TECHNICAL NOTE

NOVEL AND EFFECTIVE SENSOR VALIDATION ALGORITHMS FOR SAFETY CRITICAL SYSTEMS

Manju Nanda; J. Jayanthi and T.S. Arjun
CSIR National Aerospace Laboratories (NAL)
HAL Airport Road, Post Box No. 1779
Bangalore-560 017, India
Email : manjun@nal.res.in

Abstract

This paper describes a simple, novel and effective implementation and demonstration of sensor validation algorithms for three different and widely used signals for a safety critical embedded application. The challenge in this work was to develop the proposed algorithms with the hardware and system requirement constraints and when there were no off-the-shelf algorithms for this application. The algorithms are developed for analog, discrete and ARINC signals without compromising on the simplicity, reliability and safety. The correctness of these algorithms is verified and validated by means of rigorous laboratory tests and flight trials.

Keywords*: Sensor validation, Safety-critical embedded system, Fault tolerant, Reliability, Nuisance warning*

Introduction

Sensor Validation is required to determine the health of the sensor signals as these signals are used for performing critical flight control functions which either take decisions or provide commands. This paper describes sensor validation algorithms for the safety- critical embedded software used in the indigenously developed14-seater passenger aircraft. The safety- critical system consists of hardware based on Motorola MC68060 processor. The system provides warning to the pilot about the takeoff, landing, over speed, stall, pitch trim and the hydraulic low pressure. Since there was no off- the- shelf algorithm available for this specific application and hardware requirements, it was required to develop the sensor validation algorithm for the sensors interfaced with the system.

The paper describes simple, novel and efficient algorithms developed and successfully implemented for various aircraft sensors. The algorithms developed validate the health of the sensors and the signal value sensed by these sensors. These sensor signals are processed in the system and help the system in critical decision making. Hence it is important to detect the health of the sensor since a faulty sensor is likely to provide an undetected faulty signal giving rise to nuisance warnings or improper computations leading to catastrophe. A good validation algorithm detects a faulty signal and informs the system to prevent a catastrophe or to prevent the nuisance warning based on the fail-operative or fail-off system design.

The approach described for the validation algorithm is different from other approaches because the system has a limitation on the number of input signals. Many are simplex inputs while a few are duplex inputs. Underlying embedded hardware results in this limitation. The need to develop validation algorithms for signals that have no redundancy is a challenge because the validation algorithm should provide the sensor health status and also the data value from the sensor with high reliability. The proposed algorithms cater to these requirements because they are tailor made for the specific application. It has also been established that it can be used for simplex / duplex systems interfaces where the persistence time and tolerance can be configurable based on the algorithms.

The embedded hardware is a dual channel system, with the two channels sandwiched back to back by means of the RS422. The different sensor requirements interfacing with the system motivated to develop these validation algorithms. The sensor configuration for the system is shown in the Fig.1. As seen in the figure, some of the sensors are simplex to each of the channel, in some cases redundant

Paper Code : V66 N2/816-2014. Manuscript received on 05 Jul 2013. Reviewed, revised and accepted as a Technical Note on 28 Feb 2014

signals are going to the same channel and in the worst case scenario only one signal is given to either of the channel. The other system requirement is the availability of the sensor health status and the sensor value within a fixed time frame for the warnings and the control law computations. The algorithm developed for different signals cater to the availability of these signals on each side of the system. To cater to the sensor interface requirements, a modified moving window is used for the analog validation, a modified debouncing logic is used for discrete signals, and a persistence algorithm is used for the ARINC signals.

The verification and validation of the algorithms developed as part of the application software have been rigorously tested in the laboratory for its correctness, reliability, safety and failure management as per the civil verification and validation process standard. The flight tests proved the correctness and safety of the algorithms.

Logically the paper is divided into various sections. In section, Background Wok, gives an overview of the related literature survey to study the various methodologies for sensor validation. In section, Validation Algorithms, talks about the validation algorithms developed for the specific safety critical embedded application. The section, Validating the Signal Validation Algorithms, illustrates the techniques used to verify and validate the algorithms for its correctness and simplicity and in Conclusion section, summarizes the work done and achievement.

Background Work

Real-time systems have become sophisticated. It is a challenge to ensure the reliability, availability, safety and security in such systems. Validation of the sensor signals interfaced ensures the reliability and availability of the data used by these sophisticated systems for further processing. Bickford et.al [3] and Goebel and Agogino [5] discuss the need to validate real time signals for safety critical applications.

