

AVALON Evolved: New Features and Enhanced Applications

Francisca Coll Herrero¹, Pedro Rubio Alvarez¹

¹ Airbus Defense and Space, Av John Lennon s/n Getafe, Spain
Francisca.coll@airbus.com, Pedro.R.Rubio@airbus.com

Abstract:

AVALON is the right onboard automated tool to enhance the in-flight real-time characterization of a flexible structure, inferring relevant structural parameters such as normal modes frequencies and structural damping. Last year, the AVALON project was one of the four finalists in the Awards of engineering of ADS, ensuring the continuity of its development. This paper presents the latest advancements in AVALON, focusing on new features and significant improvements that enhance its functionality and performance. The technical enhancements implemented since the initial release are detailed, including algorithmic optimizations, calculus of the residues, new interfaces in AVALONVIEW, and more. Through case studies and experimental results, the improved AVALON framework is demonstrated to effectively represent an enhancement in the traditional analysis of vibration calculation for Operational Modal Analysis (OMA).

Key words: AVALON: Automated Vibration Analysis and Localization Oscillation Nodes, Telemetry, Modal analysis, Vibration, Power Spectral Density, WTT: wind Tunnel Test, GVT: Ground Vibration Test, Real Time Analysis, OMA: Operational Modal Analysis.

Introduction

In the 2025 Congress, the AVALON project was presented (reference [1]). AVALON is a fully automatic tool for the analysis of vibrations in flexible structures. The tool consists in two main components: one for performing automatic calculations AVALON and another for displaying the results AVALONVIEW. AVALON arise from the need to migrate the jFlutter algorithm from Fortran to C.

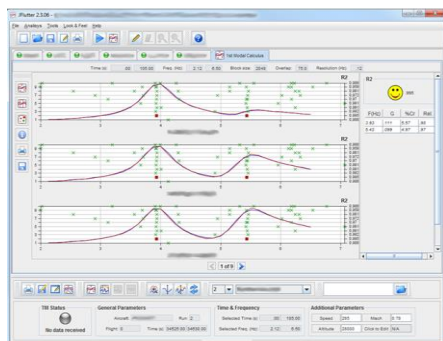


Fig 1. jFlutter

AVALON is funded by Clean Aviation HERA (Hybrid Electrical Regional Architecture). The initial objective of the development of AVALON

is to be used in Wind Tunnel Tests of the HERA scale model.

This paper discusses the new functionalities and improvements of the AVALON tool, as well as its various applications.

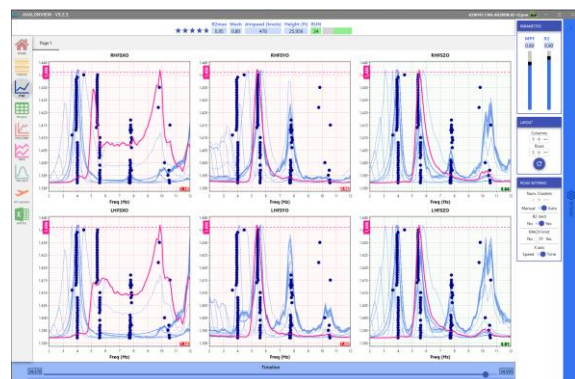


Fig 2. AVALONVIEW

AVALON architecture. Global Overview

AVALON and AVALONVIEW could be used in online and offline mode.

In offline mode, the origin is the recorded flight data. AVALON uses a configuration JSON file to access the parameters via the offline data server FxS.

The configuration file, in addition to containing the names of the parameters, also includes the calculation characteristics such as the frequency ranges for the analysis, characteristics of the PSD (Power Spectral Density) such as block size, window, and overlapping.

AVALON performs the calculations defined in the configuration file and generates a JSON file containing all the results.

