Detection of turbulence during flutter tests

P. Vacher, A. Bucharles
ONERA, Department of Systems Control and Flight Dynamics,
2, av. Edouard Belin, B.P. 4025, 31500 TOULOUSE, FRANCE
e-mail: **Pierre.Vacher@onera.fr**, **Alain.Bucharles@onera.fr**

Abstract

Flutter tests are a crucial phase of the flight test program of a new aircraft. One of the main goals is to prove the absence of unstable aeroelastic modes throughout the flight domain. This is achieved by applying calibrated excitations to the aircraft structure. Undisturbed measurements are required for this operation. Hence turbulent conditions are avoided as much as possible.
In the framework of a research program in collaboration with Airbus, a toolbox was developed by ONERA for processing flight test data. In this article, we present the tool that was designed to detect the occurrence of turbulence gusts during flight tests even when excitations are applied to the aircraft. This paper details the structure and the implementation of this turbulence detector. It also describes an innovative and efficient formulation of an adaptive bandstop filter.

1 Introduction

Flutter tests represent a major stage in the certification procedure of a new aircraft. The objectives are to demonstrate the absence of unstable mode throughout the flight envelop and to check the compliance of the aircraft actual behaviour with predicted aeroelastic models. This is achieved by applying calibrated excitations to the aircraft structure for a set of predefined flight conditions that covers the whole flight envelop. The structural modal parameters are estimated from the measurements of the aircraft responses. For the time being, these tests require turbulence free data in order to obtain accurate estimates of the modal parameters.

For sine-weep excitations that last about 2 minutes, the risk is that turbulence gusts occur in the course of the test and induce a deterioration of the quality of the data which might render the test useless. In such a situation, the test has sometimes to be performed again which impinges the scheduling of flight tests and, of course, increases the cost of the flight program.

In the framework of a research program called MEFAS in cooperation with Airbus in the years 1999–2001, a library of tools listed in table 1 was developed by ONERA for processing flight test data. Most of them are devoted to the identification of the aircraft modal parameters [1]. They have been used for testing the four latest aircraft of Airbus company.

In this article, we describe the tool named the “turbulence detector” that was developed for revealing the occurrence of turbulence during the flight tests. It is designed to be efficient during excited flight phases when the sine-weep excitation is applied to the aircraft. The goal is to provide the operator with a real-time indicator of the level of turbulence affecting the test in progress so that, if needed, he can decide to stop the test and wait for a quieter period to restart the test during the same flight. The indicator also allows him to monitor the lessening of turbulence to determine the appropriate time for relaunching the test. This improved handling of test conditions might result in a better quality of the flight data and also in an eased organization of the flight planning.

This paper is organized as followed. The first section is devoted to the presentation of the context and the requirements for the turbulence detector. The second section describes the main lines of the solution that

was developed. The following one provides a detailed description of the components that compose of the turbulence detector. The last section concerns the evaluation of the tool on real flight test data.

Function name	Operation performed
`IdTrans, IdPulse, IdBrut`	Identification procedures [1]
`polish`	Numerator refining
`visu, visuglo`	Visualization of identification results
`PtiRonds`	Monitoring the order determination [1]
`PaveBleu`	Analysis of the identified model [1]
`Izdatgoud`	Analysis of transfer estimation quality
`Kalifes`	Identification quality by mode groups
`Zefic Zefdec`	**In-operation turbulence detection**
`Oclair`	Modal pairing between flight cases
`cocktail`	Modal spatial filtering
`Adream`	Time-frequency analysis and filtering

Table 1: Tools developed in the MEFAS project

2 Context and specifications

2.1 Definitions

Before specifying the problem to be solved, it is essential to define the difference between *aerologic turbulence* and *aerodynamic noise*.

In specific flight conditions especially at low and high speeds, the measurements are affected by a permanent background noise due to the aerodynamic flow around the aircraft. The data processing techniques have to make do with this disturbance in spite of a less favorable signal over noise ratio. The level of this noise is not constant as it evolves with the flight conditions (speed, Mach number, ...).

What we try to detect is the *aerologic turbulence* which is caused by disturbances in the atmosphere the aircraft is flying through. This turbulence occurs sporadically as wind gusts and areas of air agitation are encountered by the aircraft.

This turbulence will evidently excite the structure of the aircraft. For the identification techniques currently used for flutter tests, this might lead to erroneous estimation of the modal parameters especially the damping ratios. Hence the objective is to detect the occurrence of air turbulence in the frequency band considered for flutter analysis. This obviously constitutes a daring challenge as the sine-sweep solicitations administered to the aircraft cover the same frequency range.

2.2 Requirements for the turbulence detector

This turbulence detection tool will be used in the telemetry center to monitor and manage the progression of flight tests. Several requirements follow from this operational context:

- the detection must be performed in real time,
- it must rapid, a delay of a few seconds being allowed
- it must be reliable so that the ongoing test would not be stopped inappropriately
- it must adapt to various aerodynamic noise conditions
- no a priori knowledge on the aircraft aeroelastic model can be taken into account

The detection concerns two types of flight situations:

- unexcited phases and more especially the stabilized phase previous to a flutter test
- sine-sweep tests over a specified frequency bandwidth

Pulse excitations are also used for flutter analysis. But, as their duration is much shorter (about 10 seconds), the likelihood of turbulence is smaller. The impact of turbulence is also less critical because less accuracy is sought for the identified modal parameters for those tests. For these reasons, it was not required that the detector operates for this type of tests.

2.3 Orientation of our development

At Airbus, this development was a first attempt to build an automatic turbulence detector. Therefore, the tool presented in this article should be considered as a first step to test the feasibility and the efficiency of turbulence detection rather than a definitive solution.

