RESEARCH ARTICLE OPEN ACCESS

Functional Level Data Acquisition Requirement Specification Formulation for Embedded Systems: Challenges, Experiences and Guidelines

Ambalal V. Patel
Integrated Flight Control Systems (IFCS) Directorate, Aeronautical Development Agency (Ministry of Defence, Govt. of India), P.B. 1718, Vimanapura Post, Bangalore – 560017, India
Corresponding Author: Ambalal V. Patel

ABSTRACT
With the advancement in technology, many features of the large scale application systems are being embedded and miniaturized. This has led to investment of significant efforts on testing and verification of the developed embedded system. The data / signals acquired at intermediate stages of the system aids in analysis and debugging in accelerated manner. This article presents some guidelines for formulating the data acquisition / test points requirements of the core functional module (algorithms) and relevant aspects which are required to be considered during the development to cater for testing, evaluation and subsequent expansion or up-gradation of the system. The guidelines are proposed based on the experience gained over a period of time while working on safety critical embedded systems. The relevant experiences are also shared with the help of a few illustrative examples.
Keyword: data acquisition, algorithms, embedded systems, requirements, specifications

---

Date of Submission: 02-10-2017 Date of acceptance: 31-10-2017

I. INTRODUCTION

With the advancement in technology, many features of the large scale application systems are being embedded and miniaturized. This has led to put significant efforts on testing and verification of the developed system. The data / signals acquired at intermediate stages of the system aids in analysis and debugging in accelerated manner.

Significant efforts are put to test and analyze the safety critical embedded applications involving electronics hardware and software, and interfaces with the other embedded subsystems and mechanical systems. Examples of such systems are fly-by-wire flight control system, guidance and navigation systems for missiles, space vehicles, automobile applications etc.

For thorough evaluation of such critical systems, all inputs, outputs and intermediate states need to be analyzed extensively and thus are required to be recorded onboard during the operation. The inputs are usually the sensor measurements which indicate the status of the plant while outputs are the commands to drive the actuation system of the plant. The intermediate states are part of the software redundancy management and computationally intensive sophisticated application algorithms such as control laws, air data algorithms, autopilot control laws, weapon system algorithms etc. of the flight control system as shown in Figure 1. Efforts required and philosophy followed for testing and evaluation of fly-by-wire safety critical flight control system of high performance fighter aircraft have been given in Ref. [1-4]. Guidelines for acquiring the data based on the automotive crash events [6], and their investigation are given in Ref. [5]. Global Laboratory Practice (GLP) for acquisition and processing of electronic raw data are presented in Ref. [7]. Although computerized systems are used widely, there is still a need for guidance on how to apply the GLP principles to such systems with regard to electronic raw data [7]. Earlier good attempts were made to present the set of guidelines for formulation of requirement specifications for functional modules of embedded systems in Ref. [8].

This article presents detailed guidelines for incorporating the test points (signals to be tapped out for optimized system analysis) in the core functional requirements module (algorithms) and relevant aspects which are required to be considered during the development to cater for evaluation, testing and subsequent expansion or upgradation of the system. The guidelines are proposed based on the experience gained over a period of time. The relevant experiences are also shared with the help of a few illustrative examples.

The article is organized as given below. After introduction, Section 2 presents generic background of the embedded systems involving software and hardware, Section 3 presents the various constraints of the on-board embedded systems which are usually encountered. Some of the experiences gained over a period of time have been shared with the help illustrative examples in Section 4. Section 5 presents the guidelines for formulating the requirements for acquiring the data / intermediate states of the software to be implemented. Section 6 presents additional remarks on the data post processing challenges, and Section 7 concludes the paper.

II. EMBEDDED SYSTEMS

The algorithms functioning in the form of large scale software involve several nonlinear and dynamic elements, and intensive logic for reconfiguration (failure detection, isolation, selection). The logic itself is dependent on the several other discrete conditions which may be external due to user demand or the system failure or internally generated by the system due to satisfying certain criteria during the operation as intended.

The nonlinear elements such as nonlinear functions or lookup tables themselves may be fixed or reconfigurable depending upon the operating point of the plant or failure in the system. The dynamic elements involve faders (used for smooth transition of the signal from present value to the desired value over a specified time whenever reconfiguration takes place on occurrence of a specific event), filters (used for smoothening the signal within a range of frequency of interest), persistency counters (used for counting the elapsed time on occurrence of an event) etc. The analysis of the data of evaluation of multi-input multi-output system involving such elements and their interdependencies becomes very complex, See Figure 3. Over and above, the noise due to electronics hardware also adds some uncertainty on the exact matching of the data with the expected value. It may be noted that the ground test rigs used for evaluation of such safety critical systems have their own additional hardware to facilitate the data acquisition at various intermediate stages which have their inherent characteristics and these are required to be understood thoroughly. In the analysis, from the acquired data, it is required to identify and distinguish the effects of on-ground test rig and onboard system hardware characteristics on the overall clearance of the implemented system.

