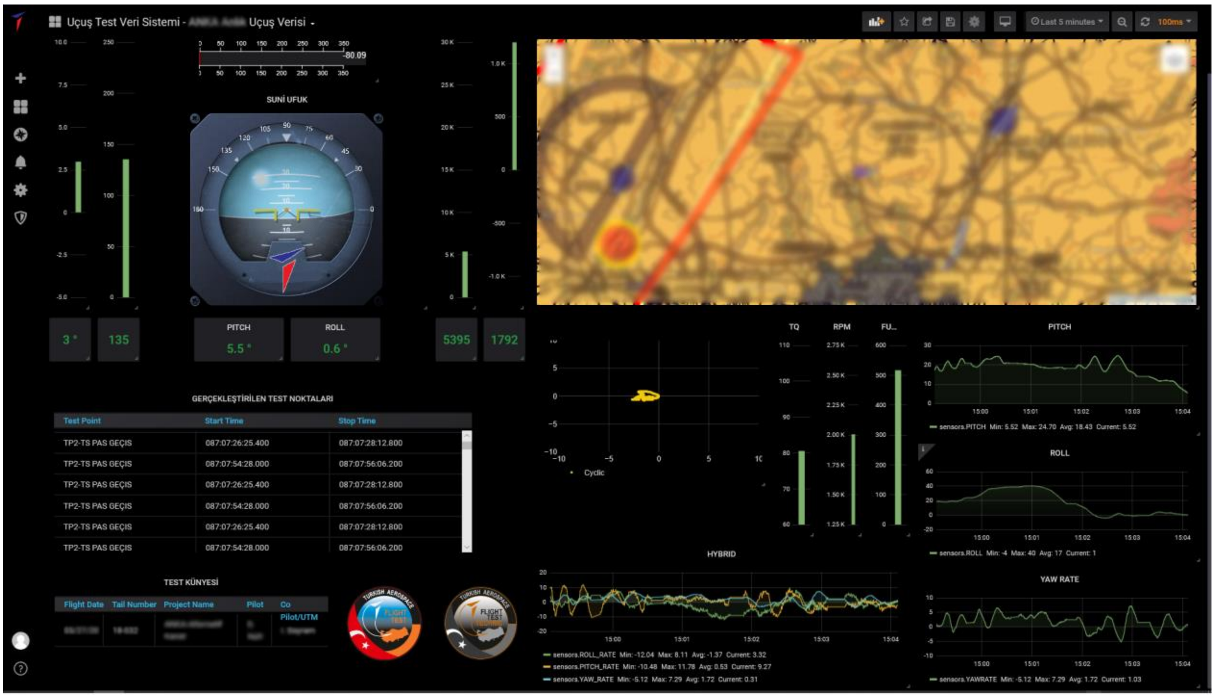

Fig 9. Flight Test Dashboard

Phase 3: Post-Flight Further Analysis

It was mentioned that data is being stored into a time-series database while visualization process. Thus, a database where post-flight analysis can be done after the flight was obtained. Storing flight data with this method helps to analyze the behavior of one or more parameters between different flights, test points and conditions such weather, aircraft configuration etc. Furthermore, important flight test data that is acquired can be transformed into more valuable way by using machine learning methods. One of the most important objectives of flight tests is to detect abnormalities on the aircraft. These abnormalities are very prone to faults when attempted to be detected by the human eye. Data models fed with sufficient data are very successful in detecting these abnormalities. Machine learning techniques are used to detect these abnormalities. Both of the two most important types of machine learning methods, supervised and unsupervised learning, can be used.

Supervised Machine Learning

In case the inputs and the outputs resulting from these inputs are known, it is called supervised machine learning. So there is a function like **f (x) = Y.**

Here, x is input and Y is output. The aim is to predict Y value based on previous data for new x values. The data should be tagged here. For example, the engine failure that occurs in a certain combination of torque, engine speed and oil temperature values should be labeled. Thus, the data model will evaluate these parameters and predict whether an engine failure will occur.

Unsupervised Machine Learning

Unsupervised machine learning is one of the widely used methods. It allows for anomaly detection by clustering method in cases where Y result is unknown corresponding to x inputs. What is done here is to group each parameter by evaluating the values of previous data, even if Y is unknown, and to identify values outside this set.

Hundreds and even thousands of parameters are collected at high frequencies according to the measurement type during a test flight. It is almost impossible for a human to evaluate such this amount of data instantly and make right decisions. At this point, where human abilities are limited, using computer-aided methods will increase the safety and effectiveness of test flight. In order to use these methods, a flight test database is required first. As mentioned in the first part of the article, while the data is visualized, it is also recorded into the database. By using this database, the data model is trained which values are normal under different conditions for each parameter. To give a simple example, when the engine speed is 3000 rpm and it is known that the oil temperature should be between 80 to 90 degrees celsius, it can be considered as an anomaly when this temperature rises above 90 degrees. Although this is easily detectable by a human and can be resolved with a simple warning mechanism, more complex techniques are needed considering the mentioned data. In this example, an inference was made by evaluating only two parameter and one rule. Although it varies depending on the type of aircraft, thousands of parameters from hundreds of sensors in an aircraft can be collected at the frequency of a few kilo hertz. In a dataset of this size, it is impossible to evaluate all parameters that may be related to each other instantly. Moreover, while the relationship between engine speed and oil temperature is strictly known in the given example, thousands of parameters in the same dataset could be related to each other. Therefore, a solution has been developed to detect abnormalities.

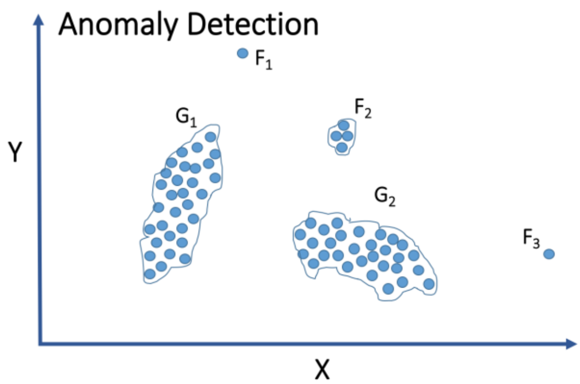

Fig 10. Anomaly Detection (Clustering)

Conclusion

The visualization and evaluation of the data collected in flight tests with analytical systems has a great importance for the schedule, cost and success of the project. With the completion of this software, it is predicted that the flight test process carried out within the company will be accelerated.