The solution proposed here is only based on a *signal processing* approach. The objective is to isolate the contribution of turbulence on *each* measurement. A result is then provided individually for each sensor on the aircraft.

It is expected that the operational evaluation of this procedure together with physical considerations about the aircraft structural behaviour and the nature of aerologic turbulence will enable us to develop in the future a more integrated tool based on the automatic merging of the information from the most appropriate sensors on the aircraft.

3 General presentation of the turbulence detector

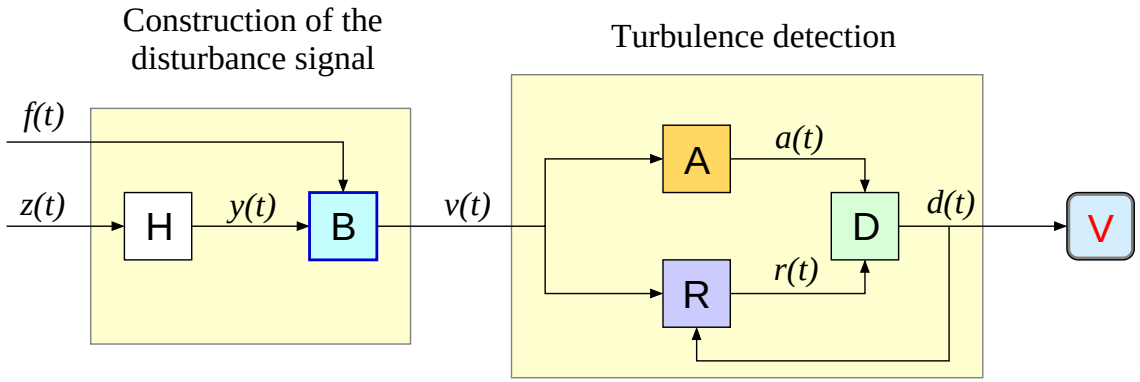
3.1 Overall organization

Let denote $z(t)$ a raw measurement issued by one of the accelerometers on the aircraft structure. The information in this signal comes from several origins:

1. the response of the airplane to piloting commands
2. the response of the structure to the sweep excitation
3. various measurement errors due to the measurement device and associated processing.
4. the background aerodynamic noise
5. the response of the aircraft to possible turbulence gusts

The objective is to detect the single contribution of the turbulence. To achieve this goal, we will use features that characterize these various contributions on $z(t)$:

- Piloting actions on the aircraft only affect the low frequencies.
- Concerning the behaviour of the aircraft under solicitation, we can suppose that the instantaneous frequency bandwidth of its response is in the neighbourhood of the current frequency of the sweep excitation.
- The measurement errors can be classified into systematic errors (bias, calibration inaccuracies, ...) and random errors which can be assimilated to a noise.
- No specific knowledge is available for aerodynamic noise except that it evolves with the flight conditions
- The aerologic turbulence generally occurs in gusts that can last a few minutes. It solicits the aircraft structure in the similar way to the excitation signal but over a larger bandwidth.



Description of the signals

- $z(t)$: Accelerometer raw measurement
- $f(t)$: Sweep instantaneous frequency
- $y(t)$: Accelerometric agitation
- $v(t)$: Disturbance signal
- $a(t)$: Mean amplitude of disturbances
- $r(t)$: Reference amplitude for ambient noise
- $d(t)$: Turbulence indicator

Description of the blocks

- H : Highpass filter
- B : Adaptive bandstop filter
- A : Averaging the signal amplitude
- R : Update of the reference amplitude
- D : Computation of turbulence indicator
- V : Turbulence display interface

Figure 1: General organization of the turbulence detector

Based on these facts, we devised the detection procedure described in figure 1. It comprises two main steps. The first one consists in elaborating what is called the "disturbance signal" $v(t)$. This signal would ideally only include the disturbances on the measurement $z(t)$, i.e. the measurement and aerodynamic noises and the turbulence. Then this signal is processed in a second phase to generate the "turbulence indicator" $d(t)$ which quantifies the likelihood of turbulence.

To obtain the disturbance signal $v(t)$, we first eliminate the contribution of the piloting actions as well as the systematic errors on the measurement. This is accomplished by the highpass filter H. Its output $y(t)$ is named the "accelerometric agitation".

The following block B aims at discarding the response of the aircraft structure to the sine-sweep excitation. It is an adaptive bandstop filter with a center frequency f_c tuned to the current value $f(t)$ of the sine-sweep signal. This filter attenuates the signal in a specified bandwidth b about f_c. Of course, some of the turbulence contribution will also be discarded. But, as mentioned above, we expect that, instantaneously, the turbulence will excite the aircraft on a larger bandwidth than the sine-sweep input. Therefore, by carefully selecting the value of b, the disturbance signal $v(t)$ retains a sufficient amount of the energy due to the turbulence.

In the second stage, the turbulence is detected by comparison with a reference signal $r(t)$ which reflects the intensity of the noise in the absence of turbulence. Several solutions were investigated to build the quantity that would be the most relevant of the presence of turbulence in the signal $v(t)$. In order to produce a reliable detection, some kind of averaging is also necessary. A local mean energy of the signal $v(t)$ proved to be the most appropriate quantity. The signal $a(t)$ computed in the block A of the diagram 1 is homogeneous to a signal amplitude. It corresponds to the square root of the mean signal energy over a sliding time window of duration T_{win} according to

$$a(t) = \sqrt{\frac{1}{T_{\text{win}}} \int_{t-T_{\text{win}}}^{t} v^2(\tau)\, d\tau} \tag{1}$$

The initial value of the reference signal is defined in the same way by considering a window of a larger duration T_{ini} at the beginning of the test.

$$r(T_{\text{ini}}) = \sqrt{\frac{1}{T_{\text{ini}}} \int_0^{T_{\text{ini}}} v^2(\tau)\, d\tau} \tag{2}$$

Of course, it is essential that no turbulence occurs during this initialization phase. The value of $r(t)$ is subsequently updated in order to refine this initial guess and to adapt to the evolution of noise conditions during the flight. This is achieved by the lowpass filter R with a large time constant. This update operation is modulated by the detected turbulence level $d(t)$ so that the reference amplitude does not increase unduly when the disturbance signal $v(t)$ is affected by turbulence.

