Proceedings of COBEM 2011
Copyright © 2011 by ABCM

21st Brazilian Congress of Mechanical Engineering
October 24-28, 2011, Natal, RN, Brazil

A FOUR-SIGNAL VOTING ALGORITHM FOR AIRCRAFT REDUNDANT SENSORS

Lívia Camargos Rodrigues de Oliveira, livia.oliveira@anac.gov.br
Roberto Kawakami Harrop Galvão, kawakami@ita.br
[1]Agência Nacional de Aviação Civil (ANAC), Av. Cassiano Ricardo, 521, Bloco B, 3º andar, CEP 12.246-870 - São José dos Campos – SP, Brasil.
[2]Instituto Tecnológico de Aeronáutica (ITA) – Divisão de Engenharia Eletrônica, Praça Marechal Eduardo Gomes, 50, CEP 12.228-900 - São José dos Campos – SP, Brasil.

Abstract. *Aircraft are equipped with multiple sensors that measure variables such as speed and angle of attack, as well as the pilots' commands. In general, each variable is measured by several redundant sensors considering the possibility of sensor fault and the severity of the erroneous measurement or its loss. All measured values of the same variable must be consolidated into a single value by a voting logic. This information can be used for indication to the pilots in the cockpit displays and/or to calculate commands to deflect the aircraft control surfaces. Consequently, if the voter provides an erroneous value of a given variable to the avionics and control systems, the pilots may be induced to take wrong decisions and the commands calculated and applied to the control surfaces will be incorrect, which may cause catastrophic accidents. Therefore, the voter must be able to detect sensors faults and to consolidate a single and reliable value for each variable, thus avoiding the propagation of faults to the avionics and control systems. The purpose of this work is to propose a voting algorithm for four signals to be applied to aircraft redundant sensors. This work also includes simulations of fault cases and verification of the proposed algorithm response in these situations.*

Keywords: *aircraft, redundancy, voting algorithm, fault detection, fault tolerance.*

1. INTRODUCTION

Aircraft are equipped with multiple sensors that measure variables such as speed, angle of attack, as well as the pilots' commands. In general, each variable is measured by several redundant sensors due to the possibility of sensor fault and to the severity of the erroneous measurement or its loss. Another reason for the use of redundancy is the increase in aircraft dispatchability, since it is allowed to take off with inoperative sensors, as established by the Minimum Equipment List (MEL), if it has more sensors than necessary for flight safety. All values measured for the same variable must be consolidated into a single value by a voting logic. This information can be used for indication to the pilots in the cockpit displays and/or by the control laws to calculate the commands to deflect the aircraft control surfaces. Consequently, if the voter provides an erroneous value of a given variable to the avionics system, the pilots may be induced to take wrong decisions and an incident or accident might occur. In a similar way, if the value of a specific variable supplied to the control laws is incorrect, the commands calculated and applied to the control surfaces will be incorrect, which has a great potential to cause catastrophic accidents. Therefore, the voter must be able to manage the measured data in such a way to detect sensors faults and to consolidate a single and reliable value for each variable, thus avoiding the propagation of faults to the avionics and control systems.

According to Newman *et al.* (2009), the main types of accident related to the Fly-By-Wire (FBW) technology are: uncommanded pitch or roll, abrupt maneuver and Pilot-Induced Oscillations (PIO). Still in conformity to this reference, design errors concerning fault detection and isolation constitute a relevant contributor for the events related to uncommanded pitch or roll.

The incident with an A-320 in August 9th, 2001 during the approach in Heathrow Airport (AAIB, 2001), the incident faced by a B-777 in August 1st, 2005 while climbing over the Indian Ocean (ATSB, 2007), the accident with an A-330 cruising over the Indian Ocean in October 7th, 2008 (ATSB, 2009) and the catastrophic accident with an A-330 that was flying over the Atlantic Ocean in June 1st, 2009 (BEA, 2010) are a few examples of aeronautical incidents and accidents possibly caused by sensor faults and problems in the detection and isolation of these faults.

Newman et al. (2009) stated that, in order to avoid incidents and accidents caused by uncommanded pitch or roll, it is necessary to make improvements in the fault detection and isolation logics, particularly for the second fault of the same type and combinations of different types of faults.

It is worth noting that the voter does not avoid the occurrence of sensor faults. However, it should be able to detect them and to prevent their propagation to the systems that employ voted signals. Therefore, such systems may be allowed to keep operating with a consolidated value based on the measurements of other redundant sensors. If multiple faults occur in such a way that it is not possible for the voter to consolidate a reliable value of the measured variable (the required reliability being defined by the criticality of each system), a degradation of the mode of operation of the aircraft or loss of some functions may occur, which is still preferable to retaining the systems functions and the mode of operation with incorrect data.

Some works related to voting algorithms have been published in the literature. Latif-Shabgahi *et al.* (2004) propose a functional classification of the voting algorithms employed in real time embedded control systems. Parhami (1992),

(1994) presents another classification for voters and describes several algorithms of the exact, inexact and approval types. Latif-Shabgahi *et al.* (2003) introduce a family of voters based on weighed average suitable for aircraft redundant sensors. Patton (1991) describes analytical methods for fault diagnosis focusing on aircraft sensors monitoring. Lee (1994) proposes a method for validation of the sensor-measured values through causal relations and their relationships. Some patents have been registered in the United States by the Boeing Company (1984), (1981) with respect to signal selection – especially the mid-value selection technique – and fault detection, which are directly related to voters.