There exist various algorithms to perform the data validation coming from the various sensor interfaces. The validation techniques consist of a model based validation techniques, fixed sample validation techniques and variable samples validation techniques as discussed by Ray and Luck [1]. In a model based validation technique the signals are mathematically modeled. This approach is effective but it is suitable only in case the signal nature is not known completely or if there are redundant sensor inputs. In the variation sample validation technique, each sensor signal is analyzed and the stabilization time of each signal is studied. The algorithm uses this stabilization time for computing the validation of the different sensor data. In the fixed sample validation techniques, the algorithm uses fixed time for computing the validation of different sensors. The fixed and variable sample validation approach is less complex and time taken to implement is also relatively small. Model-based techniques used to implement input processing algorithms become complex as we need to generate the code from the model. The model design should be able to represent the algorithm correctly, only then the auto code generated will be correct. This is not straight forward and hence it becomes difficult to implement the validation algorithm using model-based approach as compared to conventional approach of fixed sample and variable sample techniques. Further, to model the system for the first time is complex and time-consuming.

The advantage of model-based validation technique is that we can simulate the algorithms for various scenarios, once correctly implemented, it is easy to change and maintain as compared to the variable and fixed sample technique. Table-1 shows the comparison between the validation techniques.

Some of the other validations techniques are discussed by M. R. Napolitano et.al [6], P. H. Ibargengoytia et.al [7], S. Kasemsumran et.al [8], H. Chafouk et.al [9], K. E. Holbert et.al [2] and X. Wang and Holbert [10]. Napolitano et.al [6] uses the neural network and kalman filters based approach for sensor failure detection used for the flight control systems application. Ibargengoytia et.al [7] describes the importance of Bayesian network in detecting faulty sensors during the data validation. Kasemsumran et.al [8] discusses the cross-validation techniques in chemo metrics applications. Chafouk et.al [9] discusses the parity space approach for validation. Holbert et.al [2]

Table-1 : Comparison of the Validation Techniques

Technique	Code Complexity	Development Life Cycle	Effectiveness
Model Based	Complex	Faster	Effective
Variable Sample	Less Complex	Comparatively slow	Effective
Fixed Sample	No Complexity	Comparati-vely slower	Less Effective

discusses fuzzy logic and Wang and Holbert [10] discusses neural network based data validation approaches for power plant applications.

Many other papers, such as one by Heimdahl et.al [11] discuss the importance of correctness of data in safety critical applications. Zhang and Li [12] describe the detection of online system failure and diagnosis for a dynamic system using multiple models. R. L. Bickford et.al [13] discusses about process improvement due to the real time sensor validation.

The background works on the literatures available for validation of data and signals helped in analyzing the existing techniques. None of the techniques were found appropriate as the current application has some pre-defined constraints such as, dedicated hardware, pre-defined number of signals available (not possible to have duplex signals due to the number of signal limitations) and the mix of analog, discrete and ARINC signals.

Validation Algorithms

The basic approach for algorithm of all the signals is same. In case signals are from more than one source, the signals are compared for a persistence time and with a specific threshold value. The threshold values are decided based on the accuracy and the criticality of the system. The persistence time is decided based on the rate of change of the signal. The value of the signal that comes from a single source is monitored before declaring its health. Sometimes signals from two sources are interfaced to the two channels which results in cross-comparison of the values within the threshold after the persistence time. The accuracy of the validation technique is determined by comparing the validation design to the code implemented. The validation design considers the safety requirements to validate the signal before processing it for critical functionalities. The algorithms also consider the tolerance value for the analog and ARINC signals. For analog signals the tolerance proposed is 2% of the full range of the analog signals. For ARINC signals, the resolution of the ARINC data determines the accuracy. Accuracy does not hold for discrete signals as they exist in only two states.

Following sections describe the validation algorithms developed for the analog, discrete and ARINC signals.

Analog Signal Validation

Analog signals are continuous time varying signals. They are characterized by amplitude, frequency and the phase of the signal. Various aircraft sensors provide analog signals, which have different frequency and amplitude. Analog signals that come from various aircraft interfaces are angle of attack vane sensors, aircraft hydraulic pressure system, pitch trim position sensor from the elevator surface, and the fuel tank sensor from the fuel system. The signal variation for angle of attack vane sensors is different from that of the hydraulic pressure sensor or the fuel tank sensor. The vane sensors, fuel tank sensors are simplex and each of the system channels receive this signal. In case of hydraulic and pitch trim position sensor, the signal is available on only one channel. All of these signals are validated using the modified fixed size moving window approach. The vane sensor and fuel tank signals are validated on each channel of the system and also cross-compared with the signals from the cross channel before using it for computations. The quality of the signals after the validation is indicated with a status flag, which is a binary flag.