AVALONVIEW accesses the JSON file and displays the results

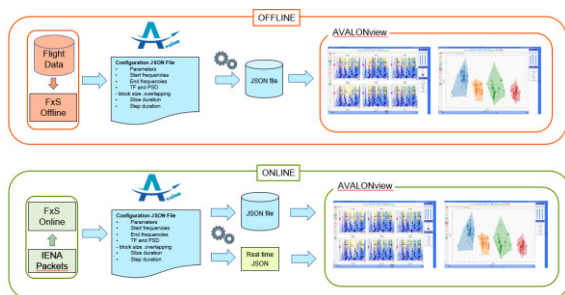


Fig 3. Online and offline mode

In on line mode, the origin is the IENA Packets, the configuration file is the same as online mode.

AVALON uses a configuration JSON file to access the parameters via the online data server FxS.

AVALON performs the calculations in real time and generates partial JSON files containing the vibration analysis and transmits them via telemetry to the ground station. The complete JSON is also stored onboard.

AVALON architecture. Detailed Process

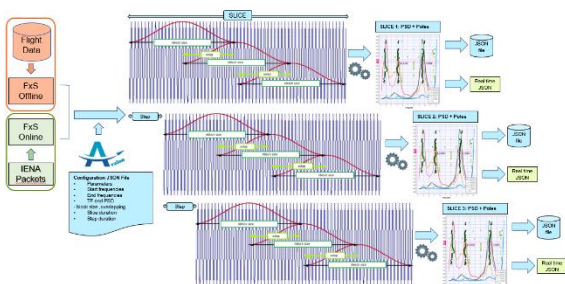


Fig 4.Detail Process

The configuration file contains two relevant parameters to understand the content of the result file. These parameters are:

- Duration Slice: Duration of each analysis in seconds.

- Interval size: displacement time between one analysis and the next in seconds.

The steps are as follows: after the first "Duration Slice" seconds have passed, and for each "interval size" rate, the Power Spectral Densities (PSD) and the corresponding modal parameters are calculated, and a JSON file is generated.

Each PSD and modal analysis calculations corresponds to a SLICE.

The time assigned to each SLICE is the final time of the analysis segment.

AVALON architecture. Small TM bandwidth

This type of architecture is applied in online use case where AVALON calculations are performed on-board and the results are sent by telemetry to a ground station.

This architecture is particularly suitable when the aircraft has a small telemetry bandwidth. Why? Because only the results are sent to the Ground Station (GS) via telemetry, reducing the amount of data transmitted.

For small-scale aircraft such as HERA or SIRTAP, AVALON is implemented on a microcomputer-based platform.

It can also be installed on big aircrafts like A400M, MRTT and C295 that have an FTI console.

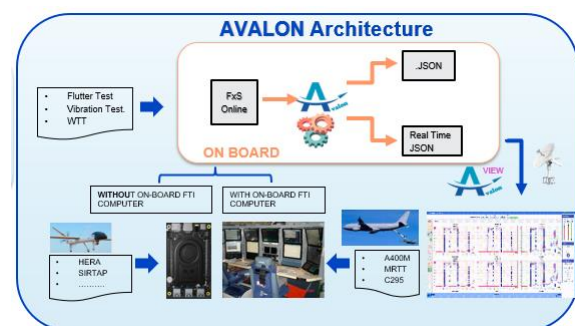


Fig 5 AVALON On-board

On the ground station, AVALONVIEW accesses these real-time JSON files and displays the results, providing valuable insights for our analysis.

Improvements in AVALON

Since the first version of AVALON, numerous improvements have been made in terms of speed and precision, and new calculations have also been implemented.

- Improvements in the calculation accuracy:
The new version allows dividing the frequency analysis range into several sub-

ranges, which makes the modal calculations more accurate.

- Reduction in computation time for Power Spectral Densities (PSDs):

The PSD calculations for each parameter are executed once across the entire frequency range, from lowest to highest. The implementation in C++ enables parallel processing using multithreading, significantly improving efficiency.

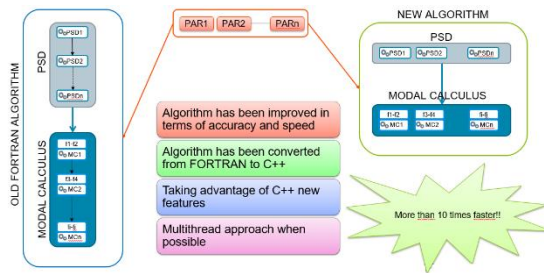


Fig 6. AVALON Improvements

- Reduction in computation time for modal analysis.

