Self Learning Architecture Based Product Support Systems of Fighter Aircraft for Improved Performance

Madhusudana Rao Vanamala, Saravanan S, Vijay V Patel, and Prasad NSSRK Nalli
Aeronautical Development Agency (ADA), PB No: 1718, Vimanapura Post,
Bangalore-560 017, INDIA
Email: vmsrao@jetmail.ada.gov.in

Abstract: *The demand for improved performance of fighter aircraft is growing exponentially as threat scenario is rapidly changing. As a result, the fighter aircraft is becoming extremely complex platform with integration of state of art mission and flight critical systems along with essential systems like electrical, hydraulics, engine, fuel, brake management/monitoring and control systems.*

The need for high availability of an aircraft is becoming more essential as performance to cost ratio of an aircraft fleet is increasing. This requirement mandates to have high performance product support systems in squadron for improving the turnaround time and intern enhancing the availability of an aircraft.

The self learning or machine learning architecture based 2nd line (Intermediate Line) rugged, portable flight control system (FCS) testers were developed to meet ever demanding airforce squadron requirements. These testers have unique architecture for extensive self testing, real time and continue health monitoring, usage monitoring, monitoring of important additional parameters like shock, temperature and humidity of the test system to configure suitably at architectural level, perform all required tests and produce test results which are correct and dependable.

This paper clearly brings out architectural design aspects, dynamic reconfiguration of the system based on self test results, usage of health parameters for fast and correct decision making use of inference engine, self learning/machine learning through local as well as global (through Network) field test data and dynamic reconfiguration of test sequences for improved performance with reduced test duration and for enhanced safety. These self learning systems reduce dependency on experienced personnel for fault isolation to the great extent.

Keywords**:** *Self learning, Machine learning, System Architecture, Fighter aircraft, flight control system, mission critical, flight critical, availability, 2nd line testers, Inference Engine, Health monitoring, Dynamic reconfiguration, Test sequence, Aircraft On Ground(AOG).*

1. Introduction

Advanced electronic systems used in various applications are mostly intelligent systems having capability to perform complex tasks [1]. This is due to drastic reduction in size, weight, power, and cost of embedded systems and also exponential increase of computational power, re-configurability and high reliability.

The high performance fighter aircraft is platform having System of Systems (SoS) integration. It is a system where state of art technologies from different discipline made to interface to perform a complex mission. Subsystems or Line replaceable Units (LRUs) are based on advanced technologies and performing large number of real time tasks. Maintenance of these systems in air force squadrons has paramount importance towards ensuring maximum availability of aircrafts.

Product support is a multilevel activity ranging from in-situ testing on aircraft, squadrons lab level testing, Service bay level testing and at OEM (Original Equipment Manufacturer). AOG (Aircraft on Ground) can be minimized only with effective test and repair system in place. The most important levels of testing in this chain at squadron are 1st line and 2nd line.

Deploying more efficient and effective testers in these areas are going to improve significantly in reducing AOG time and improving maintenance efficiency with minimum spares resulting cost effectiveness. Hence, it is required to adopt suitable technologies for these product support systems to correctly and quickly identifying

faults towards reducing turnaround time of aircrafts. Self or machine learning architecture based systems will greatly contribute in these areas for improved performance with effective decision analysis [2].

2. Self Learning Architecture

Architecture is an important part of system design. It is one of the key elements which decide overall system performance. Architecture should enable system to learn incrementally and perform better with usage. The basic futures of the architecture required for product support systems are:

1. Architecture should be modular and reconfigurable.
2. Architecture should have provision for continuous monitoring and periodic generation of health/critical parameters.
3. Provision of maximum coverage of Power On Self Test (POST).
4. Monitoring and reporting of instrumented / environment data.
5. Test cases or test sequence should be modular and re-structurable.
6. Inference Engine with Fault tree analysis.
7. Provision for logging/storing of test results.