The detection is simply achieved by analyzing the ratio $q(t) = a(t)/r(t)$. At this stage, it is not easy nor desirable to produce a binary result limited to two answers: turbulence or no turbulence. From an ergonomic point of view, it is more convenient and pleasant for the operator to be able to sense the onset of turbulence which often appears gradually rather than to switch abruptly from a quiet to a turbulent state. From a detection point of view, it is also difficult to devise a reliable detection test on the values of $q(t)$ because a trade-off between false and missed probabilities of detection has to be made.

Therefore, the output $d(t)$ of the turbulence detector is a degree of likelihood of the turbulence. Based on first experiments with this method, we can define two thresholds $q_{\min}$ and $q_{\max}$. Above $q_{\max}$, we can affirm almost surely the presence of turbulence. Similarly, when $q(t)$ is lower than $q_{\min}$, the level of turbulence can be considered as negligible. Thus the turbulence indicator signal $d(t)$ delivered by the detector takes its values between 0 and 1 according to

$$d(t) = \begin{cases} 1 & \text{if} \quad q(t) \geq q_{\max} \\ 0 & \text{if} \quad q(t) \leq q_{\min} \\ \dfrac{q(t) - q_{\min}}{q_{\max} - q_{\min}} & \text{otherwise} \end{cases} \tag{3}$$

The last point concerns the presentation of this result to the operator. Instead of a conventional plot of the values of $d(t)$, we proposed a color display evolving from the green to red as d(t) increases from 0 to 1. The colormap associated to the values of $d(t)$ is depicted in figure 2. Such an interface is more intuitive and more appropriate to attract the operator's attention in the atmosphere of flight tests which is sometimes very tense. Because of their appearance as colored stripes, these interfaces are dubbed "Bayadère[1] stripes".

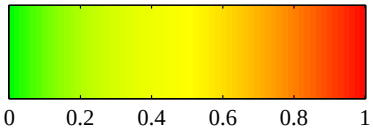


Figure 2: Colormap for Bayadère stripes

3.2 Implementation of the turbulence detector

The turbulence detection tool was implemented in MATLAB®. Extensive use was made of the routines of the Signal Processing toolbox [2]. The implementation was developed with two main objectives in mind:

- the flexibility to use this tool in various configurations
- the efficiency of the real-time processing

[1] Bayadère: a design of brightly colored stripes, mostly produced in India. The name is derived from the Bayadère dance.

The term “flexibility” means that the turbulence tool is designed to process efficiently several measurements in parallel with *similar or different settings*. This feature enables the operators to adapt the settings to the physical characteristics of each measurement and to the test being processed. In the development phase, it also allows him to run concurrently several settings to find the most performant configuration. This tool must also sufficiently fast to cope with the measurement sampling period of 128 Hz.

The detection tool is composed of two MATLAB® routines with the following calling syntax[2]:

```
 zefpar                    = Zefic(fsamp, tuning)         ;
[turbind, zefpar, intsig] = Zefdec(meas, fexcit, zefpar) ;
```

The routine `Zefic` is an initialization procedure that is executed off-line before launching the real-time function `Zefdec`. It performs the following operations:

- assign all the tuning parameters
- compute preliminary quantities in order to save time for the real-time routine `Zefdec`

The first argument `fsamp` is the sampling frequency. The second `tuning` is optional. It is a MATLAB® structure that enables to modify all the settings which are defined by default values. The function output `zefpar` gathers all the parameters necessary to perform the real-time detection.

The routine `Zefdec` carries out recursively all the operations described in figure 1. It is able to process several measurements in parallel making use of the efficient matrix operations in MATLAB®. Hence, at each sampling time t_k, it is feeded with the current values of a vector `meas` of n_z measurements $z(t_k)$ that will be processed in parallel and with the value `fexcit` of the frequency of the excitation signal $f(t_k)$. The turbulence indicator $d(t_k)$ is computed in the vector `turbind` for each measurement in `meas`. The variable `zefpar` is updated. The internal signals described in figure 1 are accessible in the structure `intsig`.

This routine `Zefdec` comprises 3 successive time phases :

- the *first measurement* processing which essentially consists in initializing all the internal arrays.
- the *initialization phase* where the initial value of the reference noise level $r(t)$ is computed. Its duration is T_{ini} (equation 2).
- the *active phase* where the whole detection scheme is operating.

The structure of these two routines thus offers wide possibilities of use. Several turbulence detectors with different settings can be run concurrently by computing several sets of parameters `zefpar` with the function `Zefic`. At each sampling time, the function `Zefdec` is then called several times with these different variables `zefpar`. Of course, each call to `Zefdec` can also process a bunch of several measurements.

4 Detailed description of the turbulence detector

In this section we detail the boxes of the diagram 1 by describing the method used for signal processing and the implementation that was adopted.

4.1 Highpass filter

In this application, we are not much concerned by the signal distortion but mainly by the computational load. So we opted for an elliptic Cauer filter [3, 2, 4, 5] because the filter attenuation specifications can be met with the lowest possible order.

