33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS
2022
SWEDEN

A FAULT INTEGRATOR BASED VOTE&MONITOR DESIGN FOR GRACEFULL DEGRADATION IN MULTIPLE REDUNDANT SAFETY CRITICAL SYSTEMS

Muhammed Enes Arslan[1], Ece Tugrul[1], Mustafa Yuce[1], Nilgun Sever[1], Emre Yigit[1], Tolga Inal[1]

[1]Turkish Aerospace, Aircraft Division

Abstract

Vote and monitor functions are one of the most critical part of the design especially on safety critical multiple redundant systems. Monitoring functions monitor the systems health and welfare during operation in order to make sure that the system is supplied with healthy information during normal, degraded, erroneous and faulty operation. Voting functions are a part of monitoring functions which allows the system to select correct inputs with rigor which adds additional robustness to the system. These two cannot be separated nor they can be compared as they work coherently fulfill the single top purpose of redundancy management for multiple redundant safety critical systems. Fault integrators have been employed in such designs for quite long time. Fault integrator design may vary from system to system. In this study, we developed a novel fault integrator to monitor the validity of the inputs for such safety critical aircraft systems.

Keywords: fault integrator, vote, monitor, safety critical

1. Introduction

Multiple redundant safety critical systems have quite a lot of applications in military and civil platforms. Aviation applications are more critical as this study focuses on such design. The safety criticality of the systems derive the design decision for redundancy in a level of architecture and LRU (Line Replaceable Unit). In this paper architectural decisions and safety objectives which derives such decisions will not be discussed and the system architecture is already decided to be multiple redundant.

Fault integrator designs have been employed in similar algorithmic designs before and their success has been proved on several safety critical systems and aerospace platforms [1-5]. Monitoring of multiple redundant LRUs outputs on a multiple redundant system needs careful consideration during design and the built in capabilities of the multiple redundant LRUs are utmost important as this capability is the main driver of the design. Multiple redundant inertial sensor output monitoring on a multiple redundant flight control system can be given as an example. Voting part of such redundancy management algorithms needs a robust decision maker logic in which fault integrator design becomes important. The design philosophy depends mainly on application specifics and systems architecture but still a valid similar design philosophy can be employed in different parts of the system to design a robust redundancy management algorithm.

In this study we developed a new fault integrator based vote and monitor algorithm for a multiple redundant safety critical system

2. Method

2.1 Main Goal

The main goal of designing such a redundancy management algorithm is to provide a single consolidated input from multiple redundant input channels such as multiple redundant inertial sensor inputs. The method we have employed in this study is summarized below:

- Receive the built in test validities (if exists) for signals to be voted
- Exclude the signals which are invalidated by BITs(Built-In Test)
- With remaining signals, perform voting. Obtain a single consolidated input

- If a signal is invalidated during the process, exclude it from the voting
- Supply consolidated signal to the necessary functions
- Repeat

The overall structure of this method is given below:

- Built-In-Test Validity Construction
- Voting Algorithm
- Final Validity Construction

2.2 Implementation

For a flight control system application the method and structure given above will be used for all sub systems and LRUs that supply input this safety critical systems such as actuators, pilot stick, pilot rudder pedals, inertial sensors and air data probes.

Validation of built in results is one of important aspects of such designs and LRUs BIT capability determines most of the outcome from this function. The functional flow of this BIT validation algorithm is as follows:

- Get Built-In-Test validities from the LRU
- Evaluate the validities, obtain the Built-In-Test Validity for the signal
- Send the obtained validity to voting function
- Repeat

Detecting the erroneous operation LRU or sensor is important in such system as the methods defined above can identify failure cases like total loss, partial loss or system level loss like electrical power loss. In order to be able to identify the erroneous operation sub system/LRU/sensor, the method given below is employed:

- Get the validated redundant signals
- Check if any of the signals deviates from the median for prolonged time
- If a faulty signal is detected, set it as invalid and exclude it from voting
- With healthy signals, calculate the mean
- Send the mean to necessary functions
- Repeat

At the final stage redundancy management algorithm combines built-in test result validities and voting algorithm results to make the final conclusion on inputs health to consolidate all inputs:

- Get Built-In-Test validities from LRU (Step 1)
- Get voting validities (Step 2)
- Set the signal as invalid if any of the validities on Step 1 and Step 2
- Supply the final validity to the voting function