Once the PSDs have been calculated, the modal analysis process is parallelized for each frequency range.

Additionally, the algorithm's speed has been enhanced using Forsythe orthogonal polynomials and GAUS method, which will be discussed in a later section.

- With all these implementations, the computational speed of AVALON has improved by factor of 15 compared to its previous version.

Launcher AVALON interface

A launcher AVALON interface tool has been developed. The interface is divided into distinct blocks:

- Configuration Block:** This section allows the user to select a JSON configuration file that contains the analysis parameters and characteristics.
- Special Parameters Block:** This block is dedicated to auxiliary parameters such as speed, altitude, Mach number, etc. Each text box can be used to input the name of the parameter or, if the parameter is not available, a numerical value.

- Execution Block:** Here, the user can choose the execution mode: real-time, deferred, or access to a CSV file.
- Output Block:** In this section, the user can specify the name of the JSON results file. Additionally, a CSV file is automatically generated for visualization in Spotfire.

This modular design ensures a user-friendly and efficient workflow for executing AVALON with customizable parameters and output options.

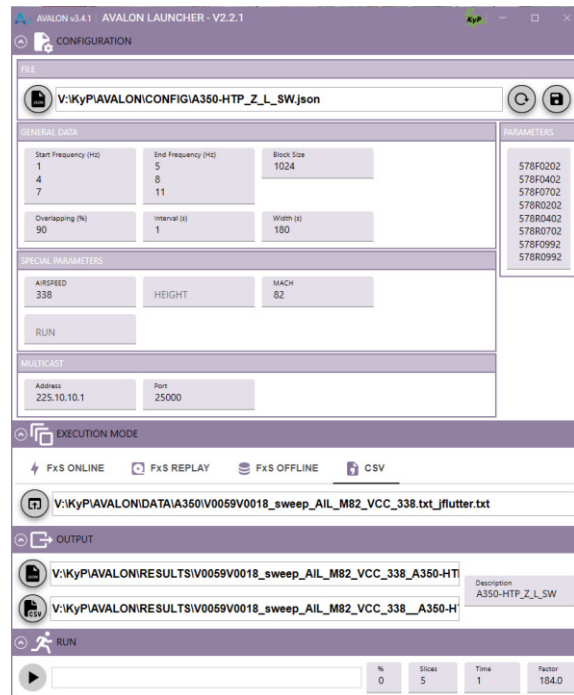


Fig 7. launch AVALON interface

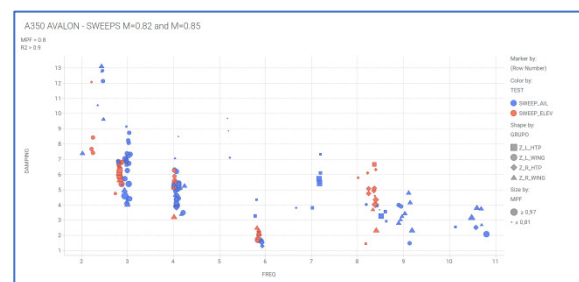


Fig 8. AVALON results represented in Spotfire

AVALON algorithm

The concepts of the AVALON algorithm can be found in Ref. [1]. The most important steps are summarized below:

Given a system, the relationship between the unknown inputs $x(t)$ and the measured response $y(t)$ can be expressed as:

$$G_{yy}(i\omega) = H(i\omega)^* G_{xx}(i\omega) H(i\omega)^T \quad (1)$$

Where:

- G_{yy} is the PSD matrix of responses.
- G_{xx} is the PSD of inputs.
- H is the transfer function.

The TF can be written in partial fraction with the poles and residue form:

$$H(i\omega) = \sum_{k=1}^n \frac{R_k}{i\omega - \lambda_k} + \frac{R_k^*}{i\omega - \lambda_k^*} \quad (2)$$

Where n is the number of poles, λ_k is the pole and R_k is the residue.

Suppose the input is a white noise so its PSD is a constant matrix $G_{xx} = C$ Ref[2].