In general, the literature presents voting algorithms for systems with double or triple redundancy, or generic algorithms to vote N signals. In this work, these concepts are extended to the quadruple redundancy case. This architecture has been increasingly applied in the aeronautical environment due to the growing importance and criticality of control systems for flight safety.

This works aims at presenting a four-signal voting algorithm oriented to aircraft redundant sensors. It includes the proposed algorithm itself along with the rationales behind the design choices, as well as simulations of fault cases and verification of the algorithm response.

2. PROPOSED ALGORITHM

This section presents the inputs and outputs of the proposed four-signal voting algorithm, a detailed description of each of its modules, as well as a list of the parameters used to configure the algorithm.

Figure 1 illustrates the voting algorithm proposed in this work.

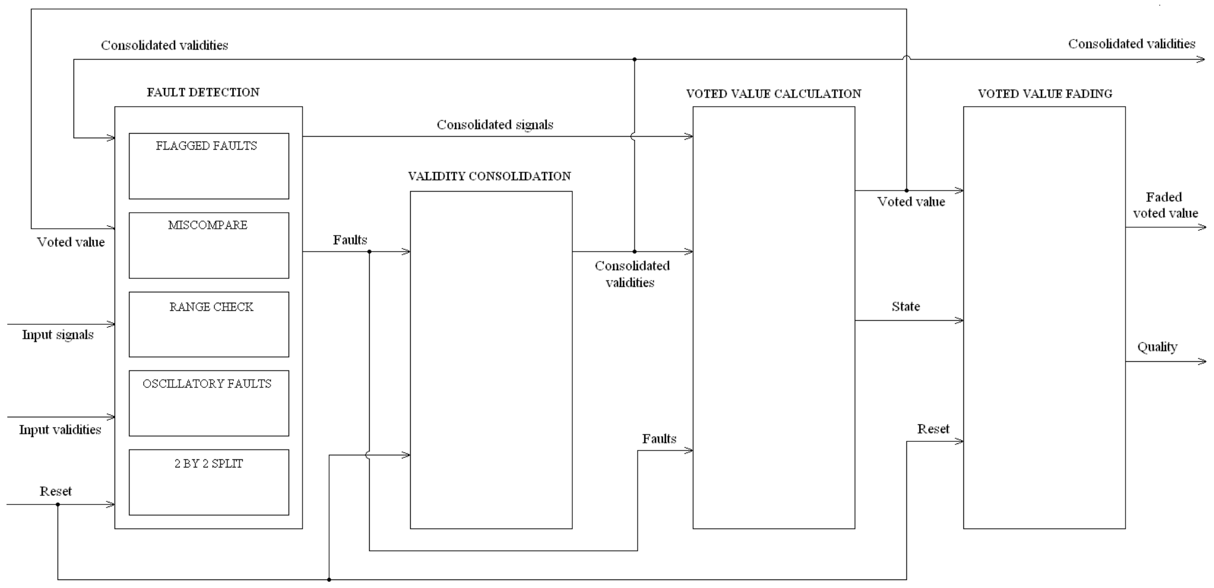

Figure 1. Diagram of the proposed voting algorithm.

The proposed voting algorithm is able to detect some types of faults, as well as to provide some degree of fault isolation and identification by determining which signal is faulty and what is the fault type. In addition, fault accommodation is automatically carried out, as the voted value is calculated on the basis of healthy input signals only.

2.1. Inputs and outputs

The voting algorithm inputs and outputs are presented in Tab. 1 and 2, respectively.

Table 1. Algorithm inputs.

Description	Unit
Signal 1	Same unit as the measured variable
Signal 2	Same unit as the measured variable
Signal 3	Same unit as the measured variable
Signal 4	Same unit as the measured variable
Input validity of signal 1	Dimensionless (Boolean number)
Input validity of signal 2	Dimensionless (Boolean number)
Input validity of signal 3	Dimensionless (Boolean number)
Input validity of signal 4	Dimensionless (Boolean number)
Reset signal	Dimensionless (Boolean number)

Table 2. Algorithm outputs.

Description	Unit
Faded voted value	Same unit as the measured variable
Voted value quality	Dimensionless (Integer between 0 and 4)
Consolidated validity of signal 1	Dimensionless (Boolean number)
Consolidated validity of signal 2	Dimensionless (Boolean number)
Consolidated validity of signal 3	Dimensionless (Boolean number)
Consolidated validity of signal 4	Dimensionless (Boolean number)

2.2. Parameters

The proposed algorithm contains several configurable parameters that allow its application in various situations with reasonable flexibility. Table 3 summarizes the algorithm parameters.

Table 3. Algorithm parameters.