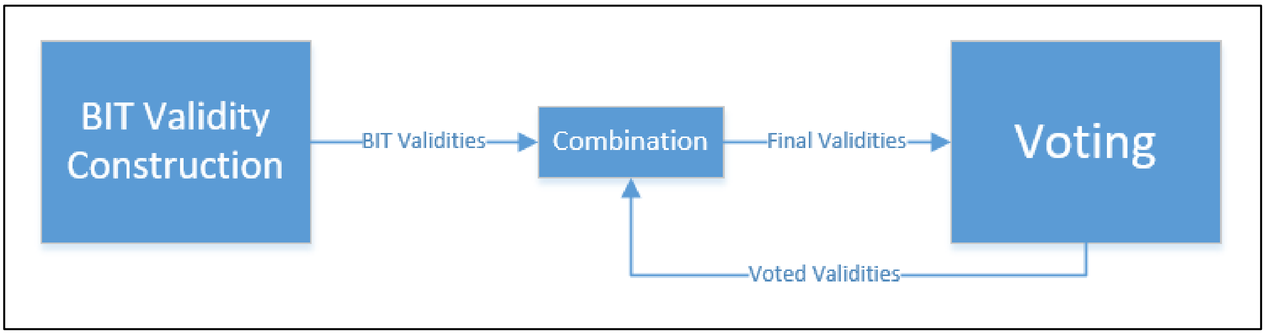

Figure 1 – Final combination of BIT validity results and voting monitors

The fault integrator design prevents the system from transient nuisance alerting and also acts as a gatekeeper for the final decision of vote and monitoring function. The rigor and robustness of the fault integrator is dependent upon derived thresholds for that specific safety critical application such as flight controls.

2.3 Validation of the Voting Algorithm

We generated test harness to validate vote and monitor algorithm by using Matlab Simulink Design Verifier tools. The test harness includes test sequences to test vote and monitor algorithm with several different input sets to check whether the designed algorithm is correct, compliant against requirements, testable and valid. In figure 2, an example for developed test harness model can be seen.

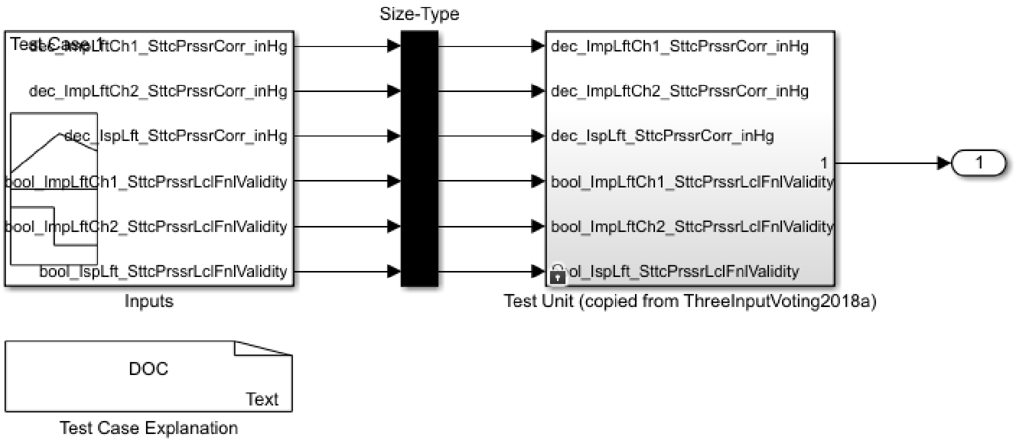
Size-Type
Test Case 1
dec_ImpLftCh1_SttcPrssrCorr_inHg
dec_ImpLftCh2_SttcPrssrCorr_inHg
dec_IspLft_SttcPrssrCorr_inHg
bool_ImpLftCh1_SttcPrssrLclFnlValidity
bool_ImpLftCh2_SttcPrssrLclFnlValidity
bool_IspLft_SttcPrssrLclFnlValidity
Inputs
dec_ImpLftCh1_SttcPrssrCorr_inHg
dec_ImpLftCh2_SttcPrssrCorr_inHg
dec_IspLft_SttcPrssrCorr_inHg
1
bool_ImpLftCh1_SttcPrssrLclFnlValidity
bool_ImpLftCh2_SttcPrssrLclFnlValidity
ol_IspLft_SttcPrssrLclFnlValidity
Test Unit (copied from ThreeInputVoting2018a)
1
DOC
Text
Test Case Explanation

Figure 2 – Test Harness Model

Functional test sequences can be applied to this test harness to test functionality of the developed algorithm. It also helps to make a semantic check (Figure 3). The signal builder allows to develop automatic test cases in order to be able to test all possible test cases.