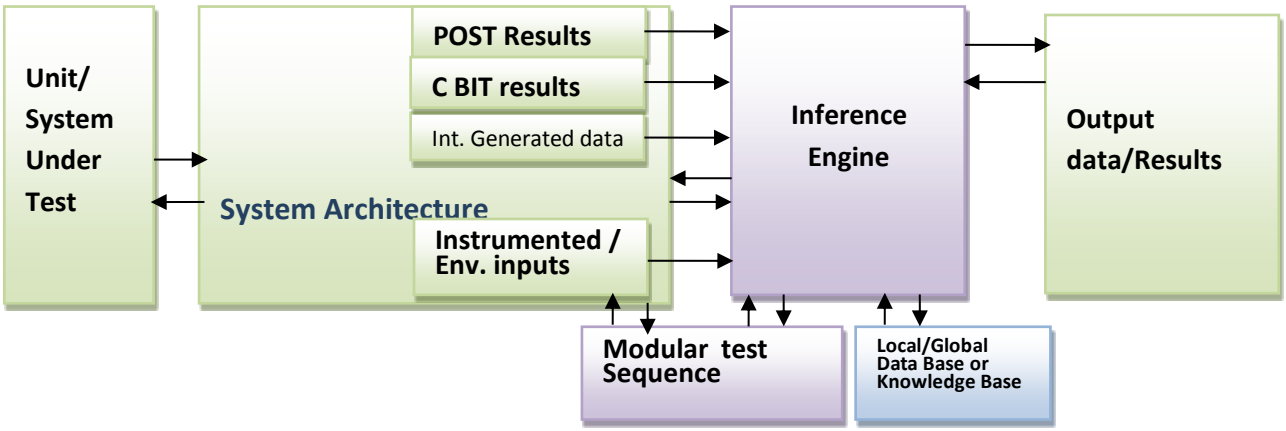


Fig 1: Block Schematic of Self Learning Architecture for 2nd Line

The following aspects need to be considered for the test system design:

1. The system architecture should provide maximum testability by design and assess complete system health.
2. Internal reference parameters (like Reference Voltages, Power supply voltages, etc.) generated/used for acquisition and processing of external inputs need to be monitored continuously.
3. External parameters affecting the system output/health should be monitored periodically.
4. As a back ground operation, system health (C-BIT) should be monitored continuously.
5. Apart from functional testing, timing test is also essential for embedded systems. Evolutionary testing enables a fully automated search for extreme execution times [3].
6. There should be provision for trend analysis of already/so far generated test data/results.
7. Efficient Inference Engine(IE) for inferring health status based on complete set of inputs for generating output results.

The block schematic of self learning architecture implemented in 2nd line (Intermediate line) testers is given in fig 1.

2.1 Sequence of Operation

Sequence of operations is given in flow diagram 1. At the top level, system exhibits two different architectures or configurations based on the type of user log in. Administrator mode, full system resources are available. In this mode of operation, apart from normal test functions, it provides features for adding or deleting users, access to data base for read, delete or modify, Access to USB interface for data/programs loading or down loading, running hardware diagnosis programs, calibration of analog circuitry, etc,. In operator mode, architecture limits to provide only normal test features for carrying out 2nd line operations.

During normal mode, on Power ON, system performs POST functions to assess health of the hardware and software modules. On successful completion of POST, it stores the POST results in data base with time and session details. If POST has failed, it waits for operator intervention for permitting system to come up and accordingly updates to data base. Based on the kind of failure in POST, system clearly indicates the extent of features available for testing.