After the entire embedded system (including the hardware and software) is cleared for initial onboard evaluation during prototype development, there could be more number of iterations for the software upgradation to embed all the functional requirements gradually in a step by step manner. These new requirements need to be assessed thoroughly at on-ground test platforms. The evaluation at on-ground hardware-in-loop test rig is classified as ‘black box’ testing. In the ‘black box’ testing one does not have an idea of how the functional requirements are written / coded but only based on the available knowledge of the functional requirements, the testing needs to be carried out and results to be assessed.

Therefore, capturing the appropriate intermediate states / signals of the complex, logic intensive algorithms is very important in order to interpret the results, understand their cross-correlations, effects of on-board and on-ground test rig hardware which ultimately helps in clearance of the implemented software in an accelerated manner.

Therefore, the system designer must understand the test setup and associated features apart from the constraints of the onboard hardware. Based on relevant knowledge, one should incorporate the required test points (also called intermediate states which are to be acquired during the testing either on-ground or on-board) from the algorithms residing in the form of software in the on board computer.

III. CONSTRAINTS OF THE ON-BOARD SYSTEMS

With available hardware architecture of the embedded system, the increase in software functionality leads to the constraints on the execution time, i.e., the specific computations of the algorithms should be completed within the specified time frame. In case of the safety critical fly-by-wire system of the high performance combat aircraft, these constraints play a critical role. The onboard software has to play a crucial role of sending the data on the multiple data recording devices while doing the complex and voluminous computations in real time involving several tasks of which a few are mentioned below (as an example for flight control system):

- Corrections to the sensor measurements and redundancy management (failure detection, isolation and selection),
- Control command computations (Primary control laws and Autopilot) and reconfiguration under normal and all failure scenarios,
- Communication / data transfer with the other subsystems (between Flight Control System and Avionics, Engine Control, Hydraulic System etc.)
- Synchronization of the data across multiple channels and received from multiple subsystems involving inherent time delays,

Automatic flight control system generic controller block schematic is shown in Figure 2, which is complex and computational intensive involving several reconfigurations for working under normal and all failure modes. The data receiving and transmission also takes a countable fractional time with the increase in the number of signals to be acquired and affects the overall execution time of the software. Therefore, tasks to be performed by the software need to be prioritized. It is obvious to prioritize all the tasks of real time computations and command generation over data acquisition. However, data logging is equally important for the post operation (flight) analysis, clearance of next activity (flight), and envelope expansion during development phase, especially. The data to be acquired involves various categories of signals with their inherent characteristics and associated facts as given below:

- Analog signals: They inherently contain the system noise,
- Digital signals: They are output of the hardware (Analog to Digital converted output) or software and have the Least Significant Bit (LSB) variation. The LSB variation appears as spikes and it depend on the resolution of the signal,
- Discrete signals: They indicate the health / status of the hardware elements or reconfiguration / failure occurred within the software during operation. They are binary signals indicating either 1 (True) or 0 (False).

IV. EXPERIENCES GAINED OVER A PERIOD OF TIME AND LESSONS LEARNT

Flight data acquired in the recorder is used in development phase (clearance of the next flight of aircraft or phase of testing) as well as in final operations / services for preventive maintenance, and feedback to the design house for improvement of the product. It is an important part of any product life cycle. Therefore, for the hardware and software development, especially for the functional / algorithm level development, robustness, reliability, consistency, interpretability, analyzability of the recorded data is very essential and it needs to be taken into consideration while developing a safety critical product. A significant work has been carried out on the data attributes, data quality and metrics [9].

This section presents a few experiences gained over a period of time and lessons learnt during the design, development and evaluation of fly-by-wire flight control system of high performance fighter aircraft [1-4]. These examples are based on the black box testing (Hardware in loop testing and analysis):

1) Attempts were made to consider set of minimum and maximum values of all inputs by forming a multi-dimensional hypercube for all reconfigurations to compute the local and global minima and maxima to decide the test point ranges. This technique may work for linear systems (linear algorithms) only. However, most of the real world application algorithms are nonlinear and hence it does not guarantee achieving of local and global minima and maxima by considering the minimum and maximum range of the inputs to the multi-dimensional hypercube. Section 5 on guidelines provides the resolution to this problem.
2) Illusion on fader implementation being incorrect due the range of the signal acquired being narrower than actually required. In real time on-board software, the faders were working correctly, however, due to clipping of the recorded signal due to narrower range than the actual required, it appeared that the fader time not implemented correctly in the on-board software. Down the line test points were found to be matching correctly, which led to infer that the fader functionality is correctly implemented and the issue is with the range of the signal being recorded was not specified correctly. Refer to Figure 4 for illustration.
3) Higher rate signal discretes missing in the recorded data: The procedures to compute the discrete were executed at 20Hz while the discrete signals were recorded at 10Hz. The effect of setting of one discrete for a one frame was seen in the final outputs but the corresponding toggled status was not recorded at 10Hz. See, Figure 5 for illustration.
4) The random noise in the system affects the computations, for example the output of the washout filter (high pass filter) ideally becomes zero at steady state in the absence of noise. However, random noise makes it non zero leading to infer the results are wrong. Therefore, appropriate measures were required to account for this fact to define the pass / fail criteria and the suitable range of such test points. See Figure 6 for illustration.