The filter is specified by the passband and stopband edge frequencies f_p and f_s and the associated attenuation R_p and R_s. The default values for these parameters were determined to attenuate the aircraft rigid dynamic

[2]The word “zef” is French slang for “wind”. The suffix “ic” stands for “initial conditions” and “dec” for “detection”.

and to preserve its aeroelastic response while trying at the same time to minimize the order of the filter. We adopted the following default values of the filter which produces a filter of order 3.

$$f_s = 0.3 \text{ Hz} \qquad R_s = 40 \text{ dB}$$
$$f_p = 1.0 \text{ Hz} \qquad R_p = 0.2 \text{ dB}$$

The numerator and denominator of the filter are computed in the routine `Zefic`. The function `ellipord` of the MATLAB® signal processing toolbox is used to determine the lowest order of the filter that meets the specifications. The filter coefficients are computed by the function `ellip`. In the routine `Zefdec`, the measurements are processed by the function `filter`.

4.2 Adaptive bandstop filter

The block B in the diagram 1 is one of the crucial pieces of the turbulence detector. Hence it deserves a longer description.

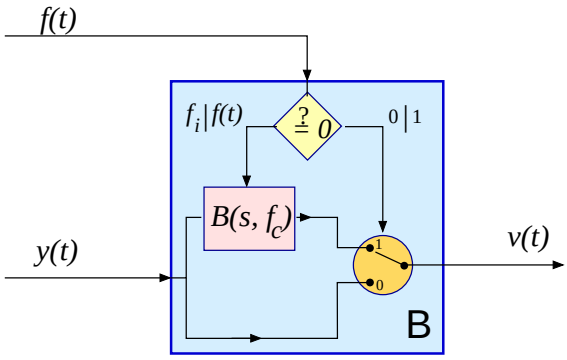


Figure 3: Description of the block B

This block is organized as depicted in figure 3. The sub-block $B(s, f_c)$ defines the actual adaptive bandstop filter with a central frequency equal to f_c. When the excitation frequency is equal to 0 which, by convention, means that the sine-sweep excitation is not active, the output $v(t)$ is switched directly to the agitation signal $y(t)$. Nevertheless the bandstop filter $B(s, f_c)$ is running and processing the data with the value f_c equal to the initial frequency f_i of the sine-sweep. Although the output of the filter is not used, the purpose is to eliminate the transient phase when the sine-sweep starts. At this moment, the output is switched to the filter output and the central frequency f_c of the filter B is constantly updated by the current sweep frequency $f(t)$. There is also a time period at the end of the sweep when the filter keeps on processing the data with f_c equal to the final frequency of the sweep. This time lag enables the structural modes in the neighbourhood of this final frequency to damp down before switching back to the signal $y(t)$.

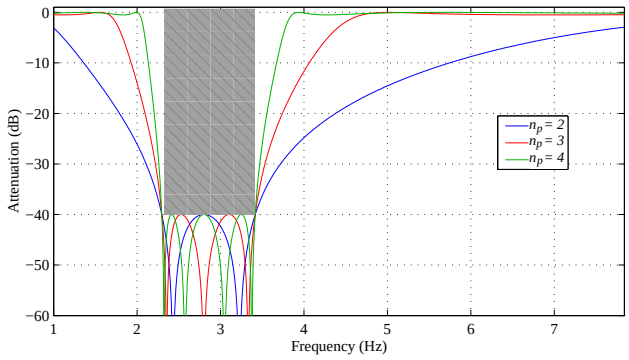


Figure 4: Specifications of the bandstop filter

Similarly to the highpass filter, we used an elliptic formulation for the design of $B(s, f_c)$ in order to get an efficient attenuation with an order as low as possible. Instead of the usual specifications, we found it more convenient to specify the filter by its order n_d and the specifications of its *stopband* (central frequency f_c, bandwidth[3] b, attenuation R_s). The passband is only specified by its equiripple level R_p. As shown in figure 4 which depicts the stopband specifications and the filter frequency responses for several values of $n_p = n_d/2$, the choice of the order enables to tune the trade-off between the filter selectivity and the computational load with the same attenuation in the stopband.

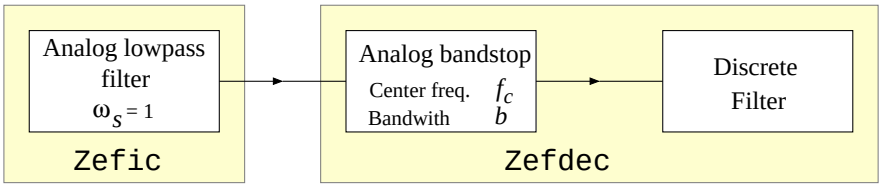


Figure 5: Design steps of the bandstop filter

For a given central frequency f_c, the filter is designed according to the analog prototyping procedure [3, 2, 4]. The steps of this procedure are depicted in figure 5:

1. an analog lowpass filter is designed with a stopband *edge pulsation* ω_s equal to 1.
2. the filter is transformed into an analog bandstop filter with desired bandwidth b and center frequency f_c.
3. the analog filter is converted to a digital filter.

Our approach slightly differs from the conventional procedure[4] because the prototype filter in the first step is usually computed for a *cut-off pulsation* ω_n equal to 1 instead of ω_s. The purpose of this modification is to preserve the characteristics of the stopband while the order of the filter is changed as depicted in figure 4.

This computation of the analog lowpass prototype can be carried out off-line in the `Zefic` function. In the second step f_c takes the values of the current sweep frequency $f(t)$. Hence the last two steps needs to be performed at each sampling time in the `Zefdec` function. Thus we need to find out an efficient way of implementing these transformations.

As bandstop filters generally involve high order filter (twice the order of the analog lowpass prototype), a state-space representation is a favourable filter description to avoid numerical problems encountered with high order polynomials. Let $(A_p,\ B_p,\ C_p,\ D_p)$ designate the state-space representation of the prototype analog lowpass filter of order n_p.