Module	Description	Unit
General	Frequency of the algorithm execution	Hz
Flagged faults detection	Time window size	Execution cycles
	Number of values equal to 1 in the time window	Execution cycles
Range check	Lower limit	Same unit as the measured variable
	Upper limit	Same unit as the measured variable
	Detection persistence time	Seconds
Miscompare detection	Threshold of the instantaneous miscompare	Same unit as the measured variable
	Persistent miscompare threshold ON	Same unit as the measured variable
	Persistent miscompare threshold OFF	Same unit as the measured variable
	Time constant of the 1st order filter	Seconds
Oscillatory faults detection	Smaller frequency of the bandpass filter	rad/s
	Greater frequency of the bandpass filter	rad/s
	Oscillation amplitude threshold	Same unit as the measured variable
	Sliding window size	Seconds
	Detection persistence time	Seconds
Split detection	Smaller threshold	Same unit as the measured variable
	Greater Threshold	Same unit as the measured variable
	Detection persistence time	Seconds
Validity consolidation	Healing time	Seconds
Voted value fading	Transition time	Seconds

The algorithm response may vary significantly when it is configured with different values for the parameters. In general, the parameters' configuration task of a voting algorithm should follow some basic principles:

- A specific analysis for each voted variable should be conducted;
- The fault magnitude should be considered in view of its effect on the system (in this case, the aircraft) and the admissible fault magnitudes, i. e., the impact on the aircraft and pilot caused by a measurement of a certain variable distinct from its real value;
- An appropriate trade-off between false alarms and undetected faults should be achieved;

- Sensors' characteristics, such as precision, resolution, sampling rate, range, among others, should be taken into account;
- Following a preliminary adjustment according to the guidelines above, it is advisable to fine-tune the parameters during the flight test campaign.

2.3. Fault Detection

The fault detection module is composed of sub-modules of range check and miscompare, oscillatory fault and 2 by 2 split detection, which are the unflagged fault types detectable by this algorithm. There is also some processing of flagged faults. The following subsections describe each of these logics.

2.3.1. Flagged faults detection

The input validity of each signal, before being used by the voter to determine the flagged faults, goes through a logic in order to avoid false alarms. For such, two parameters must be provided: the size of a time window and the number of samples of the signal in that interval that must have the value 1 (true) for the validity in the output of this sub-module to be also 1. This is a moving time window, i. e., in each execution frame, this validity is determined by the last consecutive values of the input validity of each signal.

Regardless of this sub-module output validity, if the signal input validity is 0 (false), the value of the corresponding signal in the previous frame is held and the measured value in the current frame is discarded so as to prevent the voter from operating with invalid data.

The Reset signal has no effect over the flagged faults detection.

2.3.2. Miscompare detection

This module aims at detecting divergent signals through comparison. In order to accomplish it, two types of miscompare are used: the instantaneous and the persistent. These calculations are based on the comparison between the value of each signal in the current execution cycle of the algorithm and the voted value in the previous cycle. The instantaneous miscompare verifies the absolute value of this difference instantly and, if it is greater than a given threshold, the divergent signal is removed from the voted value calculation in that execution cycle, although no fault is declared.

The persistent miscompare, on the other hand, gets the instantaneous miscompare and passes its value through a first-order low-pass filter. A fault is declared when the filtered value is greater than an upper threshold. The fault indication is removed when the filtered value gets smaller than a lower threshold. Such hysteresis avoids that faults be declared and removed many times when the filtered value oscillates around the threshold, if there were only one parameter.

The miscompare detection has some peculiarities when there are only two valid signals. In this scenario, as depicted in section 2.5, the voted value is calculated as the average between these signals. There is no meaning in calculating the instantaneous miscompare in this case and removing one of the signals from the voted value computation if it diverges from the voted value in the previous execution cycle. In fact, if there is no abrupt variation in any of these signals, both are equally distant (in absolute value) from the previous voted value, except for small differences arising from numerical computation issues. Therefore, the instantaneous miscompare is disabled when there are only two or less valid signals. The disadvantage of this approach is the propagation of spurious peaks to the voted value in the two valid signals case, although the peak magnitude in the voted value is reduced by a factor of two due to the fact that the voted value is calculated as the average between the two valid signals. On the other hand, the persistent miscompare is still computed for fault detection purposes. However, if a miscompare fault is declared for any of these signals, the remaining one is automatically declared as failed too, since it is not possible to define which of them is correct. It is relevant to note that, regardless of the fact that the voted value is the average between the two signals, the miscompare detection for each of them may trip in distinct execution cycles due to the history accumulated by the filters and to numerical differences. That is why both signals should be discarded in this scenario.

In the case of flagged fault or out-of-range fault, the instantaneous and persistent miscompare detections are disabled for the corresponding signal(s). Disabling the instantaneous miscompare means not declaring fault; for the persistent miscompare, however, it means retaining the filtered value, which avoids that the filter continues to operate with known invalid signals and to accumulate an erroneous history. This fact could cause a wind-up phenomenon leading to a substantial increase in the filtered value. Moreover, disabling the miscompare detection prevents that, when there are only two valid signals and one of them becomes invalid due to flagged fault, the logic discards both signals improperly.

The confirmed or suspect 2 by 2 split fault also disables the miscompare detection. If there is no further information about the specific variables being voted, there is no way to decide which pair is correct. This feature is needed to prevent the voter from declaring miscompare faults for the two signals that constitute the pair that is more distant from the voted value in the previous execution cycle. This could occur more quickly than the confirmation of the 2 by 2 split fault, if it really exists, depending upon the distance between the two pairs of signals, the miscompare threshold and the time for confirming the 2 by 2 split fault. Hence, in order to avoid this undesirable situation, if a 2 by 2 split fault is suspected or confirmed, the persistent miscompare filter is retained in its previous value and the instantaneous miscompare detection is disabled.