Then:

$$G_{yy}(i\omega) = \sum_{k=1}^n \left[\frac{R_k}{i\omega - \lambda_k} + \frac{R_k^*}{i\omega - \lambda_k^*} \right] C \sum_{s=1}^n \left[\frac{R_s}{i\omega - \lambda_s} + \frac{R_s^*}{i\omega - \lambda_s^*} \right]^H \quad (3)$$

After some mathematical operation, the output PSD can be reduced to pole residue form as follow:

$$G_{yy}(i\omega) = \sum_{k=1}^n \frac{A_k}{i\omega - \lambda_k} + \frac{A_k^*}{i\omega - \lambda_k^*} + \frac{B_k}{-i\omega - \lambda_k} + \frac{B_k^*}{-i\omega - \lambda_k^*} \quad (4)$$

The least common denominator of the equation (4) is:

$$(i\omega - \lambda_k)(i\omega - \lambda_k^*)(-i\omega - \lambda_k)(-i\omega - \lambda_k^*) = (\omega^2 + \lambda_k^2)(\omega^2 + \lambda_k^{*2}) \quad (5)$$

Then, the relation between the poles of $G_{yy}(j\omega)$ and $H(j\omega)$ is:

$$\lambda_k^{system} = \sqrt{-\lambda_k}$$

The identification algorithm implemented in the program is based on the Rational Fraction Polynomial method (RFP).

The PSD of the outputs is fitted by a fraction where the numerator is a polynomial matrix and the denominator is a polynomial.

$$G_l(s2) = \frac{\sum_{k=0}^m a_{l,k} s2^k}{\sum_{k=0}^{2n} b_k s2^k} \quad (6)$$

$l = 1, \dots, N_f$

Where:

$$s2 = s^2, \quad s = i\omega_r = i2\pi f_r, \quad (r = 1, \dots, n_p) \quad \omega_r \in [-1, 1]$$

As previously mentioned, one of the improvements in the AVALON algorithm involves the use of Forsythe orthogonal polynomials to obtain the coefficients $a_{j,k}$ and

b_k . Therefore, the generation of these polynomials and their relationship to the initial coefficients $a_{j,k}$ and b_k will be examined.

Forsythe orthogonal polynomials

- Generation of Forsythe orthogonal polynomial

$$P_{-1}(x) = 0 \quad (7)$$

$$P_0(x) = 1$$

$$P_1(x) = xP_0(x) - U_1P_0(x)$$

$$P_2(x) = xP_1(x) - U_2P_1(x) - V_1P_0(x)$$

$$\dots\dots\dots$$

$$P_m(x) = xP_{m-1}(x) - U_mP_{m-1}(x) - V_{m-1}P_{m-2}(x)$$

Where:

$$U_m = \frac{\sum_{l=1}^N x_l |P_{m-1}(x_l)|^2 q_l^2}{D_{m-1}}$$

$$V_m = \frac{\sum_{l=1}^N x_l P_{m-1}(x_l) P_{m-2}(x_l) q_l^2}{D_{m-2}}$$

$$D_m = \sum_{l=1}^N x_l |P_m(x_l)|^2 q_l^2$$

and the orthogonally conditions are verified:

$$\sum_{l=1}^N P_m(x_l) P_n(x_l) q_l^2 = \begin{cases} 0 & m \neq n \\ b_{mn} & m = n \end{cases} \quad (8)$$

- Relation between the coefficients of the polynomials and the coefficients of the orthogonal polynomials

Suppose a polynomial of degree k :

$$Q_k(x) = \sum_{i=0}^k a_i P_i(x) = \sum_{i=0}^k c_i x^i \quad (9)$$

The relation between the coefficients is:

$$c_r = \sum_{m=r}^k a_m b_{r,m}$$

Where:

$$b_{r,m} = \begin{cases} 0 & r < 0 \\ 0 & r > m \\ 1 & m = 0 = r \\ b_{r-1,m-1} - U_m b_{r,m-1} - V_{m-1} b_{r,m-2} & 0 \leq r \leq m \end{cases} \quad (10)$$