The application software has a real time kernel with a scheduling of 25msec. The application software acquires, validates, computes and outputs the data every 25msec. This sampling rate of the signals is dictated by the system and matches well with the rate of change (ROC) of the signal coming from external physical systems. The basic moving window concept logic is shown in Fig.2, which uses a fixed number of samples for computing the health of the signal. Averaging of the fixed number of samples is done once the data, collected during the consecutive cycles, are within the threshold values as per the application requirement. Rate of Change of the signal is monitored in the fixed window. The fixed window of 3 samples is taken and the values of the signal in this window are compared. The thresholds for comparison, the rate at which the signal is read, the signal range are all known a-priori. The signal coming to the system input is read and compared with its previous values. If the read value lies within pre-defined tolerance value, then the signal is qualified as healthy else the signal is monitored for a pre-defined persistence time before declaring it un-healthy. During this monitoring process the signal is declared healthy and the previous healthy value is taken for the computations. The monitoring time is based on the signals settling time in worst case scenario. The accuracy and the resolution of the analog signals were analyzed end-to-end. One such example is of

the Hydraulic pressure. This signal is acquired from the left channel of the system.

Algorithm 1 : Analog Signal Validation 1

```
1. a ← Last 2 bits resolution of the 12 bit ADC
2. b ← 2 % of line noise
3. if a < rate of change of the signal every then
4.        Toln = b
5. else
6.         Toln = a + b
7. end if
8. At any time t,
9. C = Xi
10 P = Xi - 1
11. LP = Xi - 2
12. At time t + f i,
13. LP = P
14. P = C
15. C = Xi at (t + f i)
16. Diff1 = abs (P - LP)
17. Diff2 = abs (C - P)
18. if Diff1 <= Toln && Diff2 <=Toln then
19.        Xvalid = Average (C; P; LP)
20.        validFlag = Valid
21. else if Diff1 <= Toln && Diff2 <= Tolin then
22.       Xvalid = Average (P; LP)
23.        validFlag = Valid
24. else if Diff1 > Toln && Diff2 <= Tolin then
25.       Xvalid = Average (C; P)
26.        validFlag = Valid
27. else
28. if persistCnt <= cntValue then
29.         persistCnt+ = persistCnt
30. else
31.         validFlag = INVALID
32.          Xvalid = 0
33. end if
34. end if
```

The signal flow across the system is shown in the Fig.3. The accuracy of the hydraulic signal is computed based on the flow of the data across the system, the accuracy of the ADC used and the criticality of the signal for the application. To process the analog signals in the computer, we need to convert the analog signal to digital signal. The equivalent digital value undergoes the validation process and once the signal is validated, the final value is converted to the analog value by appropriate conversion formula to determine the correct value as shown below.

- Hydraulic signal analog range: 0.25V to 5.25V.
- Hydraulic signal physical range: 0Psi to 4000Psi.
- Analog to physical signal mapping :0.25V = 0PS I.
- Analog to physical signal mapping :5.25V=4000PS I.
- Accuracy loss after ADC conversion = 0.0152V.
- Scale factor = 4000/5= 800 PSI/V (with a bias of 0.25V for 0 psi).
- In terms of pressure, this loss = 12.16 PSI.
- ±2% tolerance for line noise leads to an accuracy loss of 162 psi.
- Total accuracy lost from ADC input before usage = 162+12.16 = 174.16 psi.
- This accuracy lost is acceptable for the application.

Similar analysis is carried out for all the other analog signals. The tolerance and persistence time of the different analog signals used are shown in Table- 2.

Discrete Signal Validation

Discrete signals are not a function of a continuous time argument. The discrete signals like the digital signals are either in the active state or in the inactive state. The active and inactive states depend upon the nature of the discrete signal i.e. 28V/Open or Ground/Open. The aircraft signals from various switches are usually discrete in nature. These signals are characterized by their settling time, some of the discrete signals are simplex and some are duplex for this application. The discrete signals that interface to the system are from the flap system, landing gear system, pilot control wheel, hydraulics and the pitch trim surface. The flap and the landing gear signals are duplex and provided on both the system channels but the hydraulic and the pitch trim signals are provided only on one channel and are