If the Reset signal is active, it has priority over this logic of filter output retention. In this case, the filter output becomes null and the instantaneous miscompare detection is disabled. Thus, if there is indication of miscompare – either instantaneous or persistent – for any signal, the activation of the Reset signal removes this indication.

2.3.3. Range check

One of the unflagged fault types detectable by the voter is out-of-range signal. The signal range is defined by two parameters: a lower and an upper limit. This range can be configured as the sensor measurement range or be even more restricted, for instance, according to normal operational conditions or to the excursion of that variable considered in the aircraft design.

If the value of a given signal falls out of the specified range, this fact is immediately perceived by the range check module and causes the signal to be kept in its previous value while this condition persists. If it lasts for a configurable persistence time, which is one of the algorithm parameters, an out-of-range fault is declared for the referred signal.

An out-of-range signal could be declared as failed by the miscompare detection module. However, if the variable range is known, it becomes quicker and easier to detect this type of fault, especially when there are only two valid signals. In this scenario, the miscompare detection logic would consider both signals as invalid because of its inability to define which of them is actually failed. On the other hand, the range check module invalidates only the out-of-range signal.

If the Reset signal is active, the indication of this type of fault is removed for all four signals and the persistence time counter is set to zero.

2.3.4. Oscillatory fault detection

The oscillatory fault detection is needed to avoid that oscillations in the measured signals be propagated to the control laws and even to the aircraft control surfaces. Such propagation could cause mechanical wear and fatigue if the exposure time is large. Additionally, the flying quality could possibly decrease and therefore reduce the comfort perceived by the passengers on board.

The oscillatory fault detection module should discriminate oscillations caused by a fault from actual oscillations in the measured variable. The module comprises three parts:

1. Signal filtering through a second-order bandpass filter;
2. Oscillation detection in the frequency band of interest;
3. Determination of whether or not the detected oscillation corresponds to a fault, based on comparison with the other signals.

The bandpass filter aims at attenuating the signal components in frequencies that fall out of the specified band. The minimum and maximum frequencies of the band are configurable parameters, with unit gain in the frequency that corresponds to the geometric center of the band. The occurrence of oscillations with low frequencies can indeed reveal a fault in the signal; however, it does not affect significantly the control surfaces and, depending on its amplitude, can even be detected by the miscompare detection module. Moreover, this filter removes the offset component of the signal, which makes it easier to detect oscillations through a threshold by the following sub-module of oscillatory fault detection. On the other hand, high oscillation frequencies can also denote a fault; nevertheless, they are already filtered by the control laws, which are band-limited, and so these oscillations do not achieve the aircraft control surfaces. Hence, for either low or high frequencies, it is not desirable to declare fault and then lose one of the redundant sources unnecessarily. The definition of which frequencies are considered low and high is done by configuring the filter parameters.

The oscillation detection takes the filtered signal and verifies, within a given sliding time window, if it surpasses the oscillation detection threshold a number of times that is greater than the one that corresponds to the lower frequency defined for the filter band. When the calculation result is not an integer number, it is rounded to the nearest lower integer in order to assure that an oscillation with the lower frequency of the band will be detected. In case the oscillation amplitude is greater than the threshold, the cycles are counted.

It is important to highlight that the phase lag introduced in the signal by the filter does not affect the oscillation detection, since the time window is sliding and the factors that determine the detection are the oscillation frequency and its amplitude with respect to the threshold. This phase lag could slightly increase the fault detection delay. However, this delay is not significant for the detection of this type of fault, which requires a persistence time of many oscillation cycles. Furthermore, since the filter is not part of any control loop, its phase lag does not affect the performance and stability of the systems that employ the voted value.

Once an oscillation is detected in a given signal, it is necessary to verify whether it reveals a fault or a real oscillation in the measured variable. This is accomplished by comparing the oscillation detection flag for each signal with the corresponding flags for the other signals and by checking their consolidated validities. A fault is declared only if a given signal is valid and oscillates and two or three signals are also valid but do not oscillate. If there are only two valid signals and one of them is oscillating, it is not possible to know if there is a fault in the oscillating signal (in case the measured variable is not oscillating) or in the non-oscillating signal (otherwise). Thus, in this scenario, the oscillating signal is not considered failed. Analogously, if there is only one valid signal and it oscillates, a fault is not declared. In other words: it is only possible to declare an oscillatory fault when there are at least three valid signals among the four and only one of them oscillates.

For a given signal, if the conditions for oscillation fault detection are satisfied, a possible fault is indicated. However, if it is the case that the real variable is oscillating and there is no sensor fault, the oscillation will probably be detected in each signal in distinct execution cycles of the algorithm. In this case, the first signal to be detected as oscillating will indicate a possible oscillatory fault. Before the fault confirmation, during a certain time frame (configurable parameter), the algorithm waits for the other signals to start oscillating as well. If it happens, the indication of possible fault in the first signal to oscillate is removed. On the other hand, if a fault has really occurred, this time frame works as a persistence to declare an oscillatory fault in that signal.

Once an oscillatory fault is detected, it can only be removed if the signal stops oscillating or if the Reset signal becomes active. In the latter situation, it is not verified whether the other signals are oscillating or not.