V. GUIDELINES FOR FORMULATION OF REQUIREMENTS FOR ACQUIRING THE DATA / INTERMEDIATE STATES OF THE SOFTWARE TO BE IMPLEMENTED

The guidelines given below are generic and are based on the experience with evaluation of the safety critical fly-by-wire flight control system and applicable for most of the embedded systems. They may get further enriched by including the experiences and lessons learnt from other systems.

1) Optimize the number of signals to be acquired for which:

- Identify all inputs, outputs of the functional module including their validity status which are indicated by discrete signals.
- Identify the intermediate signals of the functional module including their validity status / failures etc. to be recorded. Keep in view the execution

time constraints and the system analyzability and accordingly maintain the required number of test points.

2) Signals after the dynamic elements may be preferably acquired as solving the recursive computations offline is usually cumbersome. Refer to Figure 7 for illustration.
3) Find out the suitable locations of the signals to be tapped from the diagrams such that the acquired data aids in quick interpretation like:
- With optimal efforts, the signals in the adjacent areas are computed,
- Differentiate the results due to change in configuration or inputs.
4) Keep the range of all signals wider than their actual operating range in order to avoid saturation while maintaining the suitable resolution. The operating range of the inputs and outputs are easy to determine as they are usually a function of the sensor operating limits or output hardware driving limits. However, following care needs to be taken to determine the range of variation of several of the intermediate signals that are to be acquired:
- For the signals to be acquired after fader or multi-input single-output switch, consider the widest range of operation among all input signals to the switch with extra margin to avoid saturation. Refer to Figure 9 for illustrattion.
- Analytical way to compute the range of the signals: If the design of the functional elements have consistency, for example if the algorithm requirement about the nonlinear function is such that "nonlinear functions or the lookup tables shall not be extrapolated for out of range inputs and they shall be frozen to the limits", then the range of the output of such functional elements are known a priori, irrespective of their input value. Therefore, the range of the test point after the nonlinear element whose output range is known can be computed quickly and one does not have to consider all the computations from inputs of the that path till the test point. Refer to Figure 8 for iilustration,
- Trade-off between wide range and resolution can be minimized by incorporating multiple test points in lieu of single by splitting up the operation of the nonlinear element. However, it calls for a change in the functional module and increase in the number of test points. Therefore, priority between data acquisition and execution of algorithms to be resolved, appropriately,
- Adapting dynamic range philosophy, i.e., change the acquisition range of the signal if possible on occurrence of reconfiguration within the algorithms to balance the trade-off between saturation and resolution.

5) Keep consistency in the units of the acquired signal (such as Normal Acceleration in g's or m/sec^2, and angles in degree or pi radians – a normalized signal for consistency).
6) Discretes / flags / status / events of the signals are represented by a single bit which can be combined in words. Continuous signals may be discretized meaningfully and combined into words which aids in accommodating more number of signals for acquisition.
7) Rate of acquisition: Prioritize the rate of signal acquisition (raw to corrected / voted, intermediate signals, final free stream sensor output or control law commands / plant inputs)
- Discretes may be acquired at the least rate as their status may be inferred for in between frames of acquisition based on the other dependent signals,
- Inputs and outputs may be acquired at the highest possible rate, as they are useful to reproduce all other signals by replaying through the offline model,
- Intermediate signals may be acquired at a rate in between the highest and lowest possible available acquisition rates
- Categories of the signals and their inherent characteristics to be accounted for deciding the rate of acquisition: Analog signal (inherent system noise), digital signals (A/D conversion and LSB spikes), discrete signals / flags / events (point of transition of the signal and interpretation of their cause and effects on other parameters, especially edge & level trigger effects), delays in the signals and their time synchronization requirements
8) Offloading parameters / test points by repeating the computation procedure offline while doing data processing: Find the set of points or signals which can be recomputed offline with minimal efforts and time while doing the data decoding and post processing. It demands repeating the onboard computations offline by using the recorded input signals required for that particular process. Thus, one has to ensure that at least all required inputs are recorded. A better way to do this is to let the recorded data be replayed through the offline simulation model of the functional module. This helps to compute all test points or signals. However, limitation with this procedure is that the data is computed for the single channel representation. If onboard real time system has multiple channels, then the information about the channel failure also required to be accounted while doing offline computations. Thus, it requires modeling of the on board channel failure redundancy. This entire process therefore demands verification and validation of this offline computation process.

Thus, one has to account for the cost and efforts required versus the essentiality of recording the required signals and outcome thereof. Refer to Figure 10.

