Global Journal of Researches in Engineering: D
Aerospace Science
Volume 17 Issue 1 Version 1.0 Year 2017
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN:2249-4596 Print ISSN:0975-5861

Model-Based Analysis of Safety Critical Validation Algorithm

By Kushal K S, John Paul J, Dr. Manju Nanda & J Jayanthi
CSIR National Aerospace Laboratories

Abstract- Safe operation of a critical embedded system requires reliable information about the state of the system and signal condition of the system. Validity of sensors which measure the process variables are of great importance. Signal validation comprises of detection, isolation and characterization of faulty signals. Signals that are validated are critical for their increased availability in the system. Model-Based Engineering (MBE) approach provides means of modeling, analyzing, and validating the signals for critical embedded system design, and development. The abstract nature of the models provides mechanisms to analyze verify and validate the system functionality, at a much early stage in their development process.

Keywords: analog input processing, stall warning system (SWS), model-based engineering (MBE), safety- critical system, NI LABVIEW.

GJRE-D Classification: FOR Code: 0901996

Strictly as per the compliance and regulations of:

Model-Based Analysis of Safety Critical Validation Algorithm

Kushal K S [α], John Paul J [σ], Dr. Manju Nanda [ρ] & J Jayanthi [ω]

Abstract- Safe operation of a critical embedded system requires reliable information about the state of the system and signal condition of the system. Validity of sensors which measure the process variables are of great importance. Signal validation comprises of detection, isolation and characterization of faulty signals. Signals that are validated are critical for their increased availability in the system. Model-Based Engineering (MBE) approach provides means of modeling, analyzing, and validating the signals for critical embedded system design, and development. The abstract nature of the models provides mechanisms to analyze verify and validate the system functionality, at a much early stage in their development process.

In this paper we present MBE approach to analyze the input signal processing algorithm with the case study of analog signal for Stall Warning System (SWS) of an aircraft using NI LabVIEW. The approach helps in analysing the functionality and completeness of the algorithm, mathematically and by simulation. The validation of analog signals with different frequencies and amplitude establishes data prudency and maintains the integrity of avionics systems. The result of this approach highlights the advantage of using MBE which enables in analysing the data algorithm for its correctness and guarantees the properties of the model early in the development life-cycle. MBE approach also helps in re-assuring the integrity of the system, before it is developed. This also terminates the contiguous data set and annunciates the sensor fault conditions.

Keywords: analog input processing, stall warning system (SWS), model-based engineering (MBE), safety- critical system, NI LABVIEW.

I. Introduction

Validity of the data input-output of Safety-Critical Systems such as avionics systems, medical instruments, and nuclear systems is critical and needs to be analyzed for its functionality, performance, and safety. Complex avionics systems encompassing multiple critical sensor data associated with the plurality of sensors that needs to be validated for the correct data and also detect fault condition of the sensor. Productivity challenges are on the rise with the increase in the complexity of the software involved in Safety-Critical Systems. Such systems are considered to be least intolerant with safety of the overall functionality of the system, as well as dependency and reliability of the system. Model-Based Engineering (MBE) approaches provide the means of creating and analyzing the models of the Safety-Critical System. These approaches accomplish the task of providing prediction and analyses capabilities of various operational qualitative attributes like performance, productivity, reliability and security. In the development life-cycle, with the application of Model-Based Engineering (MBE) approaches, the system-level problems, are detected at an early stage during their development process which are usually discovered during the system integration and acceptance test, using the conventional approaches. This avoids in rework during the later stages of their development and also minimizes the maintenance cost. The paradigm of developing the application software for Safety-Critical Systems are shifting with the introduction of Model-Based Engineering approaches. The graphical representation of the Safety-Critical System control algorithm application software, using MBE, serves the primary purpose of representation of the application software as a re-usable executable model. This can be re-used at any stage during their development process in the entire life-cycle. This executable model analyzes the predicted system properties and validates them against the system implementation, considering the evaluation of various quality assurances attributes in the course of analyses.

Models representing a Safety-Critical System integument of data-flow, topology and the behavioral aspects of the application software, are created using a suitable modeling language. This modeling language is regarded as formal representation of the application software being developed with precise perception of the actual implementation of the control logic. This ultimately forms the basis for conformance testing against the system requirements.

Modeling language used in creating the models serves as a multi-purpose graphical representation encompassing all the aspects of the Safety-Critical System application software as an abstract. Laboratory Virtual Instrument Engineering Workbench (LabVIEW), a system design and development environment from National Instruments, in conjunction with the Visual Programming Language (VPL) [1] provides the means of creating the Safety-Critical System models. LabVIEW basically makes use of VPL, wherein the application software is created by manipulating the software components graphically as symbols along with the

Author α σ ρ ω: Aerospace Electronics & Systems Division, CSIR National Aerospace Laboratories Bangalore, India.
e-mails: ksk261188@gmail.com, johnatlan@gmail.com, manjun@nal.res.in, jayanthi@nal.res.in

textual notations. LabVIEW makes use of "G" language – a graphical language used for systems design and simulation.

Using LabVIEW the Safety-Critical System application software model is created precisely in coherence with the system requirements, describing all the aspects of the system. This MBE approach with LabVIEW renders the model to define the domain oriented abstractions that define the system components, states, transitions between the states. The validation of such clutter less models to cover different levels of formalities like model checking [3], prototyping and simulation [2]. By combining the effectiveness of the domain specific abstract models and its implementations with prototyping and simulation the application software model is validated for its functionality.

The paper is organized as follows: section II, provides an insight into the related works carried out with MBE. With section III, we introduce to the characteristic features of analog input signal data in Stall Warning System (SWS) and the need for validation. Afterwards, section IV focuses on the analog input processing algorithm and the implementation of the algorithm for different analog signals of SWS, using LabVIEW. Section V sketch the results and summarize the overall validation process. Section VI concludes the work discussed in the context and comment on future work to be done with this reference.

II. Literature Survey

The need to validate the health of the input data signals which are interfaced with certain critical decision making systems is very much essential, in aerospace applications and avionics systems. This helps in the differentiation of the signals as faulty and healthy. The process of differentiating the signals as two different categories, are handled by efficient algorithms developed and implemented. Manju Nanda et al. [4] had proposed a contemporary validation algorithm in validating the signals and thus helping the underlying systems in making appropriate critical decisions. Also the need to verify the correctness of the abstracts of the models of a system under test, corresponding to Model-Checking and Model-Based Testing was proposed by Stefan Gulan et.al.[7]. The early introduction of these approaches for safety mechanisms relevant to the safety of the software was evaluated with an industrial case study sufficiently to address the relevant failures in the systems and its software components.