When a possible oscillatory fault is indicated for any signal, the last value of that signal is kept for the purpose of the voted value calculation. This logic aims at preventing that, before the fault is declared, the oscillation is propagated to the voter output. Nevertheless, during the time interval comprised between the beginning of the oscillation and the possible fault detection, the oscillation may be inevitably propagated. However, the voter output will remain between the amplitude limits imposed by the remaining valid signals, since the voted value is calculated as the median among the valid signals.

2.3.5. Split detection

A 2 by 2 split scenario is identified when two signals agree between them given a small threshold, the other two signals are also close to each other by the same criterion, but the distance between the two pairs of signals is greater than a larger threshold. Additionally, for the detection of a 2 by 2 split scenario, all four signals must be valid considering flagged, out of range, oscillatory and persistent miscompare faults. The rationale behind this logic is that a 2 by 2 split usually results from a generic fault (design error) and it is considered that this type of fault does not occur in combination with other faults. This scenario of inconclusive split between sets of signals can occur when the number of signals to be voted is even. For unordered signals, one way to determine the occurrence of the 2 by 2 split scenario is by checking the following conditions:

1. All four signals have the consolidated validity equal to 1;
2. The absolute value of the difference between each signal and the median of the four signals is greater than or equal to half the greater threshold;
3. The maximum absolute value (regarding all four signals) of the difference mentioned in the previous item does not exceed, by a magnitude greater than the smaller threshold, the minimum absolute value of the same difference.

Item 1 ensures that there is not (or, at least, it has not been detected) any other type of fault in any of the four signals. Item 2 guarantees that each signal is distant from the median by at least half the greater threshold, or, in other words, that the two middle signals are distant from each other by, at least, the value of the greater threshold. Finally, item 3 assures that the extreme signals (the greater and the smaller among all four) are distant from the closer middle signal by, at most, the value of the smaller threshold, i.e., the two smaller signals are close enough to each other to be considered a pair (the same must be true for the two greater signals as well).

The instantaneous detection of the 2 by 2 split scenario generates a suspicion fault flag, which can be confirmed if the scenario remains the same by a given persistence time, specified as an algorithm parameter. Meanwhile, the voted value is kept in its previous value instead of being calculated as the median among the signal values, which is incorrect regardless of which pair has the correct value.

In this module, the Reset signal is used to remove the 2 by 2 split fault indication, which cannot be removed after the healing time. The Reset signal also sets to zero the persistence time counter.

2.4. Validity consolidation

The validity consolidation module receives from the fault detection module information regarding flagged and unflagged faults detected by the voter. From this information combined with a fault retention logic based on a healing time, a consolidated validity is generated for each signal. This validity is assigned the logic value 1 (true) if no fault has been detected for the corresponding signal, except the instantaneous miscompare, and 0 (false) otherwise. Once the consolidated validity of a given signal becomes null, it can only be restored if the healing time has elapsed after all faults detected in the signal are removed by their fault detection module. The healing time, which is a configurable parameter, is useful to avoid that faults be declared and removed in a sequence and the signal be successively discarded and used in the voted value calculation.

The 2 by 2 split fault is a severe fault usually due to design error. Therefore, it cannot be removed even if the scenario changes and remains during the healing time. All the other faults, either flagged or unflagged, are capable of being removed.

The consolidated validities calculated by this module are provided to the voted value calculation module. Additionally, they are stored in order to make maintenance actions easier after the end of the flight.

The Reset signal is used in this module to set the healing time counter to zero. Moreover, since it is also used in the fault detection modules to remove all existing fault indications, it causes the consolidated validities of all signals to be restored. In practice, the Reset signal can be activated before each flight or after maintenance actions that repair detected faults.

2.5. Calculation of the voted value and its quality

The voted value calculation module takes into account the consolidated validity of each signal and computes the voted value from the valid signals, i.e., the ones without any type of flagged or unflagged fault detectable by the voter, including the instantaneous miscompare, which does not affect the consolidated validity of the signal, but removes it from the voted value calculation in that specific execution cycle. The voted value is then computed as the median among all valid signals. If there is no valid signal, the voted value is frozen in its previous execution cycle value.

The median was chosen because it is less susceptible to the influence of outliers when compared to the average.. Indeed, if there are three or four valid signals and one of them shows a peak with amplitude such that the signal outreaches the greater value among the remaining valid signals, the average is affected by the fraction of the peak amplitude relative to the number of valid signals. However, the median remains bounded among the other valid signals' values, regardless of the peak amplitude. For two or just one valid signal(s), the median and the average give the same outcome.

The greater the number of agreeing signals, the greater the probability that the voted value is correct. Thus, the voted value quality in the k-th execution cycle is expressed in terms of the number of redundant valid signals used in its calculation, which can vary between 0 and 4. The user of the voted value then decides, considering its quality, if its integrity is enough for the value to be used for a given purpose. The greater the severity of the effects of an erroneous value, the greater the required integrity. Furthermore, the voted value quality is used to provide the flight crew with the total or partial degradation of a given measurement so that the necessary operational procedures can be taken.

The Reset signal has no direct effect over this module.

2.6. Voted value fading

The voter has 16 states with respect to the input signals and their validities after the fault detection module. This number is the result of the combinatory analysis of the validities of the four signals. It is important to mention that this validity is composed by the consolidated validity combined with the instantaneous miscompare detection.