Computation of the coefficients $a_{j,k}$ and b_k

Instead of computing the coefficients $a_{j,k}$ and b_k directly, the coefficients c_k and d_k are determined by minimizing the following expression:

$$G_l(s2) = \frac{\sum_{k=0}^m c_k \phi_{l,k}}{\sum_{k=0}^{2n} d_k \theta_{l,k}} \quad (11)$$

Where:

- $\phi_{l,k}$ value of polynomial of degree k at $s2_l$
- $\theta_{l,k}$ value of polynomial of degree k at $s2_l$

The weighting function used for the numerator and denominator orthogonal polynomial will be:

- $q_l^2 = 1$ for polynomial $\phi_{l,k}$
- $q_l^2 = G_l^2$ for polynomial $\theta_{l,k}$

Therefore, due to the orthogonal conditions:

$$\sum_{l=1}^N \phi_{l,k} \phi_{l,j} = \begin{cases} 0 & k \neq j \\ \phi_k & k = j \end{cases} \quad (12)$$

$$\sum_{l=1}^N \theta_{l,k} G_l^2 \theta_{l,j} = \begin{cases} 0 & k \neq j \\ \theta_k & k = j \end{cases}$$

The algorithm consists in obtain the matrix polynomial c_k and the coefficients d_k that minimize the expression:

$$e_l = \sum_{k=0}^m c_k \phi_{l,k} - G_l (\sum_{k=0}^{2n-1} d_k \theta_{l,k} + \theta_{l,2n}) \quad (13)$$

Expressed in matrix equation:

$$\{e\} = [P]\{c\} - [T]\{d\} - \{\omega\} \quad (14)$$

Where:

$$[P] = \begin{pmatrix} \phi_{1,0} & \cdots & \phi_{1,m} \\ \vdots & \ddots & \vdots \\ \phi_{L,0} & \cdots & \phi_{L,m} \end{pmatrix} \quad (15)$$

$$[T] = \begin{pmatrix} G_1 \theta_{1,0} & \cdots & G_1 \theta_{1,2n-1} \\ \vdots & \ddots & \vdots \\ G_L \theta_{L,0} & \cdots & G_L \theta_{L,2n-1} \end{pmatrix} \quad (16)$$

$$\{\omega\} = \begin{pmatrix} G_1 \theta_{1,2n} \\ \vdots \\ G_L \theta_{L,2n} \end{pmatrix} \quad (17)$$

Being N_f the number of responses, the expression to minimize is:

$$E = \sum_{j=1}^{N_f} \{e_j\}^T \{e_j\} \quad (18)$$

Then the system to be solved is: $A^T A X = A^T B$

$$\begin{pmatrix} [DP] & [0] & \cdots & [0] & [X_1] \\ [0] & [DP] & \cdots & [0] & [X_2] \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ [0] & [0] & \cdots & [DP] & [X_{N_f}] \\ [X_1]^T & [X_2]^T & \cdots & [X_{N_f}]^T & [DT] \end{pmatrix} \begin{pmatrix} \{c_1\} \\ \{c_2\} \\ \vdots \\ \{c_{N_f}\} \\ \{d\} \end{pmatrix} = \begin{pmatrix} \{h_1\} \\ \{h_2\} \\ \vdots \\ \{h_{N_f}\} \\ \{C\} \end{pmatrix} \quad (19)$$

Where:

$$[DP] = [P]^T [P] \quad [DT] = \sum_{j=1}^{N_f} [T_j]^T [T_j] \quad [X_j] = -[P]^T [T_j] \{h_j\} = [P]^T \{\omega_j\} \quad \{C\} = -\sum_{j=1}^{N_f} [T_j]^T \{\omega_j\} = \{0\}$$

Due to the orthogonal conditions, [DP] is a diagonal matrix and $\{C\} = \{0\}$

These coefficients are obtained by applying GAUS method.

$$\{c_j\} = [D_P]^{-1} (\{h_j\} - [X_j]\{b\}) \quad (20)$$

$$\{d\} = ([D_T]^R)^{-1} \{h_0\}^R \quad (21)$$

Where:

$$[D_T]^R = [D_T] - \sum_{j=1}^{N_f} [X_j]^T [D_P]^{-1} [X_j] \quad (22)$$

$$\{h_0\}^R = -\sum_{j=1}^{N_f} [X_j]^T [D_P]^{-1} \{h_j\} \quad (23)$$

Once the coefficients c_j and d , have been identified, $a_{j,k}$ and b_k are obtained using equation (10).