After the POST stage, if system initializes normally, it provides the following functions:

a. Pre-Installation Checks
b. Acceptance Tests
c. SRU level fault Identification tests
d. GO/No GO Tests
e. Semi-automatic test mode

The important functionality of the tester is to provide fault isolation of Unit Under test (UUT) to a Shop Replaceable Unit (SRU). In this mode of operation, tester will start testing SRUs in a sequence and home on to the fault. In this method, the test cases are executed in predefined sequence and continue till fault is encountered. For example, UUT is having 10 SRUs and each SRU is taking 15 minutes of time to go though all test cases associated, then it requires about 150 minutes (2hours and 30 minutes) to home on to the problem correctly if fault is in last i.e 10th SRU. It is very important that turnaround for LRUs should be as low as possible to minimize AOG time during critical phase of operations like war time, etc. It is essential to improve the performance of product support systems for improved maintenance and better availability of aircrafts.

Secondly it is required to identify the faulty SRU without damaging other working modules in UUT, so that available resources can be used effectively. It is also important to diagnose the fault correctly and completely so that no latent faults or hidden / undiscovered faults will show up during UUT usage on aircraft.

To address all these, system architecture is configured suitably with self learning [4] features.

2.2 Key Self/machine learning features

2.2.1. Fault diagnosis Time Reduction:

System architecture provides provision for storage of final UUT test results with complete details like **a.** UUT part Number, **b.** UUT Serial Number, **c.** Operator details, **d**. Squadron details, **e**. Tester details, **f**. Faulty SRU details, **g.** Fault details, **h**. Test state and program/inference Engine details, **i**. Start time and End time, **j**. Support/health/Critical parameter details, etc,. on to local Database or to a external server Database using built in Network interface. These results can be retrieved and perform dynamic trend analysis. As usage of the system is more frequent, then the test data available for individual LRUs (same type) will also grow. For example, The LRU is having 10 SRUs and same type of LRU is tested by this equipment 15 times, then the test data/results of 15 LRUs will be used for trend analysis. If trend analysis indicates that 1st module is faulty in 2 units, 6th module is faulty in 3 units, 7th module is faulty in 9 units, and 10th module is faulty in one unit, then 2nd line test system architecture will orient to start testing first with 7th module considering higher probability rather than staring with 1st module to gain the test time.

This results in significant reduction of time in overall diagnosis of the fault and identifying the faulty module.

2.2.2 Power on Self Test (POST) time reduction:

Detailed and effective POST is essential for ensuring the correct system functionality during usage in fault diagnosis process. For every power on, system will get in to POST mode and carryout POST and these results are used for assessing health of the system towards dynamically reconfiguring the system accordingly. The reconfigured system will provide full or reduced functionality based on user requirement if faults found are in non critical areas. At the end of the POST, results will be stored in to database for future use. This data provides vital information about test system failure pattern due to usage in field or due to test system ageing. As number of Power ON sequences increases, database will grow and intern will help to carryout effective trend analysis. This POST failures trend analysis data will also helps in assessing the reliability of particular hardware segment of the test system.

This trend analysis data further helps in system architecture to learn and orient on to a particular critical segment of the system to focus quickly and more deeply to assess complete and correct health of the test system. It also helps in assessing and stock piling of critical spares for effective maintenance of the test system.

2.2.3. Internal generated parameters:

Test results should correctly indicate faulty module in UUT. This decision not only depends on the output results along with associated inputs and also depends on health status of the test system at the time of acquiring the data for making the decision. As a back ground operation, a continuous Built in Test(C-BIT) is performed to monitoring internal reference voltages and power supply voltages used for acquisition, timing references, tolerances, other health/critical parameters, etc, to assess continuously w.r.t time so that there is no test system internal fault associated with fault monitored in UUT and declared. Inference engine uses all these data before making decision for declaring fault. In this process, system also monitors ambient temperature during the period test is conducted, humidity, shock/'G' and number of hours of system usage etc,. to aid decision process. For example, if temperature is at higher side, either the system will declare Over temperature and then will not make any decision on the test data obtained or may provide suitable compensation w.r.t to temperature and then declare results.