Table-2 : Analog Signals with Their Tolerance Values

Signal	Tolerance	Persistence Time
Hydraulic Pressure	212 psi	250 msec
Engine Torque	2.508 psig	250 msec
Angle of Attack	2.215 degrees	250 msec
Pitch Trim Position	0.5 degrees	250 msec
Fuel Quantity	40.64 Kg	250 msec

simplex. The landing gear and flap system signals are validated and then cross-compared with the cross channel validated values. Most of the discrete signals are coming from micro switches. The micro switches are activated by the dynamics of the physical system. This dynamics control the closure of the micro switch. The hardware does not provide for any de-bouncing circuitry for these discrete signals. Hence the settling time here also refers to debouncing time. The individual signals are monitored over a period before declaring them healthy or unhealthy. The state of the signal is monitored and if it is observed to be not steady for prefixed time duration then the signal is declared unhealthy. If the signal state oscillates in the beginning but stabilizes eventually during the monitoring period then the signal is declared as healthy. The persistence time or the monitoring time is selected based on the maximum allowable time required for the system for its computations and in extreme cases it is the settling time for the discrete signals from one state to another. The persistence time, settling time, for all the discrete signals is set at 500msec catering to discrete signals coming from the various sources. Till the monitoring time of 500msec the data of the discrete signal is valid and the state is same as the previous healthy value. The monitoring time value of 500msec is very small compared to the rate at which the system responds to these signals. During the persistence time of 500msec the data of the discrete signal is set to valid and previous good state is used for processing. The pseudo code for simplex and duplex signal is as shown below.

Algorithm 2 : Discrete Data Validation for Simplex Signals

```
1. At any time t,
2. Y prev = Yvalid
3. Y current = Yi
   Atime t (i + f)
4. Ycurrent = Y (t + f)
5. Y prev = Yvalid
6. if Y prev == Ycurrent then
7.        persisCnt + = persisCnt
8. if persisCnt >= toggleCnt then
9.        stableCnt += stableCnt
10. if persisCnt == maxCnt then
11. if stableCnt == stblCnt then
12.         Yvalid = Ycurrent
13.         validFlg = VALID
14. else
15.         Yvalid = 0
16.       validFlag = INVALID
17. else
18.       Yvalid = 0
19.       validFlag = INVALID
20. else
21.       Yvalid = 0
22.       validFlag = INVALID
23. end if
24. end if
25. end if
26. end if
```

Algorithm 3 : Discrete Data Validation for Duplex Signals

```
1. Yavalid ← valid value of the signal from one source A
2. Ybvalid ← valid value of the signal from other sourceB
3. At time t
4. if Yavalid == Ybvalid then
5.       Yvalid = Yavalid or Ybvalid
6.       validFlag = VALID
7. else
8.       Yvalid = 0
9.       validFlag = INVALID
10. end if
```

The validation algorithm is valid for all the discrete signals coming to the system. In case of duplex discrete signal, after the execution of the validation algorithm, the duplex signals are cross-compared for their valid states. If the valid states and the values match then the signal is valid. If the valid states do not match then one of the signal is unhealthy. This invalid state of the signal is indicated by the valid flag.

ARINC Signal Validation

Aeronautical Radio Incorporation (ARINC) 429 is a data format for aircraft avionics interfaces. It provides the function descriptions and a list of supported physical and electrical interfaces for the digital information system on an aircraft. ARINC 429 is a predominant avionics data bus for most higher-end aircraft today. It is a point to point communication having 32-bit word that contains five fields namely the parity bit, Sign status Matrix bits, data field, Source Destination Index bit and the label bit [14]. The ARINC signals are encoded and hence these signals are very robust and the noise effect on these signals is negligible. This proves the popularity of these signals in airborne embedded systems.

ARINC signals come from the various aircraft systems namely AHRS (Aircraft Heading and Reference System), ADCU (Air Data Computer Unit), EFIS (Electronic Flight Instrument System) and RDALT (Radio Altimeter). These ARINC signals are read every 25msec. The data coming from these systems are validated based on the criticality and availability of these signals. The signals from AHRS and ADCU systems are duplex whereas the signals from EFIS and RDALT are simplex.

ARINC signal themselves provide their health status so these signals are monitored over the persistence time before declaring them healthy or unhealthy. The algorithm used to implement the validation is slightly different as the threshold computations for these signals are dynamic. The pseudocode for simplex signals is shown below.

Algorithm 4 : ARINC Data Validation for Simplex Signals

```
1. At time t
2. Zvalid = Zi
3. At time (t + f)
4. Zcurrent = Zi
5. if Zssm! = NORMAL then
6. if persisCnt <= cntValue then
7.       persisCnt += persisCnt
8. else
9.       Zvalid = DEFAULT
10.      validFlag = INVALID
11. end if
12. else
13.       Zvalid = Zcurrent
14.      validFlag = VALID
15. end if
```