To improve the conditioning of the system, all responses are normalized by their maximum value, and frequencies are mapped to the symmetric interval $[-1,1]$.

Poles estimation

The poles of the system will be the square root of the zeros of the denominator's polynomial.

$$\sum_{k=0}^{2n} b_k s^k = 0 \quad (24)$$

The algorithm is executed for different orders of the denominator and numerator and keeps the best result according to the coefficient of determination R2 between the PSDs and the synthesized one.

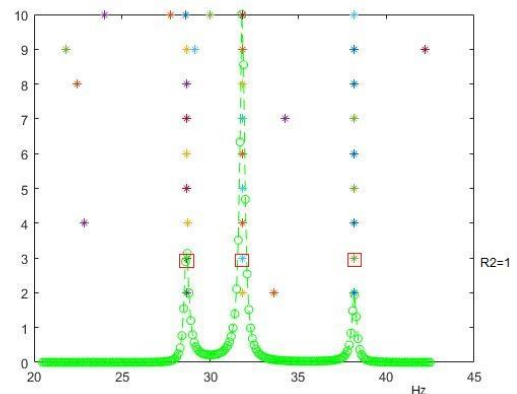


Fig 9. Automated order selection using R2

Residues calculations from Output-Only System Responses

For a linear time-invariant (LTI) system, the transfer function can be written in modal form:

$$H(i\omega) = \sum_{k=1}^n \frac{R_k}{i\omega - \lambda_k} + \frac{R_k^*}{i\omega - \lambda_k^*} \quad (25)$$

Where:

- $\lambda_k = -\zeta_k + i\omega_k$ is a complex pole.
- $R_k = |R_k|e^{i\phi_k}$ is the complex residue.
- ϕ_k contains the modal phase information.

The residue encodes the spatial modal shape and its phase.

- Cross Spectral between two responses:

Suppose we measure two outputs (x,y), of the same system:

The Cross-Spectrum is:

$$S_{xy}(\omega) = X(\omega) * Y(\omega) \quad (26)$$

If the system is excited by a broadband.

$$S_{xy}(\omega) \propto H_x(\omega) * H_y^*(\omega) \quad (27)$$

So, the Cross-spectrum is proportional to the product of the two transfer functions.

- Behavior near a single dominant pole:
Near a resonance frequency ω_k :

$$H_x(\omega) \approx \frac{R_{xk}}{i\omega - \lambda_k} \quad H_y(\omega) \approx \frac{R_{yk}}{i\omega - \lambda_k} \quad (28)$$

Thus:

$$S_{xy}(\omega) \propto \frac{R_{xk}R_{yk}^*}{(i\omega - \lambda_k)(-i\omega - \lambda_k^*)} \quad (29)$$

The denominator is real and strictly positive:

$$(i\omega - \lambda_k)(-i\omega - \lambda_k^*) = |i\omega - \lambda_k|^2 \quad (30)$$

Therefore, the phase of the cross-spectrum comes entirely from: $R_{xk}R_{yk}^*$

So:

$$\angle S_{xy}(\omega_k) = \angle R_{xk} - \angle R_{yk} \quad (31)$$

This is the relative phase of the residues. Therefore, the relative phase with respect to one of the responses of the residue is the minimum value of the phase of the cross-spectrum around the pole.

Validation

The algorithm validation has been done with theoretical data and with a corresponding to a real flutter test.

In the first case, a theoretical transfer function was defined, and the poles were identified without any errors.

In the second case, the results were compared with the jFlutter Tool.