9) Distribute the data acquisition across multiple channels judiciously such that in case of a specific level of channel failures, one should be able to reconstruct the signals required for analysis.
10) On ground level test rigs:

- Magnitude wise pass/fail criteria (Magnitude tolerance) include ±LSB variation computed based on the number of bits and range of the acquired signal over and above the actual system level component's end-to-end tolerances,
- Time wise pass/fail criterion (Time tolerance) is required for verification of fader operation and persistence time counting. It is computed as given below based on the corresponding discrete signal:

  Time Tolerance (sec) = [Lower bound (sec), Upper bound (sec)], where

  Lower bound = (Expected time (sec) - 2*signal sampling time (sec))

  Upper bound = (Expected time (sec) + 2*signal sampling time (sec)), where 2 indicate number of samples (±1 sample) at start and end of event. For example, if the persistence or fader time is 1 second and the sampling time is 0.1 second (computations at 10Hz), then Lower bound and Upper bound shall be 0.8 and 1.2 second, respectively, may be used as pass / fail criteria for the expected 1 second persistence or fader time.

11) Modeling of all the requirements helps to integrate it into the end-to-end expected value generation tool (offline simulation model embedding the pass / fail criteria), the results of which including the tolerance thereon are used for defining the pass-fail criteria and clearance of the system implemented.

It is preferred to have a checklist of all above points while arriving at the requirements for data acquisition.

VI. POST DATA PROCESSING CHALLENGES AND GUIDELINES

The real time data is usually recorded in encoded format. The decoding is done by using the offline data extraction tool based on the recorded data message structure. The decoding tool applies the necessary bias and scale factors to the raw data to convert it into an appropriate engineering unit and saves the data of several Gigabytes in ASCII format. The decoding tool embeds the offline procedures for computing the other derived signals. The data analysis includes:

1) **Visual analysis:** It includes plots of either data time history or cross plots of variables. Visual analysis also includes animated data plots (multiple cross plots showing the data history which enables to understands the trends), and thermal plots (indicating the gradients in the signal).
2) **Numerical / statistical analysis:** It includes the analysis of achieved maximum or minimum, mean, median, standard deviation etc. values of the data in a tabulated manner.

In either case, handling the large data for analysis involves challenges and requires automated tool(s) to deal with visual and numerical / statistical data analysis for quick interpretation. The automated tool should involve the followings:

Optimize the number of plots / figures to avoid memory constraints and display issues. The plots should be organized in a specific order such as inputs to outputs, cause and effects, and interdependencies of the parameters of the associated subsystems to quickly infer and conclude the results rather than going back and forth.

Time wise recorded history of the large data in ASCII format at different rates in multiple files represents it in the matrix form and analysis thereof can be carried out in either or both of the following ways:

- **Analysis by using column / row number wise data (Matrix format)**: by embedding the data recording message structure model into the automated tool. The unit of each column variable, conversion scale factors etc. can be organized one time in the automated tool and used for analysis. In such a case one has to invoke the column / row number of each signal while plotting or analyzing.
- **Analysis by using variable name:** Column / Row wise extracted signal is assigned with a specific variable name. Subsequently, variables are used for plotting and analysis. However, variable naming requires familiarity with unique nomenclature methodology, some expertise and relevant information of the system including knowledge of the engineering units and over and above remembering the variable names while doing analysis.

Whenever onboard software functionality changes and demands changes in data acquisition, an automated tool has to be updated to cater for the above mentioned techniques for analysis. This situation frequently occurs during the product development phase where functionality keeps on changing to cater for incremental development. However, during operation of final product, automated tool update and maintenance thereof is not required.

3) **Documentation:** The tool should be capable to publish the results (Figures, Tables etc.) in a specific document format wherein by least efforts of editing / commenting, results are quickly released for follow up actions.

For the data analysis tool, the following documents must be maintained for ease of updates and usability:

a) **Tool development procedure document:** The tool update is required once in a while due to data acquisition requirement changes. This document helps in quickly refreshing the flow of the files and the locations / files to be updated in a systematic manner. This document must be kept on updating as and when a new experience has come across or procedure change occurs in the tool development.

b) **User manual:** As the name suggests, this document helps the user to adapt the tool easily for the required purpose. This updated document incorporating the latest changes must be made available along with the updated tool that is being released.

4) **Configuration of the data analysis tools:** Since automated tools are required to be updated frequently during development phase, it is worthwhile to follow a systematic configuration procedure. Each process or part of the process in the automated tool should be commented to enable ease of maintenance. Thus, the data extraction tools, offline data processing tools are to be treated on par with the onboard application software.

VII. CONCLUSIONS

In this paper attempt has been made to provide a systematic approach for formulating the requirements for acquiring the data for analysis of the core functional modules of the embedded system. The guidelines for formulating those requirements have been arrived based on the experience while working with the evaluation of a safety critical fly-by-wire flight control system. A checklist incorporating the points mentioned in this paper may be prepared and followed while arriving at the requirements for data acquisition. These guidelines are expected to be applicable for most of the embedded systems and may get enriched further by the experiences and lessons learnt from other systems.