Algorithm 5 : ARINC Data Validation for Duplex Signals

```
1. Toln = (current ARINCvalue * scale f actor) + bias
2. At time t,
3. Zacurrent = Za
4. Zbcurrent = Zb
5. if Zassm == NORMAL && Zbssm == NORMAL then
6. if abs (Zaccurrent - Zbcurrent) < Toln then
7.       persisCnt += persisCnt
8. if persisCnt >= cntValue then
9.       Zvalid = Average (Zaccurrent; Zbcurrent)
10.      ZvalidFlag = VALID
11. else
12.       persisCnt += persisCnt
13. if persisCnt >= cntValue then
14.       Zvalid = Average (Zaccurrent; Zbcurrent)
15.      ZvalidFlag = CAUTION
16. else if Zassm ==
NOCOMPUTEDDATE77Zbcurrent ==
     NORMAL || Zassm ==
     NORMAL && Zbcurrent ==
     NOCOMPUTEDDATE then
17. persisCnt+ = persisCnt
18. If persisCnt >= cntValue then
19.       ZvalidFlag = (Zacurrent||Zbcurrent)
20.       ZvalidFlag = VALID
21. else
22.       persisCnt+ = persisCnt
23. if persisCnt >= cntValue then
24.       Zvalid = DEFAULT
25.       ZvalidFlag = INVALID
26. end if
27. end if
28. end if
29. end if
30. end if
31. end if
```

Validating The Signal Validation Algorithms

These algorithms are validated by generating test scenarios and executing them statistically and dynamically. The algorithms are validated both at the software level and the hardware-software level i.e., at the embedded level using the Laboratory Test Set Up called the Test Rig.

Validation of the Algorithms Using Tools

The validation of the algorithm is done as per the civil aerospace standard RTCA DO-178B to ensure the requirements are correct, complete and unambiguous.

The first phase includes validation and implementation, of the algorithm, using a standard test tool Rational Test Real Time tool. The test tool verifies the validation algorithm using static signal values. It cannot simulate the actual aircraft signals but the instantaneous values of the signals are statically simulated. Dynamic testing is done on the target using the Test Rig developed for this purpose. Test metrics were generated for the algorithms as per the system and safety requirements. These metrics show the correctness of the algorithm as per the system requirements, testability, and simplicity. Table-3 shows the number of test case executed to validate these algorithms showing the effort for validating them. The % indicates

Table-3 : Metrics of the Validation Results

Signal	No. of TC	Coverage (%)	Complexity	Time Taken
Analog	20	100	16	200 mins
Discrete	3	100	16	30 mins
ARINC	285	100	29	180 mins

the level of testability and coverage of 100% indicates that the algorithm is completely testable.

Validating the Algorithm Using the Custom Made Test Rig

The next phase is validation of the algorithm performing dynamic testing of the system. The real time aircraft signals are simulated using the data acquisition card (ADC and DAC cards) and ARINC cards which can be programmed to generate Analog, Discrete and ARINC time varying signals. The data acquisition card used is a PIO-DA16/DA8/DA4 for analog and discrete.

The entire dynamic test set up is, a dedicated set up, called the Test Rig that simulates the signals. The Test Rig was qualified for the correctness of its results by an independent team who witnessed these tests. The Test Rig set up is shown using the block diagram in the Fig.4.

The analog signals are simulated under various conditions. To get successful validation results, Sine wave signals and random signals are used. Sinusoidal waveform have a constant rate of change-multiple of the sampling frequency and random waveforms have values that change with sampling rate, higher amplitude to simulate noise, and persistence values. The waveforms are generated using Visual Basic graphic user interface including the data acquisition drivers. The application software is modified to include the patch software. This software monitors and logs the validation result. These logged values are sent to the test PC through a RS232 connection. The dynamic test scenarios generated for AOA signal having 0-10 V range are shown below:

- 10V signal consistently having variation less than the tolerance value of 2%.
- 10V signal consistently having variation equal to the tolerance value of 2%.
- 10V signal having variation more than the tolerance for a period of 248msec and less than the tolerance after 249msec.
- 10V signal having variation more than the tolerance for a period of 250msec and less than the tolerance after 250msec.
- 10V signal having variation more than the tolerance for more than 250msec.

The graph in the Fig.5 is of the expected test result against observed test values for the analog validation testing. As can be seen in the figure, at 5 seconds, we are simulating a failure scenario. We generate a variation in the AOA signal > tolerance value for more than the persistence time i.e. more than 250msec. We have simulated a condition where the AOA signal drops to zero, but the validation algorithm does not drop to zero but retain a previous good value with a status flag showing a failure. Hence there is a difference between the expected and observed value. Similar approach is used for discrete and ARINC signals. Fig.6 represents the discrete signal simulation and Fig.7 represents the ARINC signal simulation.

Figure 8 shows the test software developed in the data acquisition PC to test the dynamic response of the validation algorithm.

Metrics for Validating the Validation Algorithms Using Simulation

The test scenarios i.e., the test cases were developed to test these algorithms, which include 80 test cases for analog signals, 260 test cases for discrete signals and 203 test cases for ARINC signals.