The exponential growth in the discovery of the system-level faults at the development stage in SDLC process has adverse turn-around effects, such as increase in the cost of development and the rigorous changes that may be subjected to the system and its underlying software components. This can be overcome with the use of Model-Based Analysis and Validation approaches as suggested by Peter H Feiler [8]. With MBE a single source approach is provided to the developer to analyze the system for its operational attributes as well as the system qualitative attributes. This is more reliable through model annotations. Also Bringmann et.al. [9] has proposed with the help of an industrial case study that the quality assurance with the model-based developments with the introduction of graphically testable models representing the software components of the system under test enables the user to understand the architecture of the algorithm of the software. It was also suggested that it helps in expressing complex, fully automated closed-loop test scenarios in real-time. Using MBE approaches for modeling the software component of a system, it is like performing the actual implementation of the algorithm or the software component and defining their relationship between each other. This was proposed by Andree Blotz et.al [10]. The models are created using different languages and are regarded as the formal representation of their functionalities. With all these findings the complex processing algorithm of SWS/AIC i.e. the Analog Input Processing algorithm is modeled using NI LabVIEW. Its operations, capabilities in validating the input data from various aircraft interfaced sensors are discussed in this paper.

III. Analog Input Signal Data of SWS

The digital computer based Stall Warning System (SWS)/ Aircraft Interface Computer (AIC) [5], a state-of-art designed around a customized APM2000 module by SAGEM. The module consists of two processing units Motorola MC68060 along with the co-processor units MC68360, designed as per DO-178B standards [6]. The system consists of input interfaces to different signals of types ARINC429, Discrete and Analogous and processes them accordingly. The SWS systems also provide a stall warning by activating the shaker actuator of the aircraft and indicating them suitably through LED, on the Caution Warning Panel (CWP).

The analog signals read by APM2000 kernel when invoked by the application software and the inputs that are acquired are converted into digital signals for processing by the processor units using a 12-bit Analog-Digital Converter (ADC). The acquired values are made available to the Analog Input Processing algorithm application software in a specified memory location for further processing by the kernel. Any fault or error during the acquisition is examined and returned after the validation by the Analog Input Processing algorithm application software.

a) Angle of Attack (AOA) Sensors Interface

The Angle-Of-Attack (AOA) is the physical angle measured between the chord of the aircraft wing and

the relative wind direction. There are basically two AOA vane sensors that act as potentiometers. Both the left and right potentiometer sensors are excited by 11.3V DC power supply. As an analog input for the system, the physical range is from -10° to +40°. The DC analog signal ranges from 0V corresponding to -10°, while 11.3V corresponds to +40°. The value of the sensor reading is stored in the memory and used to estimate the AOA value during validation.

b) Fuel Remianing Input Interface

The SWS/AIC have the interface to the fuel system to get the fuel remaining in the fuel tank of the aircraft. This input from the interface is obtained as an analog input from the fuel tank capacitor probes provided for both the left and right channels in the aircraft. The physical analog range of the fuel remaining varies from 0V DC corresponding to 0kg, to 5V DC corresponding to 1000kg. But there is a constraint of minimum fuel remaining to be 12.5kg. The fuel value corresponding to the amount of fuel present during the take-off is stored in the memory location. This value is used for the weight estimation during the validation.

c) Torque Pressure Transducer Sensor Input Interface

The torque pressure transmitter is connected to the torque meter in the cockpit. The SWS/AIC system shall have an interface with the torque pressure transducer and reads the engine torque as an analog input. The transducers are present on both left and right channels of the aircraft. The engine torque input ranges from 0 to 5V DC, corresponding to 0 to 44.34psig equal to 3.684V. Here the torque value being greater than or equal to 16.85psig (38% or 1.3999V) indicates that the aircraft is ready for take-off.

For the landing condition the torque value shall be less than or equal to 20% (8.868psig or 0.736V) on both the left and right engines. Even if there happens to be a value of 50% torque on one engine (22.17psig or 1.824V) if the other engine is in-operative with a torque value less than 10% (4.434psig or 0.368V).

d) Pitch Trim Position Sensor Input Interface

The SWS/AIC have the interface with the pitch trim potentiometer to get the position of the pitch trim tab as an analog signal from the potentiometer. The pitch trim position sensors are physically combined as a single channel from both the right and left channels. The pitch trim sensor is excited with a voltage value of 11.3V. The analog input signal acquired from the pitch trim position sensor varies from -15o to +8o. This input is used to limit the electrical travel of the actuator. An input voltage value of 11.08V is read when the excitation voltage of 11.3V is provided across the potentiometer with a bias of 5.54V. The error tolerance is defined to be +/- 0.065V.

e) Hydraulic Pressure Sensor Input Interface

The SWS AIC have been interfaced with the hydraulic pressure sensors to get the value of the hydraulic pressure as analog input. The input range varies from 0-4000psi, corresponding to 0.25V to 5.25V DC.

IV. Implementation of Analog Input Processing Algorithm

The continuously time varying signals from a plurality of aircraft interfaces like AOA sensors, Fuel Tank sensors, Pitch Trim sensors, Hydraulic Pressure sensors are obtained and are independent of each other. These signals obtained from various sensors, which are either simplex (from a single source) or duplex (from dual sources) needs to be validated and compared to measure the quality of the signal/data received and indicate it via a validity flag.

The rate of change of the signal from the physical system is matched to a sampling rate of 25millisecond. This is the same time duration for which the algorithm is designed to read the signal/data recorded by the sensor. During the continuous acquisition of signal/data from the sensor, for a persistence time of 250millisecond, implying that 3 samples are considered in an averaging window, the rate at which the acquired signal/data changes is constantly monitored. This rate of change is compared with a pre-defined threshold value. The recently acquired signals are compared with the previous samples of signal/data and monitored for their values, which may be well within the specified tolerance value. Based on the comparator output the signal/data is termed as healthy or unhealthy, by monitoring it over the period of persistence time. The process of analyzing the analog input signals is as shown below in Figure 1. The basis for this algorithm is that, any signal acquired from any of the aircraft sensors is to be declared as valid or invalid, based on the fixed sample approach. The acquired signal is declared valid or invalid, over a persistence time of 250ms, based on the comparison with their difference between the current valued signal/data that has been acquired with the previously acquired signal (if nothing then to be considered as zero), termed as Difference |B| and the difference between the last previous valued signal/data to that of the previously acquired signal/data as Difference |A|. The magnitudes of both the differences are then compared with the priori specified tolerance band (default +/- 2% of the nominal value – full range value). This range check is carried out for all the acquired input signals/data and the engineering value conversion is done for the analog inputs.