VIII. ACKNOWLEDGMENTS

The author is grateful to the Aeronautical Development Agency (ADA), Bangalore, India for permitting to publish this work.

REFERENCES

[1] Ambalal V. Patel, Vijay V. Patel, Girish S. Deodhare, and Shyam Chetty, “Clearance of Flight-Control-System Software with Hardware-in-Loop Test Platform”, AIAA Journal of Aircraft, Vol. 51, No. 3, May-June 2014, DOI 10.2514/1.C032404.

[2] Ambalal V. Patel, Vijay V. Patel, Girish Deodhare and Shyam Chetty, “Flight Control System cleareance using dynamic tests at Hardware-In-Loop Test Platform”, Proceedings of International Conference on Avionics Systems (ICAS) 2008, held at RCI, Hyderabad, during February 22-23, 2008.

[3] Guruganesh R., Shyam Chetty, Ambalal V. Patel and Girish Deodhare, “Clearance of LCA Flight Control Laws on Various Ground Test Simulation Platforms”, Proceedings of International Conference on Avionics Systems (ICAS) 2008, held at RCI, Hyderabad, during February 22-23, 2008.

[4] Ambalal V. Patel, Vijay V. Patel, Girish Deodhare and Shyam Chetty, “Flight Control System cleareance using static tests at Iron Bird”, Proceedings AIAA Guidance, Navigation and Control (GNC) conference and exhibit, paper No. 6203 in session No. 31-GNC-14, held at Keystone, Colorado, USA during August 21-24, 2006.

[5] Incident Response - Data Acquisition Guidelines for Investigation Purposes1, CERT-EU Security White Paper 2012-04, Version 1.4 – May 2015

[6] Thomas C. Mercer, Keith S. Schultz, Recording Automotive Crash Event Data, International Symposium on Transportation Recorders, May 3 - 5, 1999 Arlington, Virginia

[7] Guidelines for the Acquisition and Processing of Electronic Raw Data in a GLP Environment, Working Group on Information Technology (AGIT), Release Date: 1 December 2005, Version: 01.

[8] Ambalal V. Patel, “Formulation of Functional Level Data Acquisition Requirement Specification for Embedded Systems: Challenges, Experiences and Guidelines”, UPIN: G1-073, e-proceedings, National Conference on Large Scale Multi-Disciplinary Systems of National Significance (LAMSYS-2016) held at Satish Dhawan Space Centre SHAR, Sriharikota, India during June 24-25, 2016.

[9] Li Cai and Yangyong Zhu, “The Challenges of Data Quality and Data Quality Assessment in the Big Data Era”, Data Science Journal, 14: 2, pp. 1-10, DOI: http://dx.doi.org/10.5334/dsj-2015-002.