On Flight Validation

During the test flights, the validation algorithm is tested for reliability. The validation algorithm is a part of the application software which undergoes the flight tests. The post flight data analysis indicates that the embedded system did not send any nuisance warning during the flights. The only warnings sent by the system were the valid warnings indicating the preciseness and reliability of the validation algorithms. Figs.9 and 10 shows the post-flight of Angle of Attack (AOA) signal analysis. Table-4 shows the metrics of the total number of flights on the two prototypes and the reported system failures. No reports of system failures, implies that the software architecture is well designed and robust enough for qualified hardware.

Table-4 : Aircraft Flight Test Metrics		
Aircraft	Number of Flights	System Failure
Prototype I	123	No Failure
Prototype II	20	No Failure

The sensor validation algorithm is a part of the software architecture.

The limited trials show the correct performance of the validation algorithms during the flight trials since there were no false alarms generated by the system during the flight. System failure here means that the application software performed the functionality as it was intended to do and there was no hardware system failure from its performance point of view. These limited trails showed the correct performance of the software i.e. to provide warnings and to avoid nuisance and false positive warnings. The flight tests validate the correctness, robustness and failure resistance feature of the validation algorithm. The algorithm has prevented failures in the system.

Conclusion

Data validation of the various signals is critical for embedded applications but more so for safety critical embedded applications. Any error in computation affects the safety of the application. An efficient validation technique detects faults in the sensors and communicates it to the application. This ensures the performance and reliability of data and accuracy of the results. An inadequate validation technique causes computational error which results in embedded system failure.

The paper proposes three computationally efficient algorithms, using fixed sample algorithm with modifications to accommodate different signals, which is effective like the model based and varying sample validation technique. There did not exist an off the shelf algorithm for sensor validation with these requirements and application.

The efficacy of the algorithm is evident as results observed during the flight trials were reliable and there were no records of nuisance warning. The algorithm is versatile because it can be used for redundant or single source signals. The metrics collected at the various levels, laboratory tests, and test flights shows highly satisfactory results. The success of the tests proves the correctness of the validation algorithm with reference to system requirement and the data collected during the flights prove the reliability of the algorithm.

References

1. Ray, A. and Luck, R., "An Introduction to Sensor Signal Validation in Redundant Measurement Systems", IEEE Control Systems, pp. 44-49, February 1991.

2. Holbert, K.E., Heger, A.S. and Ishaque, A.M., "Fuzzy Logic for Power Plant Signal Validation", Proceedings of the Ninth Power Plant Dynamics, Control and Testing Symposium, May 1995.

3. Bickford, R.L., Bickmore, T.W., Meyer, C.M. and Zakrajsek, J., "Realtime Sensor Validation for Propulsion Systems", AIAA Defense and Civil Space Programs Conference and Exhibit, October 1998.

4. Bickford, R.L., Bickmore, T.W. and Caluori, V.A., "Real-time Sensor Validation for Autonomous Flight Control", AIAA Joint Propulsion Conference and Exhibit, July 1997.

5. Goebel, K.F. and Agogino, A.M., "Sensor Validation and Fusion Forautomated Vehicle Control Using Fuzzy Techniques", Journal of Dynamic Systems, Measurement and Control, Vol.123, pp.145-146, March 2001.

6. Napolitano, M.R., Windon, D.A., Casanova, J.L., Innocenti, M. and Silvestri, G., "Kalman Filters and Neural Network Schemes for Sensor Validation in Flight Control Systems", IEEE Transactions on Control Systems Technology, Vol.6, No.5, pp.596-611, September 1998.

7. Ibargengoytia, P.H., Sucar, L.E. and Vadera, S., "Real Time Intelligent Sensor Validation", IEEE Transactions on Power Systems, Vol.16, No.4, pp.770-775, November 2001.

8. Kasemsumran, S., Du, Y. -P., Li, B. -Y., Maruod, K. and Ozaki, Y., "Moving Window Cross Validation: A New Cross Validation Method for the Selection of a Rational Number of Components in a Partial Least Squares Calibration Model", The Analyst, Royal Society of Chemistry, Vol.131, pp.529-537, 2006.

9. Chafouk, H., Hoblos, G., Langlois, N., Gonidec, S.L. and Ragot, J., "Soft Computing Algorithm to data Validation in Aerospace Systems Using Parity Space Approach", Journal of Aerospace Engineering, pp.165-171, July 2007.