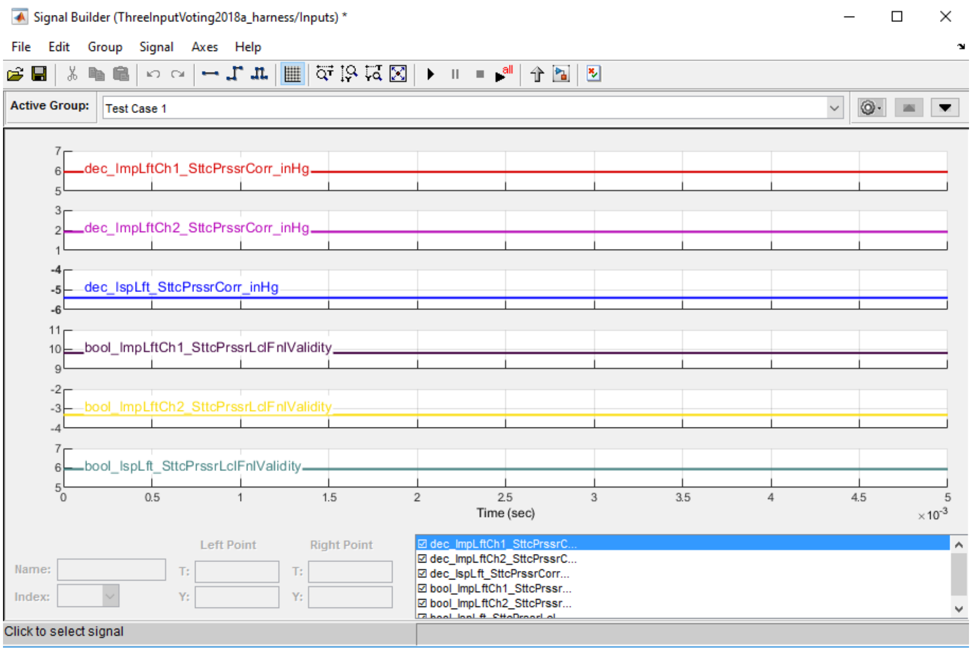
Signal Builder (ThreeInputVoting2018a_harness/Inputs) *
File Edit Group Signal Axes Help
Active Group: Test Case 1
dec_ImpLftCh1_SttcPrssrCorr_inHg
dec_ImpLftCh2_SttcPrssrCorr_inHg
dec_IspLft_SttcPrssrCorr_inHg
bool_ImpLftCh1_SttcPrssrLdFnlValidity
bool_ImpLftCh2_SttcPrssrLdFnlValidity
bool_IspLft_SttcPrssrLclFnlValidity
0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 5
Time (sec)
$\times 10^{-3}$
Left Point
Right Point
Name:
Index:
T:
Y:
☑ dec_ImpLftCh1_SttcPrssrC...
☑ dec_ImpLftCh2_SttcPrssrC...
☑ dec_IspLft_SttcPrssrCorr...
☑ bool_ImpLftCh1_SttcPrssr...
☑ bool_ImpLftCh2_SttcPrssr...
Click to select signal

Figure 3 – The Signal Builder Block with the Test Input Vectors

3. Results and Discussion

3.1 Results

In the first run, the test sequences were developed manually. After testing only 62% coverage was achieved. We also found 3 logic implementation problems within the designed algorithm. Once the algorithm is fixed, on the next run, by using signal builder, all test sequences were generated automatically. This test run ended up with 100% coverage and allowed us to validate our vote and monitor algorithm which allows the system to degrade gracefully.

3.2 Discussion

Fault integrator based vote and monitor algorithms are popular solutions throughout the industry. Developing your own algorithm in conjunction with your own safety critical application is tough and demands tremendous effort. Also validating the algorithm against its requirement set is must do activity in order to make sure that the designed algorithm is doing its intended function. Next step is software and hardware in the loop testing of this algorithm with target hardware. Validating the requirements and testing the algorithm in simulation environment helps designers to better understand their design, fix problems before moulding the design to embedded environment which in several cases a late stage to discover design errors.

4. Conclusion

Developed fault integrator based vote and monitoring algorithm prove itself as a good tool for graceful degradation in multiple redundant safety critical systems. The algorithm can endure several cascade failures and uses BIT results of each LRU to prevent the whole system from a total loss.

5. Contact Author Email Address

mailto: tinal@tai.com.tr

6. Copyright Statement

References

[1] Milijkovic D. Fault detection methods: a literature survey. 2011 *Proceedings of 34TH International MIPRO*, Crotia, 12137501, pp 750-755, 2011.

[2] Laura LP, Dziwgiel R, Wakefield GS. Templates for software fault tolerant voting on results of floating point arithmetic. *AIAA,* Paper No.93-4545-CP, pp 522-530, 1993

[3] Parhami B. Voting algorithms. *IEEE Transactions on Reliability*, Vol.34, No. 4, pp 617-629, 1994.

[4] Broen RB. Performance of fault tolerant estimators in a noisy environment. *AIAA*, Paper No.75-1062, pp 1-7, 1975.

[5] Oliveria LCR, Galvao RKH. *Proceedings of 21st Brazilian Congress of Mechanical Engineering*, Natal Brasil, pp 1-10, 2011