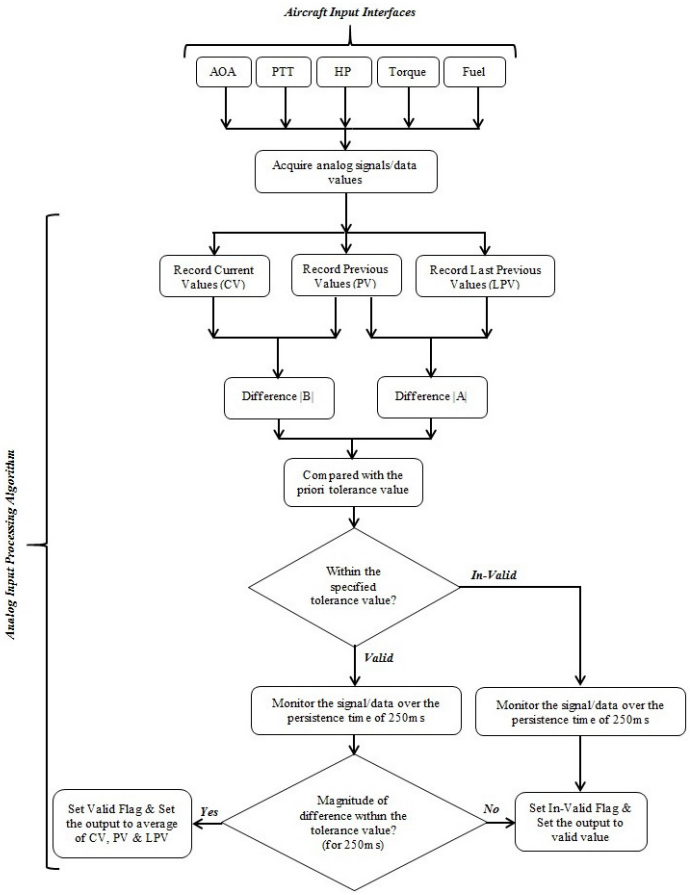

Figure 1: Control Flow of the Analog Input Processing Algorithm

In case the acquired analog input signal is out of bounds of the specified tolerance band, then the output value is clamped to the minimum value if the acquired signal value is less than the minimum value. It is clamped to maximum, if the acquired signal value is more than maximum range specified. This is done along with the storage of the association of the suitable flag status, set either as Valid or Invalid. The purpose of the acquired signal is only for the validity or the invalidity check. For all the analog inputs that are acquired from the sensors, if the signal/data is invalid then the invalid flag is set to TRUE and the output data is latched to a particular valid data that was acquired currently, previously or past the previous acquisition. In case of valid signal/data, the valid flag is set TRUE and the average value of the current, previous and the value past the previously acquired signal/data is latched as the output data. This analog input processing algorithm was developed using Model-Based Engineering (MBE) approach, using LabVIEW from National Instruments. This approach helps in defining the processing control algorithm software components as models that can be re-used and provides more efficiency and robustness as compared with other conventional approaches.

A. Translation of the Analog Input Processing Algorithm to Model

1) Functional Translation

The LabVIEW tool suite is used in modeling the analog input processing algorithm application software. The models that are designed in the LabVIEW development environment will be termed as Virtual Instruments (VI), as the models virtual represents the physical implementations of the system being designed and developed. This tool suite provides the option of categorically and hierarchically modeling features, with the option of defining the subsystem modules/components as Subsystem Virtual Instruments (Sub VI) and can be re-used as library components. This feature is used in the implementation of the Analog Validation Loop and the Data Sorting Loop. The hierarchical categorization of the loops in the model block diagram shown in Figure 2 is as follows;

i. Data Sorting Loop – This loop sorts the acquired analog signal/data in the form of an array and the array is sorted into three different categories as Current Valued, Previous Valued and Last Previous Valued Array. The initial element values in the Previous and the Last Previous Valued Arrays will be considered as zero and later is shifted with the acquisition of the new signal/data values from the aircraft interface sensors.

ii. Analog Validation Loop – This loop categorizes the analog signal/data value as a valid or an invalid signal/data by taking the difference of their magnitudes and comparing it with the priori specified tolerance value. If invalidity persists, the Invalidity Counter is incremented till the persistence time (i.e. for 250ms) as 10 counts, with each sampled signal/data value for 25ms. Upon reaching the counter value of 10 with invalidity, the Invalid flag is set TRUE and the output value is latched to the valid data value among Current Valued, Previous Valued and Last Previous Valued Array.

iii. Validity/Invalidity Declaration Loop – This loop monitors the validity/invalidity based on their magnitude differences between the Current Valued, Previous Valued and Last Previous Valued Array and comparing the difference value with the tolerance value for the full scale range. The valid/invalid flag is set TRUE conditionally based on the compared output value.

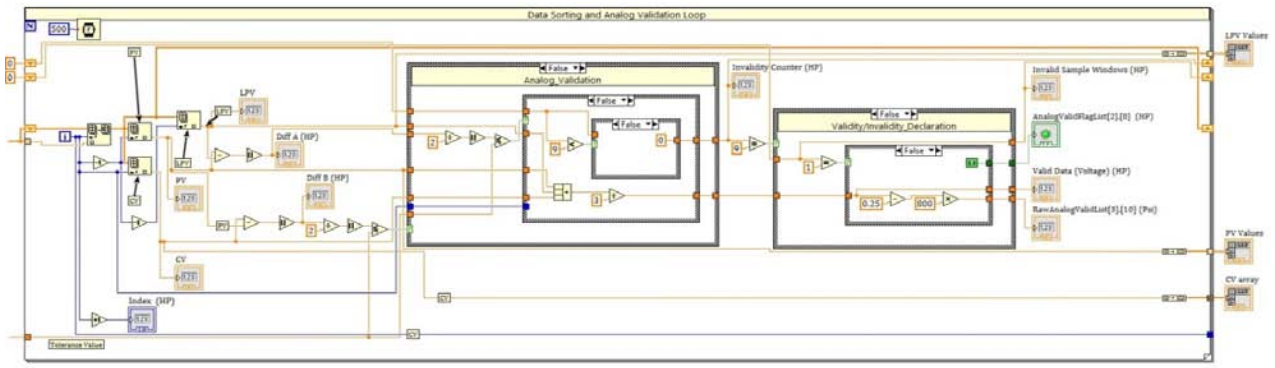

Figure 2: Analog Input Processing and Data Sorting Model (Block Diagram)

The Current Valued, Previous Valued and Last Previous Valued Array are displayed on the Front Panel in LabVIEW, as shown in Figure 3. The LabVIEW tool suite has an inherent feature of providing two sets of windows for the VI. The Front Panel layout representing the UI or the outer layer of the system that is visible with suitable controls and indicators, while the Block Diagram layout which represents the actual layout wherein the system implementation is designed and developed using various available library components as Sub VI. Each Sub VI being categorized based on their applications, functionality and characteristics (Hardware or a Software component), stored as library components that can be re-used.