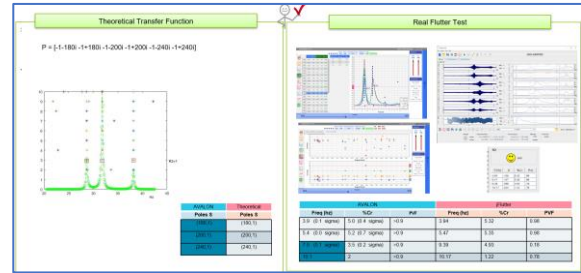


Fig 10. AVALON Validation

AVALONVIEW new features

This section highlights some of the new features in AVALONVIEW

- The new AVALON version enables analysis segmentation into different frequency ranges; however, AVALONVIEW provides an integrated view of the results.

The following figure represents an analysis divided into the frequency ranges [2-8] HZ and [7-12], whereas AVALONVIEW shows the unified analysis.

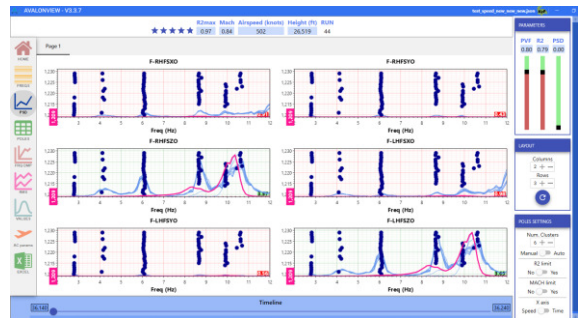


Figure 11. PSD integrated view

- Range analysis selection in addition to zooming in in the RMS section.

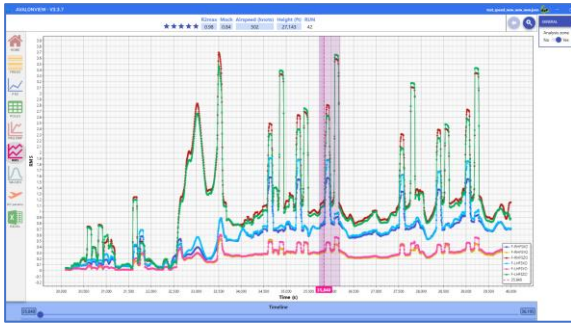


Fig 12. Analysis time slice selection in RMS section

- Mach number, altitude and speed parameters have been added to the poles interface



Figure 13. Mach, SPD and Height

Conclusions

The AVALON framework is designed to function as a real-time monitoring object, operating in seamless conjunction with the jFlutter environment. A key capability of AVALON VIEW is its ability to perform continuous tracking of vibration modes from the onset of flight. By employing advanced continuous tracking methodologies, the system ensures the uninterrupted monitoring of structural dynamics throughout the entire flight envelope.

Beyond in-flight operations, AVALON provides automated solutions for ground-based vibration testing. This includes analysis during taxiing maneuvers, Ground Vibration Tests (GVT), and other pre-flight procedures. The system performs automated characterization of the aircraft's normal modes by analysing the structural flexible response during various ground phases, such as take-off, landing, and runway taxiing.

The deployment architecture of AVALON is scalable, depending on the specific requirements of the platform and the available computational resources:

- Microcomputer-based Deployment: For small-scale platforms and UAVs—such as

HERA, SIRTAP, EURODRONE, and EFA—AVALON is implemented on microcomputing platforms (e.g., Latterpanda), providing high-performance analysis within a compact hardware footprint.

- Flight Test Instrumentation (FTI) Deployment: For larger aircraft, including the A400M, C295, and MRTT fleets, AVALON is integrated into dedicated Flight Test Instrumentation (FTI) consoles, ensuring robust processing and data management for full-scale flight test campaigns.

References

- [1] Francisca Coll Herrero, Pedro Rubio Alvarez. "AVALON: Automated Vibration and Oscillation Nodes". ETTC 2025
- [2] K.SoaI,C. Thiem,T.Meier. "Embedded Flight Vibration Testing System for online Flutter Monitoring of UAVS. IFASD 2022
- [3] Sun Wei "Output Only modal analysis – Scaled mode shape by adding small masses on the structure ", Department of Mechanical Engineering. Blekinge Institute of Technology. Karlskrona, Sweden 2008