The voted value fading module acts over the voted value only when the state of the input signals change in order to avoid an abrupt alteration in the algorithm output. A state change occurs if a fault is detected in any valid signal or if a signal is considered valid again after a fault. When one of these scenarios happens, the signals actually used to compute the voted value change. The algorithm then smoothes the voted value in order to have a soft transition from its value in the previous execution cycle to the voted value calculated in the current cycle.

The transition time is an algorithm parameter. The mathematical expression for the proposed fading scheme is given in Eq. (1), where T_s is the sampling time, T_t is the transition time, k_e is the instant when the state changed, w(k) is the faded value and z(k) is the voted value, both at the k-th execution cycle.

$$w(k) = z(k)\left[(k-k_e)\left(\frac{T_s}{T_t}\right)\right] + w(k_e)\left[1-(k-k_e)\left(\frac{T_s}{T_t}\right)\right] \tag{1}$$

This expression is valid for $k_e \le k \le k_e + \frac{T_t}{T_s}$, i.e., from the state change up to the end of the transition time. Before and after the fading, the fader output is the same as the voted value, in other words: w(k) = z(k).

During a transition, if another state change occurs, the fading logic works to make a soft transition of the output from its value in the current execution cycle to the new voted value. Equation (1) is also valid in this case, provided that the value of k_e is updated to the execution cycle in which the new state changed happened.

When there is no state change, there is no fading in the voted value, since the variations are due to variations in the measured variable itself. Hence, in the absence of faults, the fading logic becomes inactive and does not add delays in the voting path. This only occurs during fault transients, which is acceptable.

The Reset signal is used in this module to disable the fading logic so that its output is the same as the voted value.

4. TEST RESULTS

In order to illustrate the algorithm response, a test with synthetic data was developed to assess the miscompare detection logic. The parameters that affect this test were configured with the following values: Frequency of the algorithm execution = 100 Hz; Upper limit of the range check = 10; Instantaneous miscompare threshold = 2; Persistent miscompare thresholds ON and OFF = 1.5 and 1.0, respectively; Time constant of the 1st order filter = 2 seconds; Smaller and greater frequencies of the bandpass filter = 4 rad/s and 49 rad/s; Healing time = 10 seconds.

In this test, the input validities of all four signals were kept at 1, i.e., the occurrence of flagged faults was not considered. Figure 2 shows the time variation of the four input signals applied to the voter.

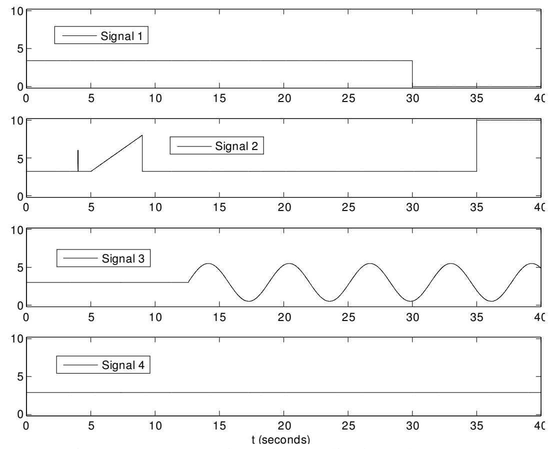

Figure 2. Input signals applied to the voter.

Figure 3 shows the instantaneous and persistent miscompare detection flags for all four signals.

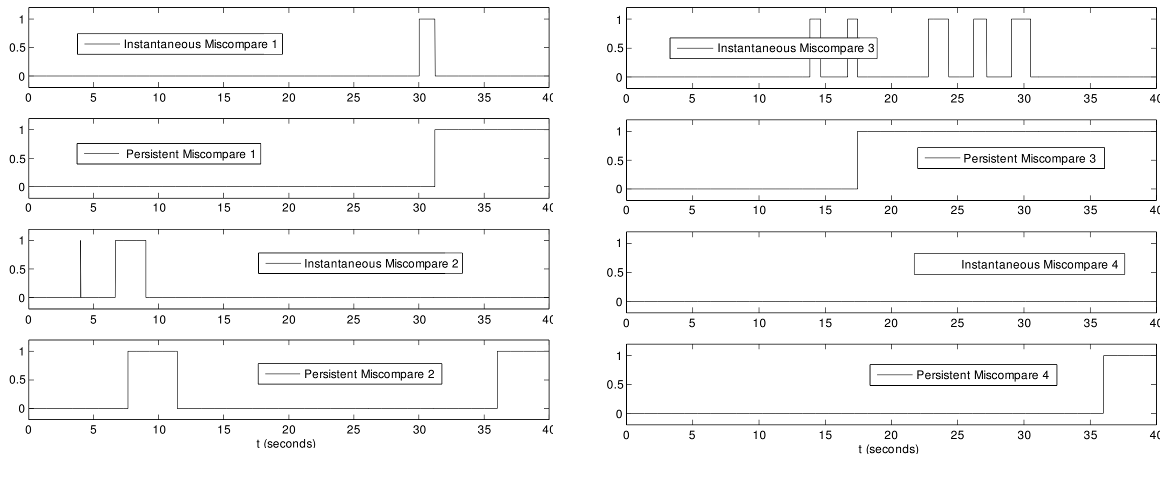

Figure 3. Miscompare detection flags for signals 1 to 4.