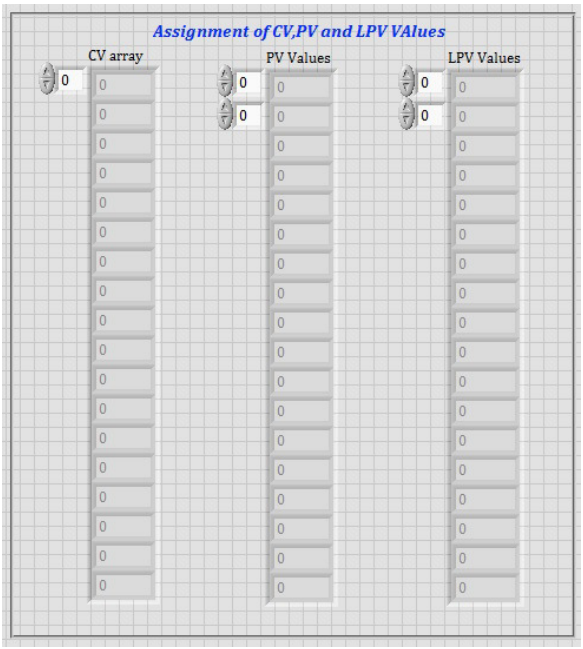

Figure 3: Acquired Analog Data Sorting as Current Valued Array, Previous Valued Array & Last Previous Valued Array (Front Panel)

2) Performance Translation

a) Hydraulic Pressure Sensor Input Interface

The actual system implementation for Hydraulic Pressure Sensor Input Interface is designed as shown in Figure 4. In case of the Hydraulic Pressure Sensor input being invalid the output data is computed based on the lower and upper limits as explained in III.E, and latched to the previous/current/last previous valid data. The valid flag is set to FALSE and there is no computation warning being generated for the Hydraulic Pressure by the SWS/AIC. The block diagram implemented is as per the specifications for its analog input signal/data scaled to its equivalent digital voltage valued array. The range is also validated for Hydraulic Pressure Sensor Input as shown in Figure 4.

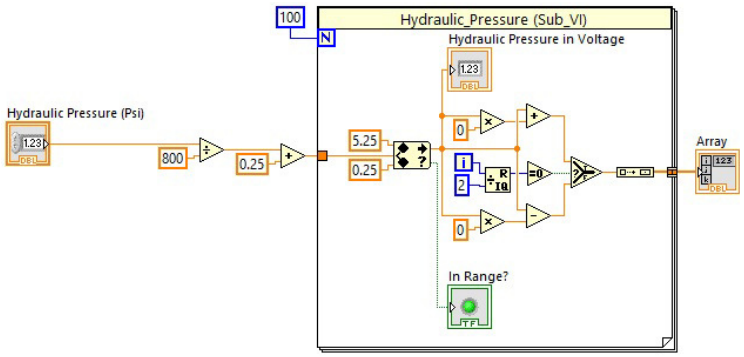

Figure 4: Hydraulic Pressure Sensor Input Interface Block Diagram as Sub VI (Virtual Instrument)

The array is fed as input into the error injection loop, as shown in Figure 5. This loop is present in all the analog input interface modules. This loop allows the user to define the error value i.e. the user has the liberty to set any error value. The tolerance is to be adjusted to the full scale range of the analog input interface and the array is resized to the actual input array size in this loop. Here in this loop, the privilege of inserting the error value into a particular index of the array is also provided. The error injection is controlled using the LabVIEW control switch termed as Error ON/OFF. This helps in inserting the error value in the index of the array during the simulation, dynamically. This is simulated and as well being considered for computation of the warning suitably from the index/time of its insertion into the array. An error valued data is inserted for a time factor of 25ms, every time in the array. This array of input with or without the error is fed into the Analog Validation and Data Sorting loop for the data to be sorted and validated as per the specified limits.

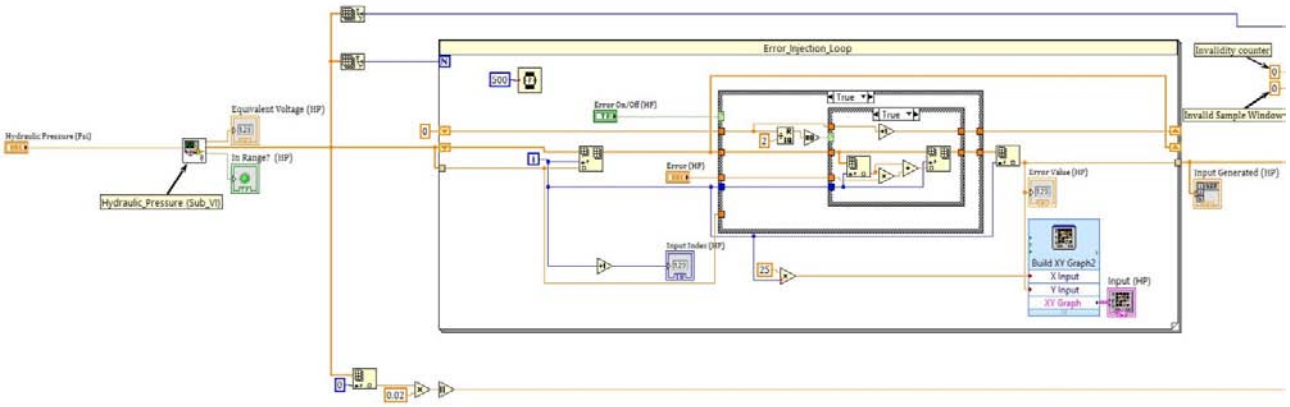

Figure 5: Hydraulic Pressure Sensor Input Interface Error Injection Block Diagram

b) Fuel Remaining Input Interface

The Fuel Remaining Input Interface system implementation is done as shown in Figure 6. There are two fuel tanks present in an aircraft. One in the right wing, which corresponds to the right channel while the other in the left wing corresponding to the left channel. During the Analog Validation, in case the right fuel tank input on the right channel is invalid then the output data is computed and latched to the previous/current/last previous valid data. This means that the Analog Input Processing algorithm shall use the average value of the last correct/valid weight estimate and the aircraft with no fuel remaining.

The same case applies with the left fuel tank input on the left channel being invalid, the output data is computed and latched to the previous/current/last previous valid data. And the SWS/AIC Analog Input Processing algorithm uses the average value of the last correct/valid weight estimate and the aircraft with no fuel remaining from the left channel. In both these cases, the valid flag is set to FALSE. If the data from either or both the channels are valid then the average value of all the three data sets, i.e. previous valued, current valued and the last previous valued data from the array is computed and the output data is set to the average value. The valid flag is set to TRUE. The process of error injection is repeated for this Sub VI also and the array data corresponding to Fuel Remaining is also sorted and validated using the Analog Input Processing and Data Sorting Sub VI loop, as shown in Figure 5.