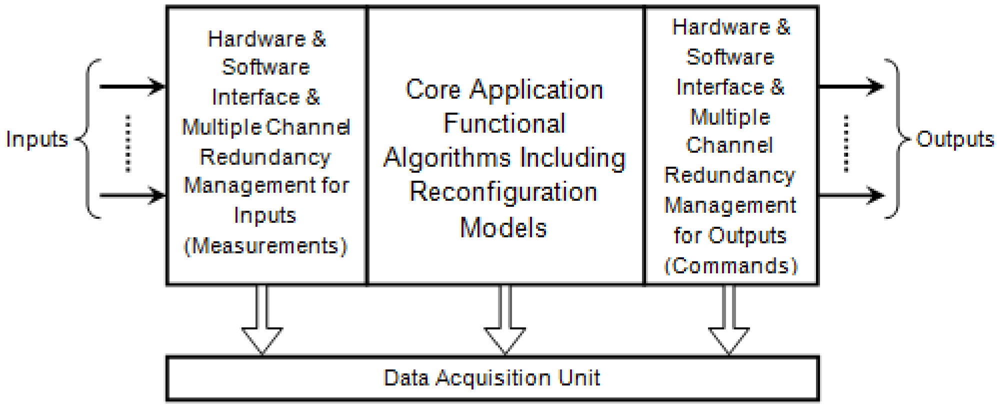


Figure 1: Embedded System & Data Acquisition Unit Schematic

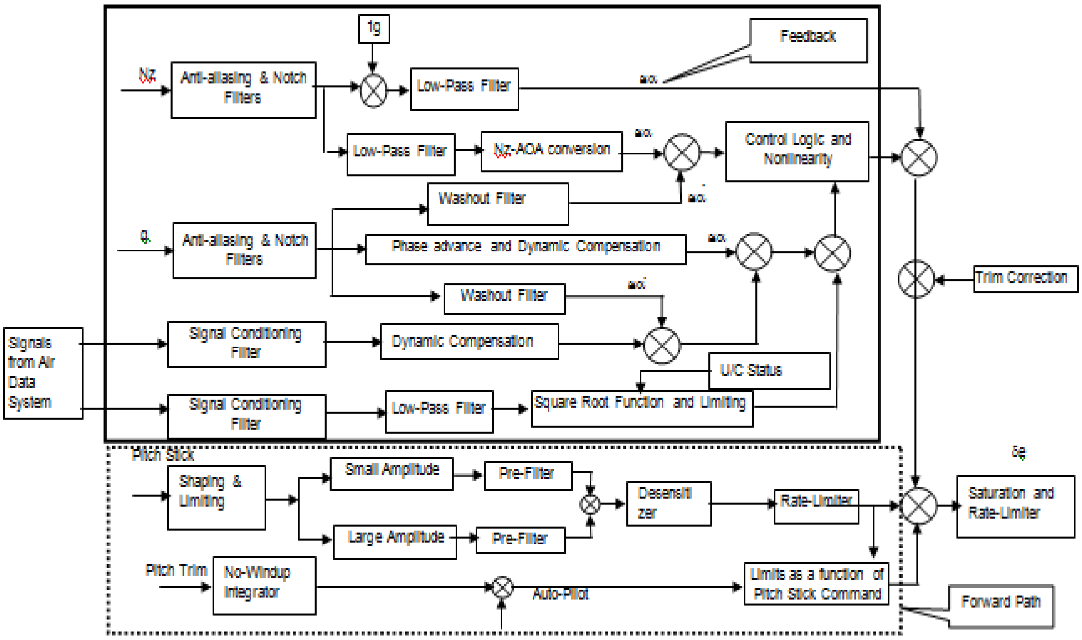


Figure 2: Automatic Flight Control System: Generic Controller Block Schematic

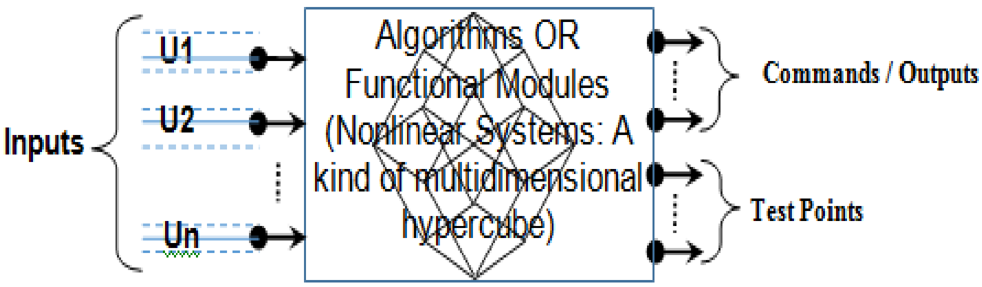


Figure 3: MIMO Nonlinear System