3. System Implementation

The system with these feature have been implemented and tested for its performance. The system consists of customized/COTS hardware for interfacing with UUT and off the self tough book for hosting Inference Engine, GUI, Database, all test application modules and high speed network interfaces. It is fully rugged and portable systems designed to qualify for JSS 55555 (Indian Joint Services Standard) requirements. The system is designed based on PA9636+ controller and with required interfaces. One of the 2nd line tester's data acquisition system typical block diagram is shown in fig 2.Test system control and processing module is shown in fig 3. The final 2nd line system realized is shown in fig 4. The few internal components like G-Meter and Hour Meter are shown in Fig 5.

The controller board is configured around Tiva™ TM4C123GH6PGE microcontroller of Texas Instruments. The controller PCB consists of the microcontroller, two 16-bit ADCs (ADAS 3023, LM3S8962-IQC microcontroller configured for implementing Ethernet MAC/PHY 10/100, and 24AA256-I/MS Serial EEPROM.

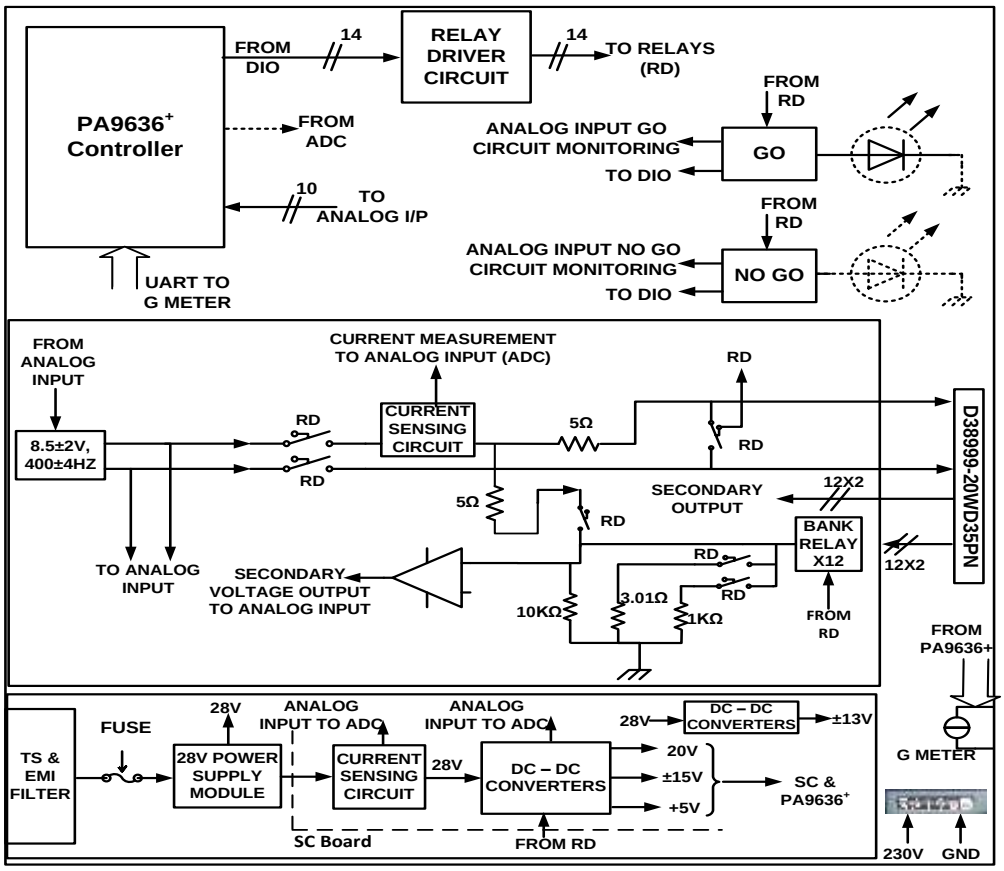
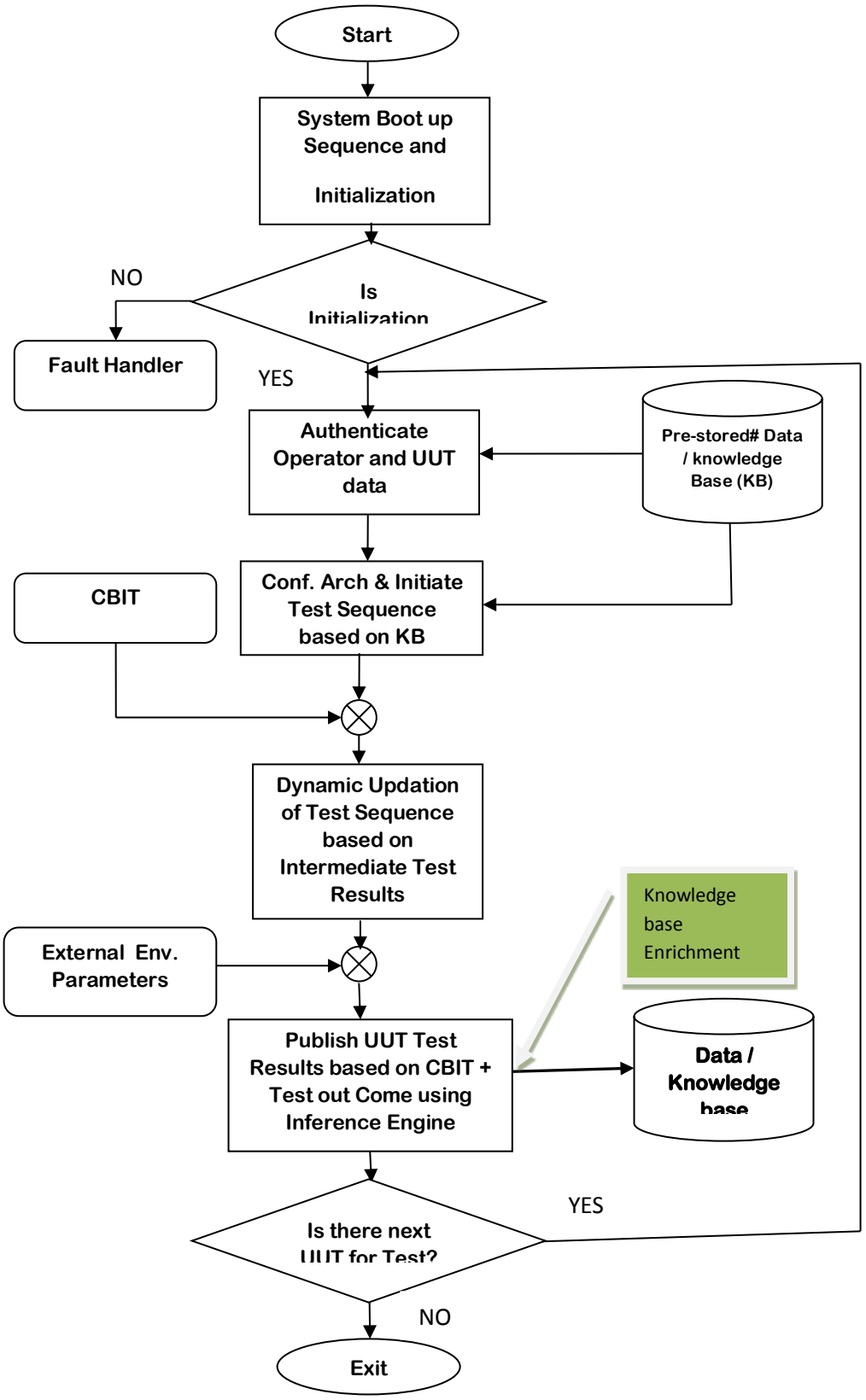


Fig. 2: 2nd Line tester for De-icing Unit (Typical) -DAQ Subsystem Block Diagram

Fig. 3: Control and Processing module

Fig. 4: 2nd Line tester for Product support

Fig. 5: G-Meter and Hour Meter



#: Initially populated data for Authentication information and data/KB structures.