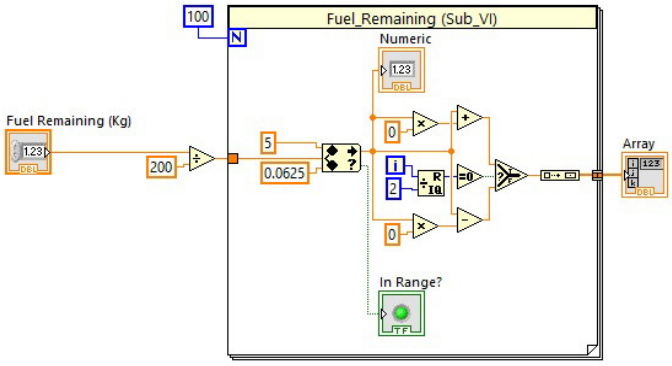

Figure 6: Fuel Remaining Input Interface Block Diagram as Sub VI

c) Angle-Of-Attack Sensors Interface

The model implementation with the given specifications in III.A, for Angle-Of-Attack Sensor Interface is as shown below in Figure 7. There are two channels, (Left & Right Wings) with AOA sensors. The range of the value sensed by the AOA sensor is being validated and the array of inputs is built. The input array is fed into the Analog Input Processing and Data Sorting algorithm Sub VI loop for validation and data sorting, respectively.

In case, the left AOA sensor sensed data is invalid then the AOA value shall be computed for the output data and latched to the previous/current/last previous valid data. Similarly in case of the right AOA sensor sensing an invalid data set, then the AOA value shall be computed for the output data and latched to the previous/current/last previous valid data. In both the cases the valid flag is set to FALSE. For the AOA sensors, as is special case, the mounting error for the left channel AOA with respect to the right channel AOA sensor with a zero bias value of 2.3634V is used in the application software, shown in Figure 7, during the data set being valid. Also the dead band of 0.724V and 0.364V shall be used for left AOA and right AOA engineering value conversions. Different fixed error value considerations are being made to distinguish and compute left channel AOA with the right channel AOA, as 1.389o and 0o respectively. A suitable method such as Ratiometric method is used to compute the AOA values for better accuracy.

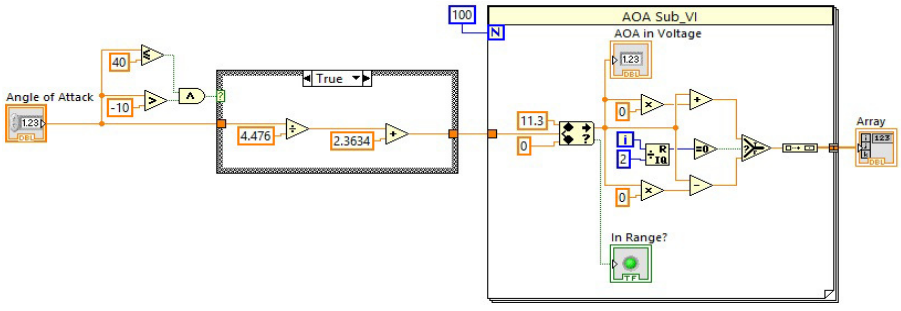

Figure 7: Angle-Of-Attack Sensors Input Interface Block Diagram as Sub VI

d) Pitch Trim Position Sensor Input Interface

The implementation of the Pitch Trim Position Sensor Input Interface model with reference to the given specifications in III.D, is as shown in Figure 8. The range of the value sensed by the Pitch Trim Position sensor is being validated and the array of inputs is built. Both the left and the right channel sensors are combined as an individual channel component for computation of analog

data, fed to the Analog Input Processing and Data Sorting Sub VI loop.

In case the pitch trim position input is invalid then the data set is computed for the output data and latched to the previous/current/last previous valid data. The valid flag is set to FALSE and a suitable warning shall be generated for the same. In this case of pitch trim sensor invalid data, the pilot switched over to the manual mode from the automated mode. In case if the data is valid, then the valid flag is set to TRUE and the output data is computed based on the average of current, previous and last previous valued data from the array, and latched to it.

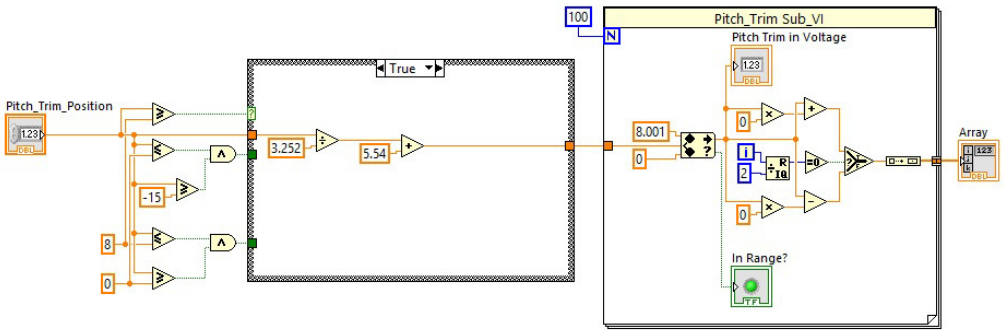

Figure 8: Pitch Trim Position Sensor Input Interface Block Diagram as Sub VI

e) Torque Pressure Transducer Sensor Input Interface

The implementation of the Torque Pressure Transducer Sensor Input Interface model with reference to the given specifications in III.D, is as shown in Figure 9. The range of the value sensed by the Torque Pressure Transducer sensor is being validated and the array of inputs is built. Both the left and the right channel sensors are combined as an individual channel component for computation of analog data, fed to the Analog Input Processing and Data Sorting Sub VI loop.

In case of left engine torque as sensed by the Torque Pressure Transducer sensor, is invalid, and then the output data is computed by the Analog Input Processing algorithm and is latched to the previous/current/last previous valid data. Similarly, in case of right engine torque sensed by the right channel as invalid, the output data is computed and is latched to the previous/current/last previous valid data. In both the cases the valid flag is set to FALSE, and the right/left engine torque is not considered for the takeoff warning computations by SWS/AIC. In case if the data is valid, then the valid flag is set to TRUE and the output data is computed based on the average of current, previous and last previous valued data from the array, and latched to it.