The consolidated validities of the four signals, the voted value and its quality are presented in Fig. 4. The faded voted value is not shown to simplify the understanding of this test.

The initial values of the four signals are 3.4, 3.2, 3.0 and 2.9, respectively. Therefore, the voted value, which is calculated as the median among these signals, is 3.1 in the beginning of the test. At t = 4.5 seconds, signal 2 exhibits an spurious peak detected by the instantaneous miscompare flag. As a consequence, this signal is removed from the voted value calculation, although its consolidated validity remains equal to 1. The voted value quality is reduced from 4 to 3 and the voted value changes to 3.0. Due to the short duration of the peak, it does not cause persistent miscompare detection. As soon as signal 2 returns to its previous value, the voted value and its quality recover their previous values.

At t = 5 seconds, a drift fault occurs in signal 2, which reaches the value 8.0 at t = 9 seconds. At the time the absolute difference between this signal and the voted value in the previous execution cycle exceeds the instantaneous miscompare threshold, the signal is removed from the voted value calculation, which happens around t = 6.7 seconds. The persistence miscompare fault is detected at t = 7.6 seconds, thus causing this signal consolidated validity to become 0. Just after the beginning of signal 2 drift, before the instantaneous and persistent miscompare detections, the increase in the value of this signal makes it outreach the values of the other signals in such a way that it does not affect the median anymore. The median then becomes equal to 3.2 (average between 3.4 and 3.0, the two mid values). From the point when the instantaneous miscompare is detected, the voted value becomes 3.0, which corresponds to the mid value between the three valid signals. At t = 9 seconds, signal 2 recovers from this fault and restores its previous value of 3.2; nevertheless, the persistent miscompare indication is removed some time later, around t = 11.5 seconds. The consolidated validity of signal 2 gets back to 1 only after the healing time has elapsed from the moment when the persistent miscompare does not indicate fault anymore.

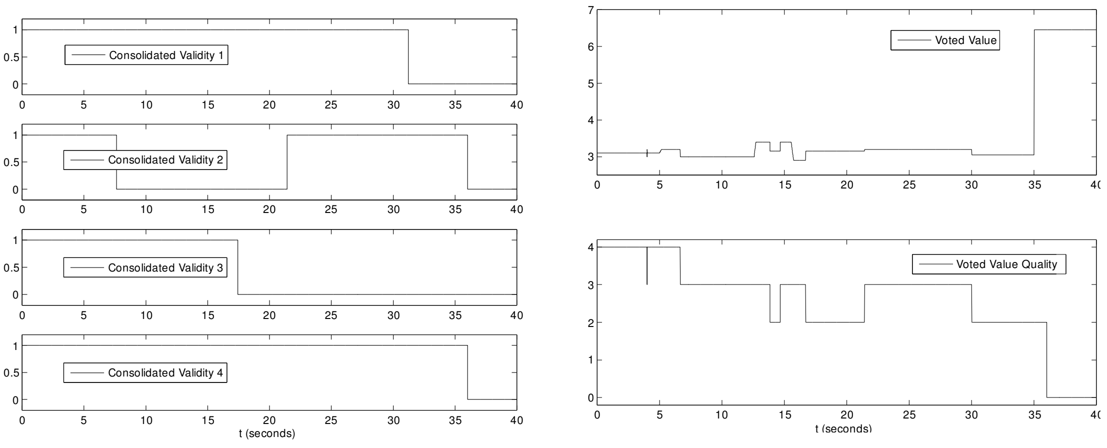

Figure 4. Consolidated validities of the four signals, voted value and its quality.

At t = 12.5 seconds, an oscillatory fault affects signal 3 with an amplitude of 2.5 and a frequency of 1 rad/s. This frequency is located out of the band defined for oscillatory fault detection and hence there is no indication of this type of fault for signal 3. However, due to the high amplitude of the oscillation, in the extreme points (peaks and valleys), the instantaneous miscompare flag is activated. Meanwhile, the output of the persistent miscompare filter keeps increasing until it exceeds the threshold around t = 17.5 seconds. At this moment, the consolidated validity of signal 3 goes to 0 and it is no longer used in the voted value calculation. The voted value quality drops from 4 to 3 and the voted value itself becomes 3.15, which is the median – that, in this case, coincides with the average – between 3.4 and 2.9 (values of signals 1 and 4, that remain valid). It is relevant to notice that, after the beginning of the oscillation and before the miscompare fault detection, signal 3 is still used in the voted value calculation (except while the instantaneous miscompare flag is active). However, the voted value remains limited by the other valid signals; in other words, the fault does not significantly affect the voted value due to the robustness of the median as the calculation method.

At t = 21.5 seconds, as mentioned earlier, the consolidated validity of signal 2 returns to 1 and thus it is again used in the voted value calculation together with signals 1 and 4.

At t = 30 seconds, signal 1 presents a freezing at zero fault, which is immediately perceived by the instantaneous miscompare detection and a few seconds later by the persistent miscompare detection, causing its consolidated validity to become 0 and reducing the voted value quality to 2. It can be observed that, when the persistent miscompare detection flag became active, the instantaneous miscompare detection flag returned to 0 because this detection is disabled when there are only two valid signals.