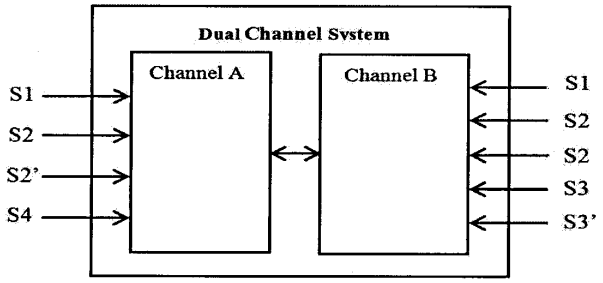
10. Wang, X. and Holbert, K.E., "A Neural Network Realization of Linear Least-square Estimate for Sensor Validation", Proceedings of the Ninth Power Plant Dynamics, Control and Testing Symposium, May 1995, pp.15.01-15.15.

11. Heimdahl, M.P., "Safety and sOftware Intensive Systems: Challenges Old and New", An International Conference on Software Engineering, May 2007, pp.137-152.

12. Zhang, Y.M. and Li, X.R., "Detection and Diagnosis of Sensor and Actuator Failures Using Imm Estimator", IEEE Tansactions on Aerospace and Electronic Systems, Vol.34, No.4, pp.1293-1313, October 1998.

13. Bickford, R.L., Bickmore, T.W., Meyer, C.W. and Zakrajsek, J.F., "Real Time Flight Data Validation for Rocket Engines", 32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Buena, FL, July 1996.

14. "Arinc 429 Protocol Tutorial", July 2004, [Online], Available: http://www.gefanucembedded.com /newsevents/whitepapers/1956/AFC-WIK.



S1: Simplex sensor signal on each channel
S2, S2': Redundant sensor signals on each channel
S3, S3': redundant sensor signals on one channel
S4: Simplex sensor signal on one channel