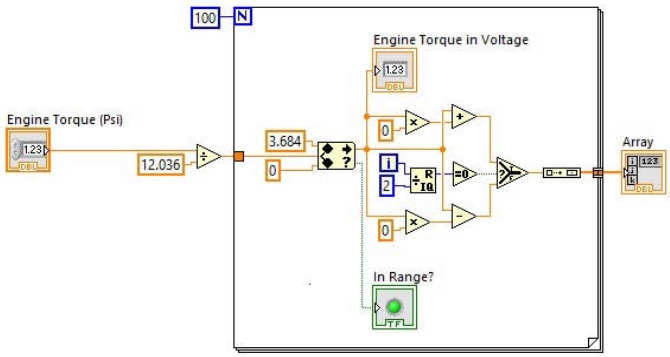

Figure 9: Torque Pressure Transducer Sensor Input Interface Block Diagram as Sub VI

V. Simulation and Analytical Results

The actual implementation of the Analog Input Processing application software is modeled using NI LabVIEW. The input data, is as per the values as given below in Table 1. The static Hardware-Software Integration (HSI) segregated test data is fed into the models respectively and is being kept in loop to create a semi dynamic nature of inputs to the system.

Table 1: Analog Inputs Data/ Test Conditions

Sl No	Signal Name	Physical Range	Scale Factor	Bias	Accuracy	Resolution	Refresh Rate
1.	RawAOALA	-10 to +40°	4.8216 deg/V	2.074V	±0.0152 V	0.0224 deg	once/ 25 ms cycle
2.	RawAOALB*	-10 to +40	4.8216 deg/V	2.074V	±0.0152 V	0.0224 deg	once/ 25 ms cycle
3.	RawAOARA	_10 to +40	4.8721 deg/V	2.0525V	±0.0152 V	0.0224 deg	once/ 25 ms cycle
4.	RawAOA RB*	-10 to +40	4.8721 deg/V	2.0525V	±0.0152 V	0.0224 deg	once/ 25 ms cycle
5.	Fuel Remaining left (LftFuelTank)	0-1000 kg for	200kg/V	-	±0.0152 V	1.016 kg	once/ 25 ms cycle
6.	Fuel Remaining right(RgtFuelTank)	0-1000 kg for	200kg/V	-	±0.0152 V	1.016 kg	once / 25 ms cycle
7.	Left Engine Torque (EngTorqueL)	0 – 44.34psig	12.035 psig/V	-	±0.0152 V	0.061 psig	once / 25ms cycle
8.	Right Engine Torque (EngTorqueR)	0 – 44.34psig	12.035 psig/V	-	±0.0152 V	0.061 psig	once / 25 ms cycle
9.	Pitch trim position (PtchTrimPos)	-15 to +8 deg.	5.3097∱ V	5.54V	±0.0152 V	0.0269 deg	once / 25 ms cycle
10.	Hydraulic pressure (HydPress)	0- 4000 PSI	800 PSI / V	0.25 V	±0.0152 V	4.05 psi	once / 25 ms cycle

The inputs are being input to the Analog Input Processing application software algorithm with the help of their respective sensor interface Sub VIs. The input data is up to a maximum of 100 index values and the constant valid data is being fed into the sub-systems. A suitable error vale (in terms of %), computed on the full scale input range is inserted at suitable index in the input array. Their equivalent DC voltage values are recorded as array values/elements and this is done dynamically, i.e. during the simulation. The system is being simulated for a particular Hydraulic Pressure value as shown in Figure 10 or Fuel Remaining as shown in Figure 11or AOA as shown in Figure 12 or Pitch Trim position as shown in Figure 13 or Engine Torque as shown in Figure 14. The Error On/Off control, regulates the control of insertion of error value to the specified hydraulic pressure/pitch trim position/fuel remaining/AOA/engine torque pressure value.

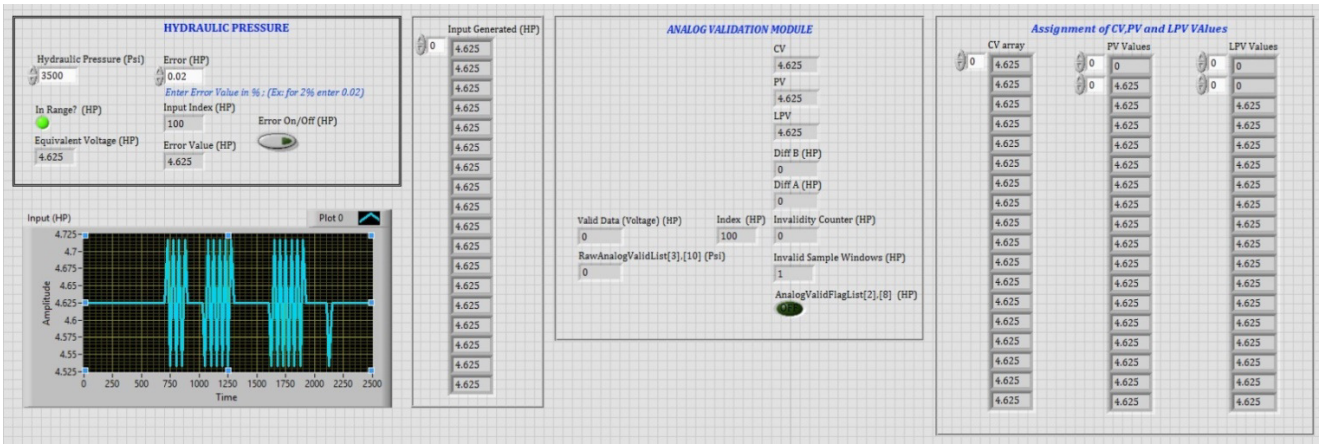

Figure 10: Hydraulic Pressure Sensor Input Interface Analysis with +/-2% tolerance band

Here in order to justify the analog input processing algorithm functionality for hydraulic pressure sensor inputs, the hydraulic pressure of 3500psi is considered (within the specified range as given in Table 1). An error value of +/-2% of 3500psi (~= 4.625V DC) is considered, at suitable time intervals (as shown in the simulation window of Input(HP)). An input array values with 100 elements is recorded as the input array, which is fed into the Analog Input Processing application software algorithm and Data Sorting algorithm loops to validate the data and sort the data as Current Valued,

Previous Valued and Last Previous Valued array, as shown in Figure 10.