The transformation into an analog bandstop filter with a central pulsation ω_c and a relative bandwidth b is described by the following relations [2] where $(A_a,\ B_a,\ C_a,\ D_a)$ is a state-space representation of the bandstop filter and I_p the identity matrix of dimension n_p with $n_p = n_d\,/\,2$.

$$\begin{aligned} A_a &= \begin{bmatrix} \omega_c\, b\, A_p^{-1} & \omega_c\, I_p \\ -\omega_c\, I_p & 0 \end{bmatrix} & B_a &= \begin{bmatrix} -\omega_c\, b\, A_p^{-1}\, B_p \\ 0 \end{bmatrix} \\ C_a &= \begin{bmatrix} C_p\, A_p^{-1} & 0 \end{bmatrix} & D_a &= D_p - C_p\, A_p^{-1}\, B_p \end{aligned} \tag{4}$$

The computation of the associated digital filter is accomplished by the bilinear Tustin transformation which corresponds to the following operations [2] where I_d is the identity matrix of dimension n_d

$$\begin{aligned} A_d &= (I_d - A_a\,/\,\kappa)^{-1}\,(I_d + A_a\,/\,\kappa) & B_d &= \sqrt{\frac{2}{\kappa}}\,(I_d - A_a\,/\,\kappa)^{-1}\,B_a \\ C_d &= C_a\,(I_d - A_a\,/\,\kappa)^{-1}\,\sqrt{\frac{2}{\kappa}} & D_d &= D_a + \frac{1}{\kappa}\,C_a\,(I_d - A_a\,/\,\kappa)^{-1}\,B_a \end{aligned} \tag{5}$$

[3] The bandwidth designates here the width of the *stopband*.

[4] The MATLAB® function `ellipap` which computes the analog lowpass filter prototype needs to be modified.

The quantity κ is the frequency warping constant which is used to match the central frequencies of the analog and digital filter. It is equal to

$$\kappa = \frac{\omega_c}{\tan \dfrac{\omega_c \, \Delta t}{2}} \qquad \text{with} \qquad \omega_c = 2 \, \pi \, f_c \tag{6}$$

where Δt is the sampling period

The equations 4 and 5 involves two matrix inversions of size n_p and n_d . We developed a much more efficient implementation by merging these two operations into a single one which is mathematically equivalent and which requires about the same amount of computation as the first transformation 4.

Let define an intermediate system $(A_i,\ B_i,\ C_i,\ D_i)$ by

$$\begin{aligned} A_i &= b \, A_p^{-1} \, / \, 2 & \qquad B_i &= -b \, A_p^{-1} \, B_p \, / \, 2 \\ C_i &= C_p \, A_p^{-1} & \qquad D_i &= D_p - C_p \, A_p^{-1} \, B_p \end{aligned} \tag{7}$$

We also need the following quantities

$$\begin{aligned} t_c &= \tan \, \pi \, f_c \, \Delta t & \qquad s_c &= \sin \, 2 \, \pi \, f_c \Delta t \\ F &= I_p / s_c - A_i & \qquad G &= F^{-1} \end{aligned} \tag{8}$$

The relations 4 and 5 can then be computed by

$$\begin{aligned} A_d &= \begin{bmatrix} G \, / \, t_c - I_p & G \\ -G & I_p - t_c \, G \end{bmatrix} & \qquad B_d &= \begin{bmatrix} G \, B_i \\ -t_c \, G \, B_i \end{bmatrix} \\ C_d &= \begin{bmatrix} C_i \, G \, / \, t_c & C_i \, G \end{bmatrix} & \qquad D_d &= D_i + C_i \, G \, B_i \end{aligned} \tag{9}$$

As the intermediate system 7 is independent of f_c, it can be computed beforehand in the routine `Zefic`. Hence in the real-time routine `Zefdec`, only one matrix inversion of dimension n_p is performed when computing the matrix G in 8.

But, instead of forming the state matrices 9 of the filter, an even more efficient implementation can be worked out by directly updating the state and the output of the bandstop filter. Let X_k denote the filter state vector before the update at the time sample t_k. This vector can be partitioned into two sub-vectors $X_{1,\,k}$ and $X_{2,\,k}$ of dimension n_p. It can easily be shown that the filter output v_k and its updated state X_{k+1} can be computed from the input y_k and the state X_k according to the Naussac formulation

$$\begin{aligned} X_{1,\,k+1} &= -X_{1,\,k} + \quad \Delta X_k \\ X_{2,\,k+1} &= \quad X_{2,\,k} - t_c \, \Delta X_k \\ v_k &= \quad C_i \, \Delta X_k + D_i \, y_k \end{aligned} \qquad \text{with} \qquad \Delta X_k = F \setminus \left(\frac{X_{1,\,k}}{t_c} + X_{2,\,k} + B_i \, y_k \right) \tag{10}$$

We adopted the following default settings for $B(s, f_c)$ which are discussed in subsection 5.1:

Attenuation in the stopband	:	-40 dB
Equiripple in the passband	:	0.5 dB
Relative bandwidth	:	0.4
Prototype filter order	:	3

4.3 Averaging the signal amplitude

In the discrete-time domain, the signal $a(t_k)$ described in equation 1 can readily be computed by

$$a_k = a(t_k) = \sqrt{\frac{1}{n_{\text{win}}} \sum_{l=0}^{n_{\text{win}}-1} v_{k-l}^2} \qquad \text{with} \qquad v_{k-l} = v(t_{k-l}) \quad \text{and} \quad T_{\text{win}} = n_{\text{win}}\, \Delta t \tag{11}$$

In spite of its simplicity, the computation of a_k by this formula proved to be expensive in computer time. Actually, at each sampling time, two operations are to be performed:

- storing the last n_{win} values v_l
- computing the amplitude a_k

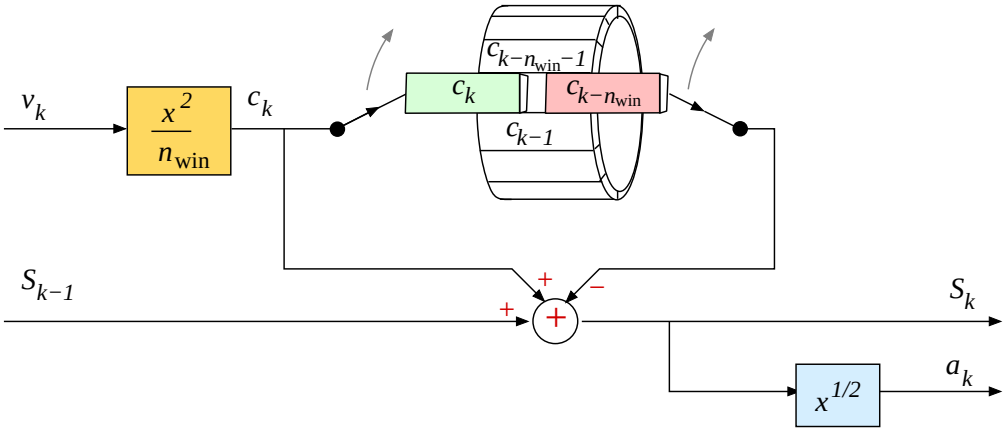


Figure 6: Computation of the averaged signal amplitude

The figure 6 illustrates the much more efficient approach we implemented. From the disturbance signal v_k, we define the quantity $c_k = v_k^2/n_{\text{win}}$. The n_{win} last values of this variable c_k are stored in an array. In order to minimize the flow of data in the computer memory, we used the concept of circular buffer depicted in 6: the new values of c_k associated with the n_z measurements replace the values $c_{k-n_{\text{win}}}$ in the *same* memory location.

The average values a_k are computed by a recursion which involves less computation than the summation in 11. Let define S_k by

$$S_k = \frac{1}{n_{\text{win}}} \sum_{l=0}^{n_{\text{win}}-1} v_{k-l}^2 = \sum_{l=0}^{n_{\text{win}}-1} c_{k-l} \tag{12}$$

Then we obviously have

$$S_k = S_{k-1} + c_k - c_{k-n_{\text{win}}} \qquad \text{and} \qquad a_k = \sqrt{S_k} \tag{13}$$

For the initialization phase, i.e. when $k \leq n_{\text{win}}$, the recursion on S_k simplifies to

$$S_k = S_{k-1} + c_k \tag{14}$$

The default width T_{win} was chosen equal to 2 seconds to be in accordance with the specification about the acceptable delay for turbulence detection.

4.4 Reference amplitude update

The initial value $r(T_{\text{ini}})$ of the reference amplitude $r(t)$ defined in equation 2 is computed by a procedure similar to 14 but over a greater number of n_{ini} samples. A default value of 5 seconds was retained for T_{ini}.

The signal $r(t)$ is subsequently updated by a lowpass filter with a long time constant to adapt to the evolution of the noise condition. Actually, the update is carried out on the averaged energy defined as $e(t) = r(t)^2$ by a first order filter. This operation is also modulated by the value of the detection signal $d(t)$. Then we have

$$e_k = e_{k-1} + K_r \, (1 - d_{k-1}) \, (v_k^2 - e_{k-1}) \tag{15}$$

As the detection signal d_{k-1} takes its values between 0 and 1, no update occurs when some turbulence is detected. The filter gain K_r is taken equal to 3 Δt / T_{rep} where T_{rep} is the filter 95 % settingly time. The default value for T_{rep} is set to 4 minutes.

4.5 Turbulence indicator

The value of the detection is directly obtained from the formula 3. For the thresholds $q_{\min}$ and $q_{\max}$, a first evaluation on real flight data led us to retain the default values 1.1 and 2.

4.6 Turbulence display interface

The perception of the color by the human eye does not depend linearly upon the intensity of the colors in the RGB code. The colormap adopted for the Bayadère stripes (figure 2) varies from green to yellow when $d(t)$ increases from 0 to 0.5 and from yellow to red for $0.5 \leq d(t) \leq 1$. The increase of the red intensity in the first range for $d(t)$ and the decrease of green intensity in the second range are computed by the following function where $i(x)$ is the color intensity.

$$i(x) = \frac{\arctan(\alpha \, x)}{\arctan(\alpha)} \qquad \text{with} \qquad 0 \leq x \leq 1 \tag{16}$$

For the coefficient α, the respective default values 1 and 3 were retained. The associated variation of intensity are depicted in figure 7.

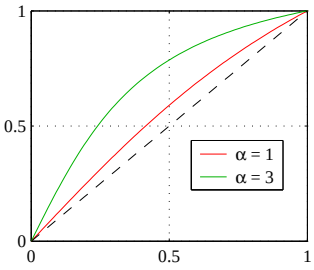


Figure 7: Evolution of color intensity

5 Evaluation on real flight data

The set of default settings described in the previous section was used for this evaluation. These values will be refined by the experts at Airbus by processing numerous flight test data. They can probably adapted more specifically to each measurement by taking into account physical considerations on the influence of turbulence on the aircraft structure.

In this section, we first discuss the critical tuning parameters of the detector. Then we present its performance to detect turbulence on real flight data. The last point concerns the computational load which establishes how many measurements can be processed in real-time.

5.1 Critical tuning parameters

The essential parameters for this filter are

- the specifications of the stopband and the order of the filter $B(s, f_c)$ in B
- the length T_{win} of the averaging window in A
- the two thresholds $q_{\min}$ and $q_{\max}$ in D

For the bandstop filter described in subsection 4.2, an attenuation of -40 dB prove to be sufficient to remove the contribution of the excitation for all the data processed. An order of 6 for $B(s, f_c)$ seems to be a nice trade-off between the computational load, the filter selectivity and its responsiveness (about 3 s). The only point that would require is a deeper analysis is the filter bandwidth b. We adopted a relative bandwidth with a default value of 0.4 but an absolute bandwidth might as well be considered.