Fig.1 Variable Sensor Configuration

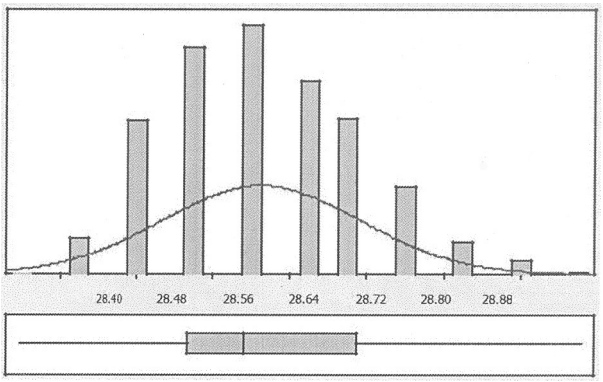


Fig.2 Moving Window Concept for Analog Signal Validation

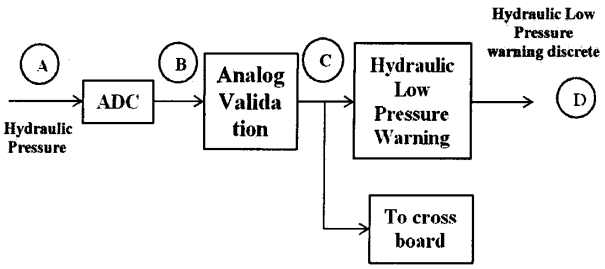


Fig.3 Hydraulic Tolerance Computation and the Signal Flow

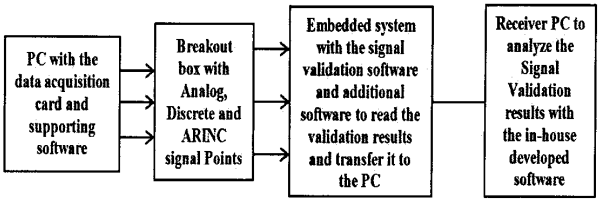


Fig.4 Test Set-up in the Laboratory

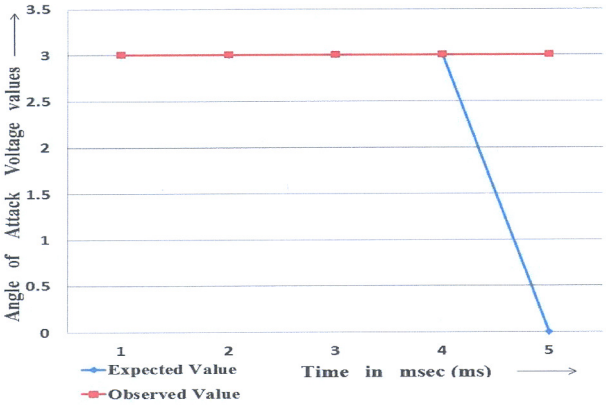


Fig.5 Validation Result of Analog AOA Signal

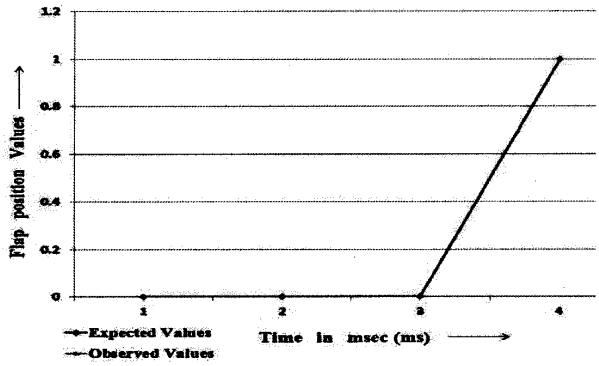


Fig.6 On Flight Data for Hydraulic Low Pressure

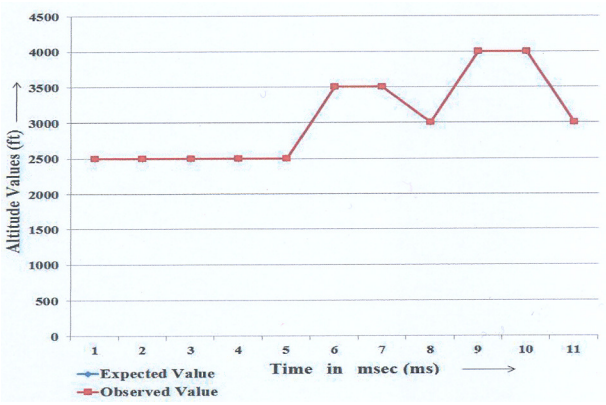


Fig.7 On Flight Data for Altitude Values

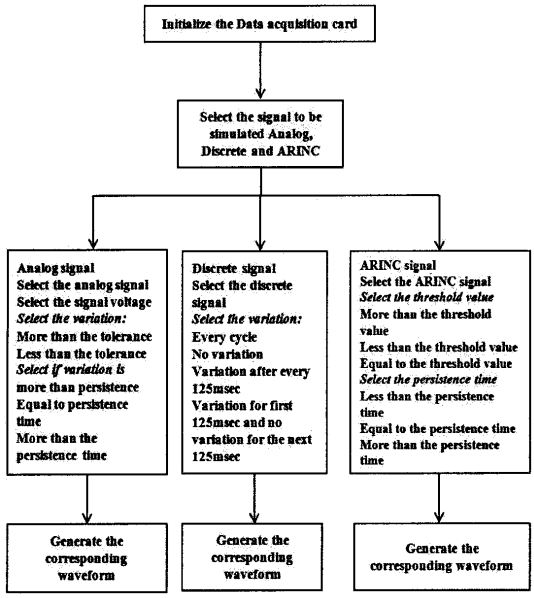


Fig.8 In-house Test Software for Data Acquisition

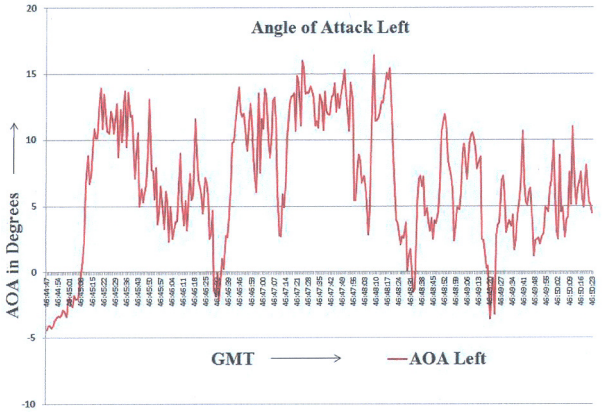


Fig.9 On Flight Data Validation for Left AOA Analog Signal

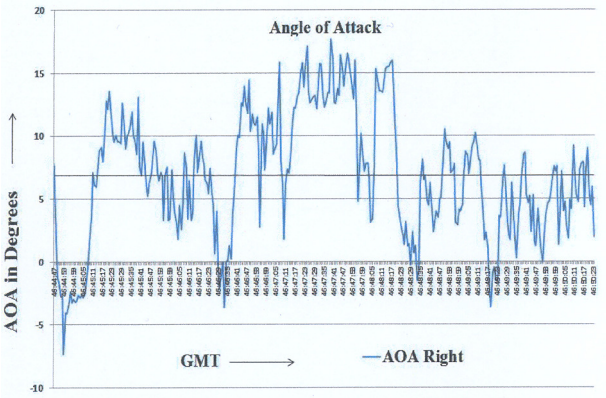


Fig.10 On Flight Data Validation for Right AOA Analog Signal