Flow diagram 1: Typical Sequence of Operations of Self leaning Machine (2nd Line tester)

4. Relative Benefits

The key benefits of this self learning machine w.r.t conventional system are listed in table 1. The self learning machines reduce or eliminate skilled human factor during test cycle [5]. Since decision is taken by Inference Engine (IE) based on test data along with Test system health data, decision taken will be more accurate and safe. Based on the POST results knowledge, it also helps in planning more accurately spares required for the system the system and intern improves the system availability. This results in reduction in life cycle cost of the test equipment.

TABLE I: The Key Benefits

Sl.No	Area	Relative Benefit	Remarks
1	Average Time to identify faulty Module	30% to 40 % less	Based on initial studies.
2	The decision accuracy	Significantly High	As IE considers complete internal information.
3	System Reliability	Very high	As POST trend analysis and C-BIT data is used
4	System Complexity	Relatively High	Additional circuit or resources needed
5	System down time	Low	As advance information/ data is available for maintenance planning.
6	Human Dependency	Reduced	Reduced dependency on highly experienced and Skilled personnel as system learn with time
7	Knowledge Retention and enhancement	High and continuous	Human dependency in decision making is minimized or eliminated.

5. Conclusions

Self or machine learning architecture is best suited for the system where repetitive operations are performed and knowledge is gained from each operation. Here system learns with number of repetitive test operations/iterations. This concept has been adopted for critical applications like 2nd line testers of fighter aircraft product support systems. The greatest advantage in this approach is to reduce human dependency (on using past experience) and reduce time taken for faulty module identification. These concepts can be adopted for similar kind of applications in all other domains. There is a scope for evolving algorithms for inference engine [2]. The trend for using self or machine learning architecture is rapidly increasing as there are significant benefits with respect to safety, reliability, time and cost. These concepts can be adopted for similar kind of systems to gain advantages discussed in this paper.

6. Acknowledgement

Authors are indebted to PS Subramanyam and Dr. Girish S Deodhare for suggestions and inputs. Authors would like to thank Dr. K Karunakar for his valuable suggestions. A special thanks to Dr. D Sita Rama Raju for his encouragement, inputs and helpful guidance. We would also like to thank FCC team of IFCS, ADA and all other members who involved in development for their support.

7. References

[1] Evgeny AlekSeyevick Leonov and others, "Architecture and Self-learning of Knowledge Based Systems use monitoring of Internet Network" Bryank State Technical University, Bryansk, Russia.

[2] ARNAUD DOUCET "An Introduction to MCMC for Machine Learning" Department of Electrical and Electronic Engineering, University of Melbourne, Parkville, Victoria 3052, Australia, MICHAEL I. JORDAN, Departments of Computer Science and Statistics, University of California at Berkeley, 387 Soda Hall, Berkeley, CA 94720-1776, USA.

[3] Harmen Sthamer, André Baresel and Joachim Wegener, Daimler Chrysler,"Evolutionary Testing of Embedded Systems", AG, Research and Technology, Alt-Moabit 96a, D-10559 Berlin, Germany, PUBLISHED in: 14th International Internet & Software Quality Week 2001.

[4] Ram Keralapura, Antonio Nucci, Chen-Nee Chuah, "A novel self-learning architecture for p2p traffic classification in high speed networks", Computer Networks, Volume 54, Issue 7, 17 May 2010, Pages 1055–1068.

http://dx.doi.org/10.1016/j.comnet.2009.10.009

[5] A. Eltaher, and M. Ayeb, M. Maurer, "A generic architecture for hybrid intelligent test systems" in Proc. 2008, 7th IEEE International Conference. 2008, E-ISBN: 978-1-4244-2915-8.