Finally, at t = 35 seconds, signal 2, which had previously restored its validity, exhibits another fault, this one of the upper limit saturation type. Since there are only two valid signals remaining (2 and 4), this fault is propagated to the voted value, yet in smaller proportion. The voted value is changed to 6.45, which is the median between 2.9 and 10 (values of signals 4 and 2, respectively). The instantaneous miscompare detection is deactivated because of the number of valid signals. The absolute difference between each signal that is still considered valid and the voted value is continuously processed and accumulated by the persistent miscompare filter until it exceeds the threshold. At this moment (around t = 36 seconds), both signals (2 and 4) are declared invalid, since the algorithm does not have enough information to distinguish which signal is faulty. From this time on, the voted value quality goes to 0 and the voted value gets frozen.

5. CONCLUDING REMARKS

This work proposed a voting algorithm to be employed in aircraft systems with quadruple sensor redundancy, since the use of this type of architecture in critical systems such as flight controls has been increasing lately. In the technical literature, voting algorithms that address specifically the detection of oscillatory and split faults were not found, as well as algorithms with a practical approach for voter development for aeronautical purposes.

The proposed algorithm is widely configurable in order to enable its application to a great variety of signals of distinct natures. In some specific cases, it may be necessary to adapt the logic to adhere to peculiar characteristics of certain variables and/or the way the signals will be used. In general, the algorithm is capable of detecting miscompare, oscillatory, out-of-range and 2 by 2 split faults. The corrupted signals are then removed and only the healthy signals are used to compute the voted value, which is the median among them. During fault transients, the voted value is faded in order to avoid abrupt changes in the output.

The tests performed with synthetic data showed that the implemented algorithm works as desired in the tested scenarios. Although tests with synthetic data are useful in the development of a voting algorithm, it is also important to test it with real data in order to mature the logic and to verify if the response meets the specific needs of the system where it is employed. It is recommended that the algorithm be applied to various scenarios that could occur during all aircraft flight phases. Therefore, it is expected that the voter design gets mature enough only during the fleet operation.

6. ACKNOWLEDGEMENTS

The second author acknowledges the support of FAPESP (grant 2006/58850-6) and CNPq.

7. REFERENCES

AAIB – Air Accident Investigation Branch. Airbus A319-131, G-EUPV: AAIB Bulletin No: EW/G2001/08/10. England, 2001. 7 Apr. 2010, <http://www.aaib.gov.uk/cms resources.cfm?_le=/dft avsafety pdf 501829.pdf>.

ATSB – Australian Transport Safety Bureau. ATSB Transport Safety Report: Aviation Occurrence Investigation AO-2008-070 - Interim Factual No. 2. Australia, 2009. 7 Apr. 2010, <http://www.atsb.gov.au/media/1363394/ao2008070 ifr 2.pdf>.

ATSB – Australian Transport Safety Bureau. ATSB Transport Safety Investigation Report: Aviation Occurrence Report 200503722. Australia, 2007. 7 Apr. 2010, <http://www.atsb.gov.au/publications/investigation reports-/2005/AAIR/pdf/aair200503722 001.pdf>.

BEA – Bureau D'Enquêtes e D'Analyses pour la Sécurité de l'Aviation Civile. Interim Report on the accident on 1st June 2009. France, 2010. 7 Apr. 2010, <http://www.bea.aero/docspa/2009/f-cp090601e1.en/pdf/f-cp090601e1.en.pdf>.

Latif-Shabgahi, G.; Bass, J. M.; Bennett, S., 2004, "A taxonomy for software voting algorithms used in safety-critical systems", IEEE Transactions on Reliability, Vol. 53, No. 3, pp. 319-328.

Latif-Shabgahi, G.; Hirst, A. J.; Bennett, S., 2003. "A novel family of weighted average voters for fault-tolerant computer control systems". Proceedings of the 3rd European Control Conference, Cambridge, England. pp. 1-10.

Lee, S. C., 1994, "Sensor value validation based on systematic exploration of the sensor redundancy for fault diagnosis KBS", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 24, No. 4, pp. 594-605.

Newman, R. L.; Lambregts, A. A., 2009, "A review of Fly-by-Wire accidents". Proceedings of the 40th Annual Seminar "Accident Prevention Beyond Investigations". International Society of Air Safety Investigators. Orlando, United States, pp. 1-10.

Parhami, B., 1994, "Voting algorithms", IEEE Transactions on Reliability, Vol. 43, No. 4, pp. 617-629.

Parhami, B., 1992. "Optimal algorithms for exact, inexact, and approval voting". Proceedings of the 22nd International Symposium on Fault-Tolerant Computing. Boston, United States, pp. 404-411.

Patton, R. J., 1991, "Fault detection and diagnosis in aerospace systems using analytical redundancy", Computing and Control Engineering Journal, Vol. 2, No. 3, pp. 127-136.

The Boeing Company. Seattle, Wash, John D. Blair. Signal Selection and Fault Detection Apparatus. C.I. G06F 11/20 U.S. 4,472,806. 3 May 1982; 18 Sept. 1984.

The Boeing Company. Seattle, Wash, Lloyd R. Tomlinson. Midvalue Signal Selection and Fault Detection Apparatus and Method. C.I. G06F 11/18; G06B 9/03 U.S. 4,276,648. 4 Sept. 1979; 30 Jun. 1981.

8. RESPONSIBILITY NOTICE