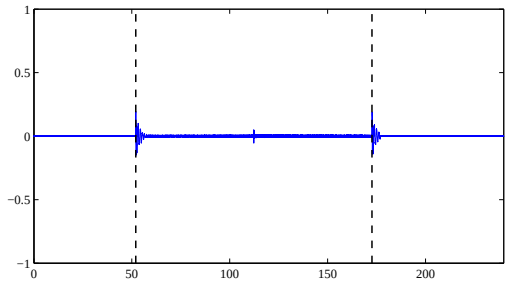


Figure 8: Filtered excitation signal

To demonstrate the efficiency of this adaptive filter, a double sine-sweep signal[5] was processed. Figure 8 depicts the filtered excitation signal. The interval where the sine-sweep is applied, is delimited by the dashed lines. Knowing that the sweep amplitude is equal to 1, it can be checked that the attenuation is equal to 40 dB except for the short transition phases (about 3 seconds) at each end of the excited phase and, to a tiny extent, at the sweep maximum frequency. It must however be noted that these transitions are sharply marked for the excitation signal. Such abrupt changes do not occur on measurements.

The length T_{win} in the block A influences the smoothing on the disturbance signal $v(t)$ and the responsiveness of the turbulence detector. The default value 2 s was chosen as a compromise between these two considerations.

Finally the thresholds $q_{\min}$ and $q_{\max}$ have a direct impact on the output of the turbulence detector. They also condition the update of the reference signal. The default values were determined by the analysis of real flight data.

5.2 Detection performance for unexcited flight phases

This flight is affected by two periods of turbulent conditions as depicted in figure 9. The first and longer turbulent phase begins approximatively at time 20 s. It increases rather gradually and strengthens till 65 s before settling down at time 80 s. It is followed by a short quiet phase until about 100 s from when turbulence

[5]The frequency of the sweep increases from f_i to f_m then decreases back to f_i.

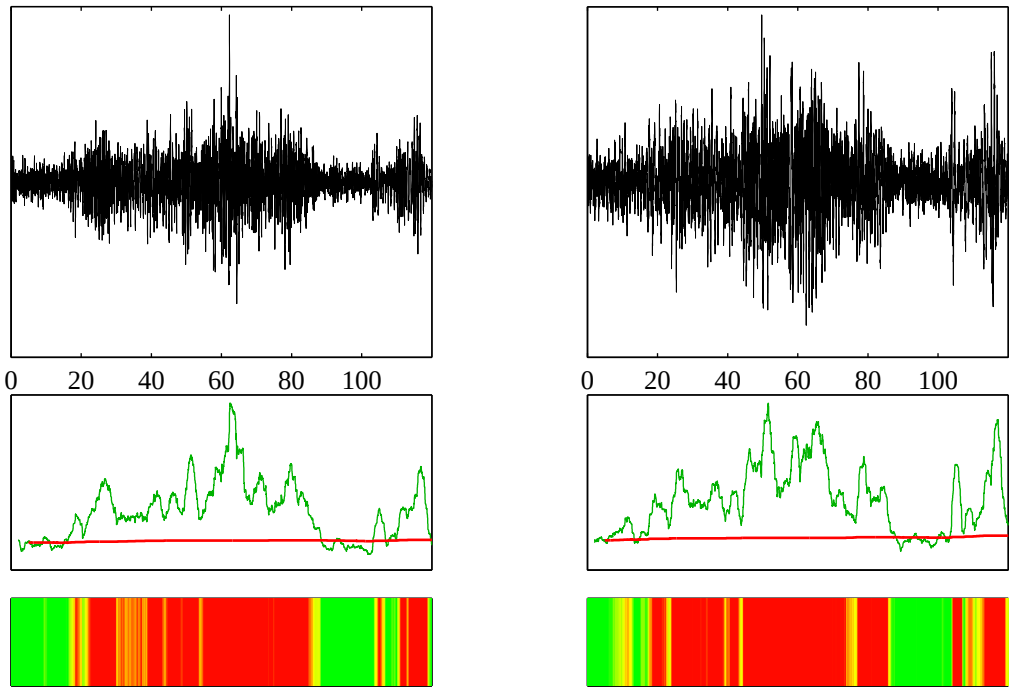


Figure 9: Turbulence detection during an unexcited flight test

persists till the end of the flight. Two sensors were considered for this analysis. The second one appears more perturbed by the background noise.

In figure 9, the upper graphic represents the accelerometric agitation $y(t)$ (see figure 1) associated with these two measurements. Below, the green line is the averaged disturbance amplitude $a(t)$ while the red line represents the reference amplitude $r(t)$. Finally, at the bottom of the figure, the Bayadère stripes computed with the signal $d(t)$ are drawn.

The long patches of red color on these stripes evidently demonstrate the capability of our tool to detect the turbulence. They also reveal that the second measurement is more sensitive to turbulence. But what is also remarkable is the ability of the detector to return to a green state after a long turbulent period. This is due to the freezing mechanism of the update of the reference amplitude $r(t)$ during turbulent conditions. The red lines in figure 9 show that $r(t)$ remains constant during these periods.

5.3 Detection performance during sine-sweep excitations

The performance of the turbulence detector were also tested on a double sine-sweep test similar to the one described in subsection 5.1. Two measurements were selected along the same axis as the application of the excitation on the aircraft structure. This situation is more severe for the detector because the contribution of the aircraft response to the excitation is more important.