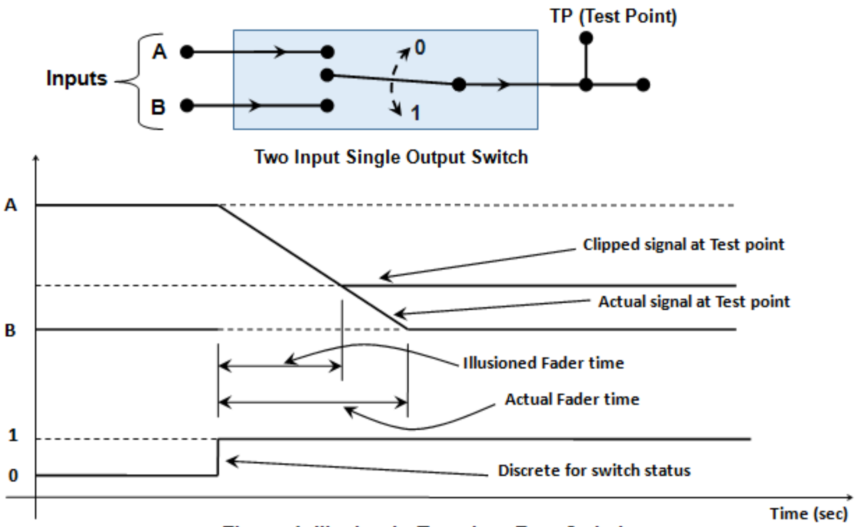


Figure 4: Illusion in Transient Free Switch

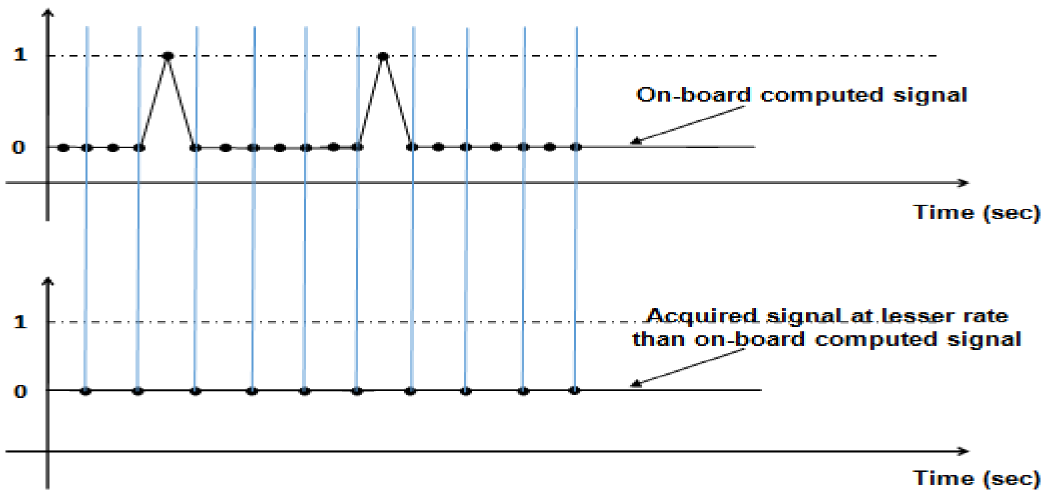


Figure 5: Higher Rate Signal Discrete Sample

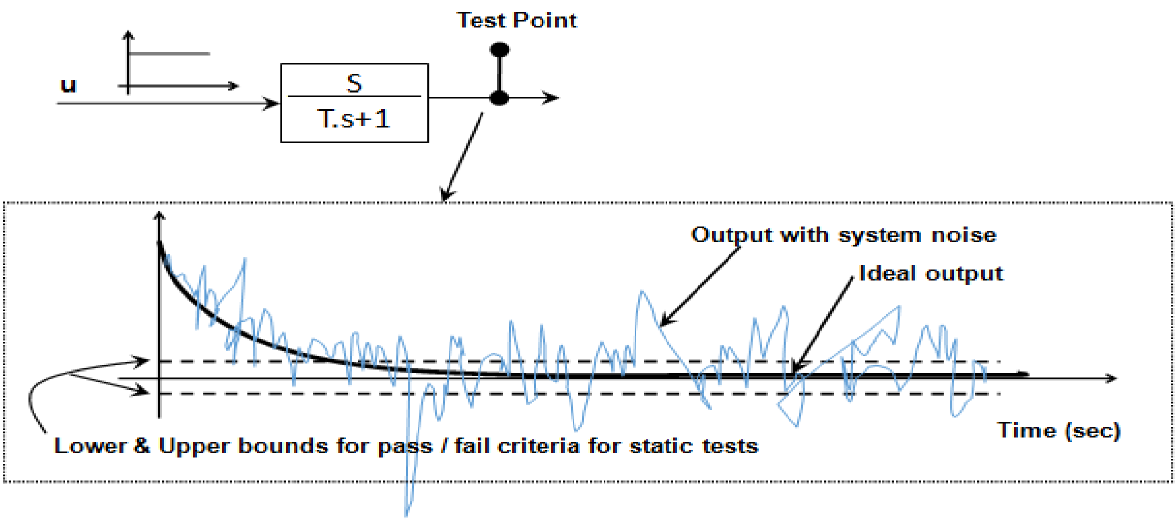
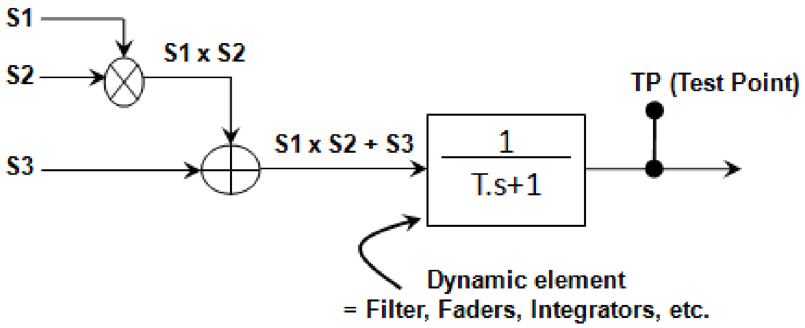
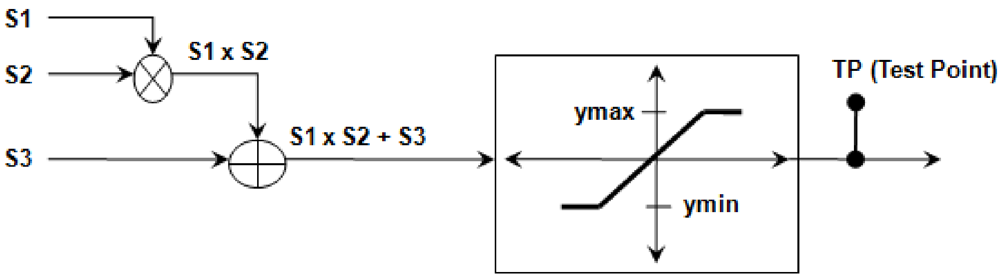