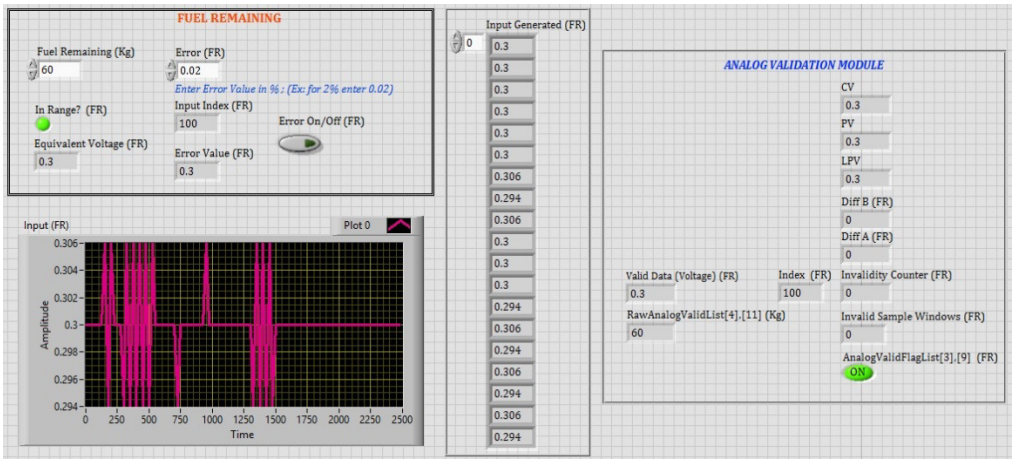

Figure 11: Fuel Remaining Input Interface Analysis with +/-2% tolerance band

The Analog Input Processing application software algorithm validates al the 100 elements in the input array and sorts them accordingly as Current Valued, Previous Valued and Last Previous Valued array. The magnitude of the differences, |Diff A| and |Diff B| are computed and the invalidity counter counts the number of invalid samples in the array correlating the magnitude of their differences correspondingly. This in-turn sets the AnalogValidFlagList to either TRUE or FALSE. Here the RawAnalogValidList data corresponds to the engineering converted value. The valid flag is set to FALSE as there is invalidity in the sensed data set with the Invalid Sample Window count set to 1.

Similarly for Fuel Remaining the data input of 60kg of fuel is considered. An error value of +/-2% of 60 kg (~= 0.3V DC) is inserted suitably in the input array of 100 elements at certain time intervals, for 25ms as each sample. The process is repeated as specified in the above section for Hydraulic Pressure Sensor Input Interface. Here it can be observed that the AnalogValidFlagList is set to TRUE as there is a valid data and the Invalid Sample Window count is 0.

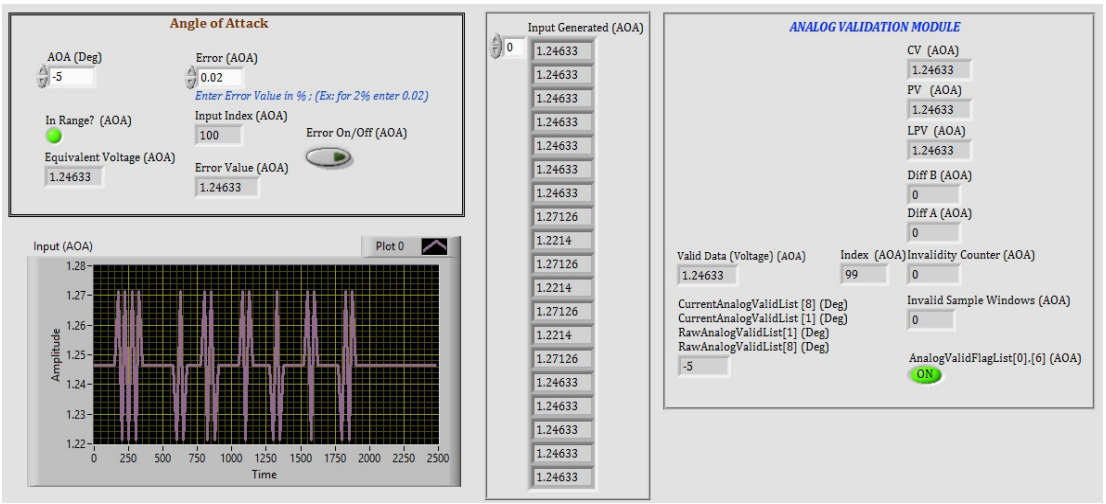

Figure 12: Angle-Of-Attack Sensors Interface Analysis with +/-2% tolerance band

For Angle-Of-Attack, the input value considered is -5deg, which is within the specified limits for the AOA analog input data. An error value of +/-2% of -5deg (1.24633V DC) is inserted into the input array for Analog Input Processing and Data Sorting VI to validate and sort the data. Here the AnalogValidFlagList is set TRUE, as there exists a magnitude of difference values that is well within the specified tolerance bands. The Invalid Sample Window count is 0.

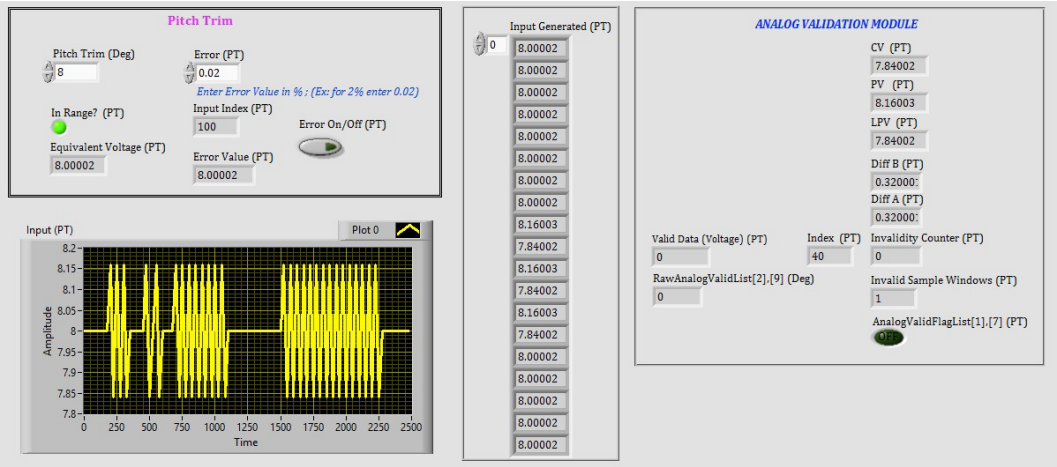

Figure 13: Pitch Trim Position Sensor Input Interface Analysis with +/-2% tolerance band

A maximum input value of 8deg is considered as input for the Pitch Trim Position sensor input interface. An error value of +/-2% of the maximum full range scale (~= 8.00002V DC) is considered and inserted into the array of 100 input elements, for further processing and sorting. During the simulation, as can be seen in Figure 13, at the index value of 40, there exists an invalidity of the samples. This is being analyzed and the AnalogValidFlagList is set to FALSE. The Invalid Sample Window count is set to 1, as there exists an invalidity, which can be inferred from the magnitude of the difference values (i.e. 0.320002) in this case at that time period is way above the priori specified tolerance value. The RawAnalogValidList corresponding to the output data is set to 0 and will be replaced with the valid Current/Previous./Last Previous value from the sorted input array upon the completion of the validation process by the Analog Input Processing application software algorithm