It is difficult from only the visual analysis of the accelerometric agitation $y(t)$ in figure 10 to distinguish the effect of turbulence from the response of the aircraft when the excitation is active. This phase is delimited by vertical dash lines in the figure.

The outputs of the adaptive bandstop filter $v(t)$ which appear in yellow in the figure reveal the presence of a gust during the second part of the excitation. The contribution of turbulence is much more important on the second measurement as revealed by the Bayadère stripes. The results on this test demonstrate several facts:

- When no turbulence occurs, the adaptive filter is efficient to cancel out the contribution of the excitation.

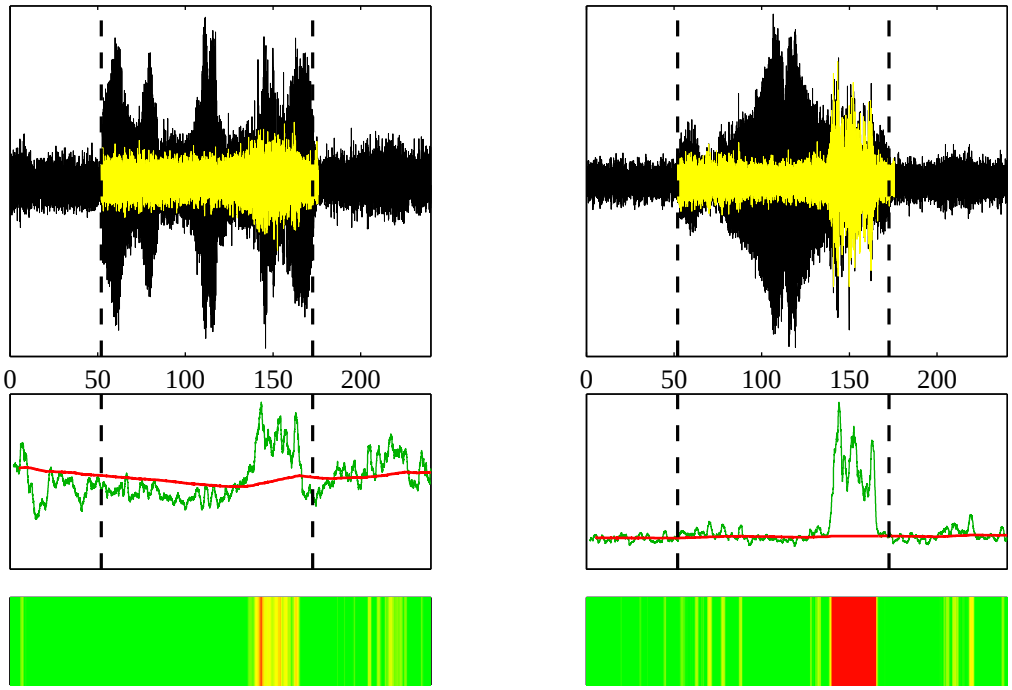


Figure 10: Turbulence detection during a sine-sweep excitation.

- The detector is able to distinguish the turbulence from the aircraft response to sine-sweep excitations.
- Some measurements are more sensitive than others to turbulence.
- The update procedure of the reference amplitude $r(t)$ is also quite appropriate as shown by the first measurement in figure 10.

5.4 Computational load

MATLAB® version	1 meas.	20 meas.
6.0 R12	22.6	27.4
7.2 R2006a	27.8	36.9

Table 2: Turbulence detector execution times (without graphics)

The detector was implemented in MATLAB® although this programming environment is not favorable to rapid recursive computations. Table 2 gives the execution times on a SUN® SunBlade 1500 with a 1.1 GHz clock rate for a test which lasts 240 s. It reveals that though the durations are longer with the more recent versions of the MATLAB®, this tool can easily operate in real-time. It has the capability to process several measurements with the same settings or between 8 and 10 measurements with different settings. Table 3 reveals that the adaptive filter is the most time consuming block of the detector which justifies the careful implementation we adopted.

Detector block	Computation ratio
H	9 %
B	30 %
A	13 %
R	7 %

Table 3: Distribution of the computational load (MATLAB® version 6)

6 Conclusion

In this article, we presented a turbulence detector able of isolating the contribution of turbulence on each measurement even in the presence of a sine-sweep excitation applied to the aircraft.

The initial goal of this development was to provide the operator with a tool that would help him to decide to pursue or to stop the test in progress. Conversely, in the case of moderate turbulence levels, this tool could also be used for the selection of the measurements for the identification by indicating those which are less affected by turbulence.

Our next objective will be to develop a global indicator of the turbulence by merging the result of appropriately selected measurements on the aircraft structure.

Acknowledgements

Pierre Vacher feels very grateful to several colleagues at ONERA : Alain Piquereau who proposed the use of an energy function to detect the turbulence, Jean-Luc Boiffier who suggested the denomination “Bayadère stripes” for the detector interface and Philippe Mouyon for his advice on adaptive filtering. The authors also thank Anne Pin-Belloc, Cécile Daudet, Jean Roubertier, Frédéric Dessillons at Airbus for their constructive and cordial cooperation during the MEFAS project.

References

[1] Pierre Vacher and Alain Bucharles. A multi-sensor parametric identification procedure in the frequency domain for the real-time surveillance of flutter. In *14th IFAC Symposium on System Identification (SYSID 2006)*, pages 636–641. International Federation of Automatic Control, March 2006.

[2] The MathWorks. *Signal Processing Toolbox: User’s Guide*, March 2006.

[3] Marc Labarrère, Jean-Pierre Krief, and Bernard Gimonet. *Le filtrage et ses applications*. Cepadues – Editions, 1978.

[4] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Signal Processing Series. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 07632, 1989.

[5] T. W. Parks and C. S. Burrus. *Digital Filter Design*. Topics in Digital Signal Processing. John Wiley & Sons, Inc., 1987.