Figure 6: Noise in the system



- **Prioritize acquisition after dynamic elements, as it is complex and laborious to solve the recursive solutions offline.**

Figure 7: Prioritize acquisition after dynamic elements



If the Nonlinear function requirement is **'No extrapolation is allowed',** then

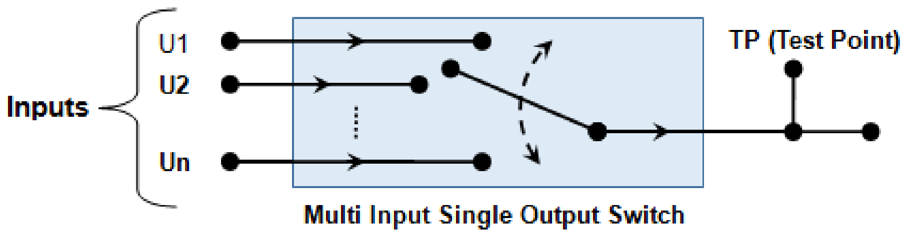
Lower Range of the TP = Ymin - Delta Margin

Upper Range of the TP = Ymax+ Delta Margin

Delta Margin = At least it should be Least Significant Bit (LSB)

Thus, ranges of the signals and computations done prior to the nonlinear element can be ignored!

Figure 8: Consistency principle of design elements

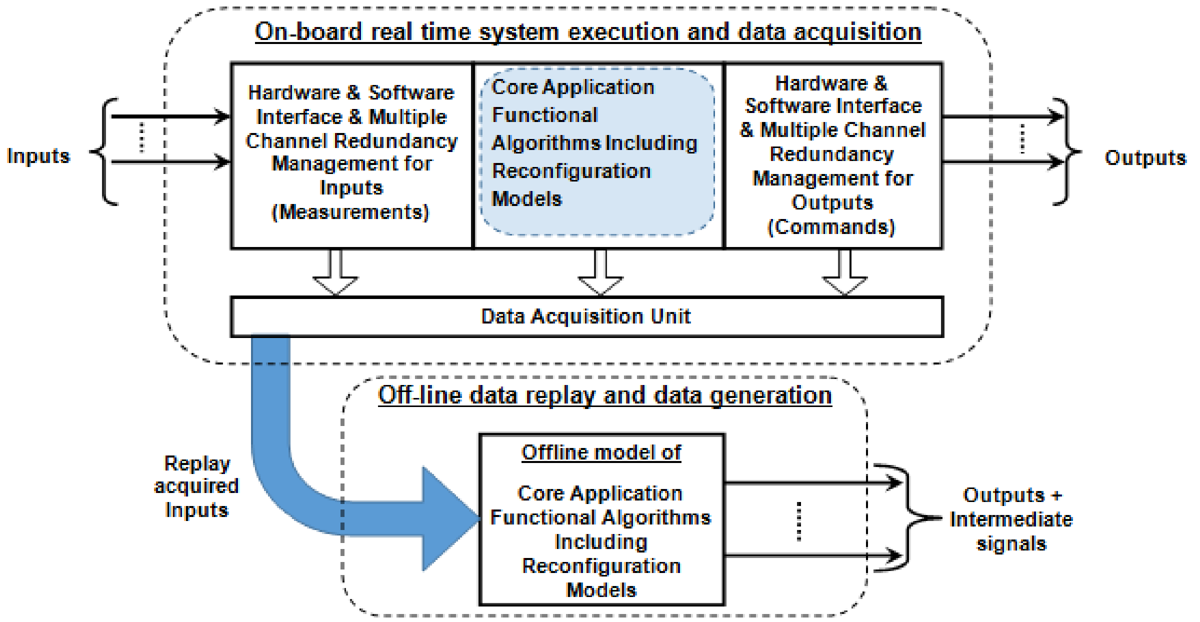


Lower Range of the TP = Lowest Range of (U1, U2, ..., Un) - Delta Margin

Upper Range of the TP = Highest Range of (U1, U2, ..., Un) + Delta Margin

Delta Margin = At least it should be Least Significant Bit (LSB)

Figure 9: Range for the Test Point at the output of the Multi Input Single Output Switch



- Offline model should be capable to replay all inputs and reproduce the intermediate signals and outputs, in case of non-recording of them.
- Multiple channel redundancy may be considered on the offline model, however outcome should be clear before investing time and efforts.

Figure 10: Offline Data Replay and data generation process

International Journal of Engineering Research and Applications (IJERA) is **UGC approved** Journal with Sl. No. 4525, Journal no. 47088. Indexed in Cross Ref, Index Copernicus (ICV 80.82), NASA, Ads, Researcher Id Thomson Reuters, DOAJ.

Ambalal V. Patel. "Functional Level Data Acquisition Requirement Specification Formulation for Embedded Systems: Challenges, Experiences and Guidelines." International Journal of Engineering Research and Applications (IJERA) , vol. 7, no. 10, 2017, pp. 75–84.