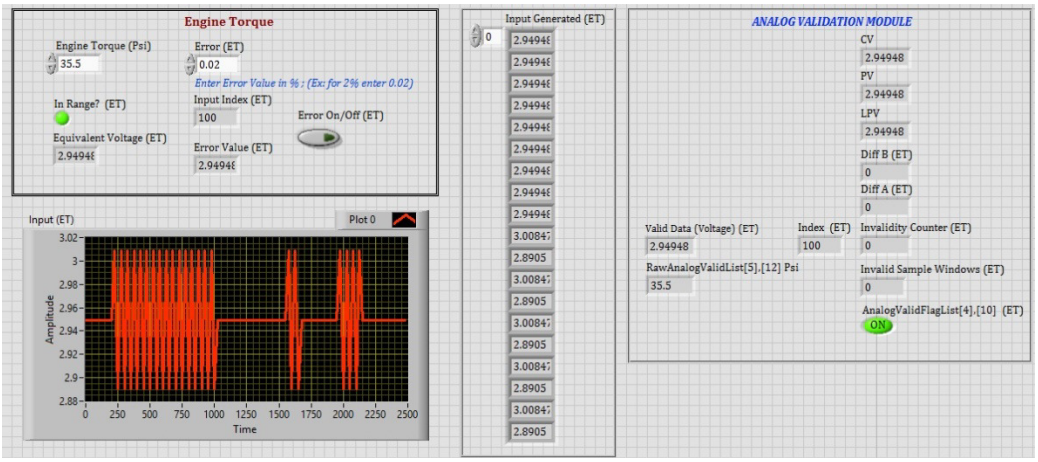

Figure 14: Torque Pressure Transducer Sensor Input Interface Analysis with +/-2% tolerance band

Similarly for the Torque Pressure Transducer Sensor input interface, an input value corresponding to 35.5psig of engine torque is considered. An error value of +/-2% of 35.5psig (~=2.94948V DC) is inserted at suitable time intervals into the input array computation. At the end of the computation by the processing algorithm for the application software, it can be seen that the valid data corresponds to as value of 2.94948V DC with the RawAnalogValidList (output data) to 35.5psig. The AnalogValidFlagList is set to TRUE as there is no invalidity processed by the processing algorithm. Also the Invalid Sample Windows count is 0.

VI. Conclusion & Future Scope

This algorithm considers the stabilization factor for the analog signals and provides suitable warnings in case of invalidity. The ambiguous nature of the analog signals with their characteristic features like amplitude, frequency, and phase may lead to uncertainty over the data being valid or invalid. This may lead to the generation of spurious and unwanted warnings during flight or on-ground testing. With MBE the signals and their validation algorithm such as Analog Input Processing application software algorithm is modeled

mathematically that proves the correctness and effectiveness of the algorithm and the need for such an approach. The objective of achieving a high reliable processing algorithm, with more complexity was successfully met.

In this paper we have proposed and implemented a novel MBE approach with the help of NI LabVIEW tool suite, with the Analog Input Processing software algorithm as a case study. The results obtained from the analyses of these models were reliable, versatile and also suitably substantiated in the removal of vexation. Validation of the implementation against the system specified requirements and the analysis of the output data against the test cases obtained during the conventional test approaches were compared and no deviations were observed. This also helped in addressing the unrealistic and unreliable situations early during the development phase, thereby reducing the overall work around cost in fixing the bug/error.

VII. Acknowledgements

The authors would like to thank the Director of CSIR-NAL, Bengaluru, for supporting this work.

References Références Referencias

1. Jost B, Ketterl M, Budde R, Leimbach T, “Graphical programming environments for educational robots: Open roberta-yet another one?”, InMultimedia (ISM), 2014 IEEE International Symposium on 2014 Dec 10 (pp. 381- 386). IEEE.
2. Huber F, Molterer S, Rausch A, Schätz B, Sihling M, Slotosch O, “Tool Supported Specification and Simulation of Distributed Systems”, Inpdse 1998 Apr 20 (p. 155).
3. Philipps J, Slotosch O, “The quest for correct systems: Model checking of diagrams and datatypes”, InSoftware Engineering Conference, 1999.(APSEC'99) Proceedings. Sixth Asia Pacific 1999 (pp. 449-458). IEEE.
4. M. Nanda, J. Jayanthi, S. Rao, "Novel Validation Algorithms for Safety Critical Embedded Software," 2008 3rd IET International Conference on System Safety, Birmingham, 2008, (pp. 1-6). doi:10.1049/cp:20080731.
5. Dr. Manju Nanda, G K Singh, K P Srikanth, “Software Requirements Data Document for SARAS SWS/AIC System”, CSIR-National Aerospace Laboratories, Bengaluru, Dec 2015, (pp. 30-49).
6. RTCA, “Software Considerations in Airborne Systems and Equipment Certification,” December 2011, http://www.rtca.org/
7. Gulan S, Harnisch J, Johr S, Kretschmer R, Rieger S, Zalman R. Model-Based Analysis for Safety Critical Software. InInternational Conference on Computer Safety, Reliability, and Security 2015 Sep 22 (pp. 111-120). Springer International Publishing.
8. Feiler PH. Model-based validation of safety-critical embedded systems. InAerospace Conference, 2010 IEEE 2010 Mar 6 (pp. 1-10). IEEE.
9. Bringmann E, Krämer A. Model-based testing of automotive systems. In2008 1st International Conference on Software Testing, Verification, and Validation 2008 Apr 9 (pp. 485-493). IEEE.
10. Blotz, A., Huber, F., Lötzbeyer, H., Pretschner, A., Slotosch, O., Zängerl, H.P., “Model-based software engineering and Ada: Synergy for the development of safety-critical systems”, Ada, Deutschland 2002 (2002).
11. Y. C. Yeh, "Safety critical avionics for the 777 primary flight controls system," 20th DASC. 20th Digital Avionics Systems Conference (Cat. No.01CH37219), Daytona Beach, FL, 2001, pp. 1C2/1-1C2/11 vol.1. doi: 10.1109/DASC.2001.963311
12. A. Ray and R. Luck, "An introduction to sensor signal validation in redundant measurement systems," in IEEE Control Systems, vol. 11, no. 2, pp. 44-49, Feb. 1991. doi: 10.1109/37.67675
13. Erbay, Ali Seyfettin, "A PC-Based Signal Validation System for Nuclear Power Plants. " Master's Thesis, University of Tennessee, 1994. http://trace.tennessee.edu/utk_gradthes/2583.