Comprehensive Validation and Debug Tool for verification of Radar Controller

1Anusha M
1Senior Engineer
1Bharat Electronics Ltd.

Abstract - **Radar Controller (RC) developed for Surveillance Radar is an advanced GUI application which enables Radar operation and monitors health status of 14 functional subsystems and is equipped with intelligent automated error handling mechanism. RC being the only operator interface with the mission-critical system, meticulous validation of RC is imperative. A comprehensive software tool has been developed to simulate the entire Radar interface with RC and to enable thorough standalone verification of RC's wide-range of functionalities. The user-friendly icons-based tool is self-explanatory, and enables flexible and quick simulation of any type of error condition handled by RC, which is difficult to validate with the integrated system. The tool comprises of exhaustive GUI and time-stamped packet-capturing utility to simplify fault-finding and troubleshooting and makes the validation process self-reliant and independent of external tools. The software modules in the tool are generic and can be reused to validate similar software applications.**

keywords - **Radar Controller, Comprehensive Validation and Debug Tool**

I. Introduction

Radar Controller (RC) developed for Surveillance Radar is a comprehensive and advanced GUI-based software application used for Radar operation and status monitoring [1]. RC is a self-reliant application and is equipped with intelligent error handling mechanism as shown in Fig. 1, to appropriately automate the error management process, exclusive of human intervention.

RC controls and monitors health status of fourteen functional subsystems of the Radar broadly classified as Antenna Control Unit, Power Distribution Unit, Transmitter, Receiver, Sensitivity Time Control, Waveform Generation Module, Signal Processor, Power Supply Modules, Power Monitoring Unit, Centroider, Tracker etc [1].

Systematic and complete validation of RC is essential as RC is the only operator interface with the mission-critical system. A comprehensive software tool has been developed to simulate the entire Radar interface, and to perform thorough standalone verification of RC and finally, to facilitate seamless integration of RC with the Radar system and ensure safe operation of the mission-critical system.

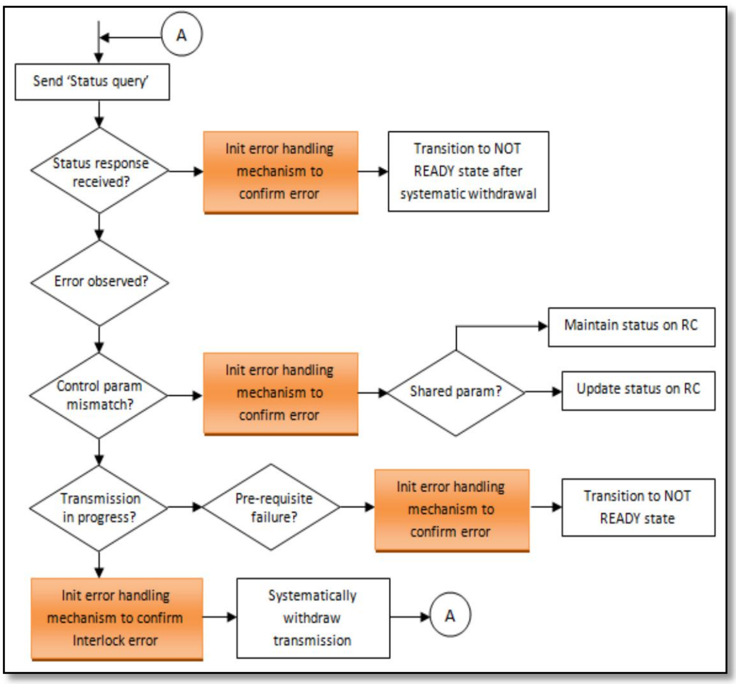

Fig 1. Overview of the Error Handling Mechanism in Radar Controller

II. Comprehensive Validation and Debug Tool

Comprehensive Validation and Debug Tool (CVDT) is a sophisticated software tool which simulates the entire Radar interface with RC in a single software application. CVDT comprises of exhaustive GUI to enable meticulous standalone verification of RC's wide range of functionalities [1][2].

CVDT enables quick simulation of any type of error condition that occurs in the Radar system and systematically corroborates respective error management response from RC. CVDT facilitates flexible verification and validation process with simulation of various probable exceptional error conditions that are difficult to substantiate with the integrated Radar system [3].

Icons-based indications enhance user-friendliness and makes CVDT self-explanatory. CVDT provides real-time text-based error message display and simplifies fault-identification. CVDT includes in-built time-stamped UDP packet display utility to alleviate troubleshooting and make validation process self-reliant and independent of external tools.

CVDT software is developed in C# using .Net framework and is modularized, which makes the software modules reusable to validate similar Radar software applications.

III. Implementation

Critical Radar subsystem parameters are classified as prerequisites, interlocks, control parameters and transmission related parameters. RC is designed to initiate error handling scheme and intelligently take automatic appropriate action respectively for each error condition related to critical parameters as classified above. RC also periodically monitors the connectivity of subsystems and is intended to take appropriate action during failure condition.

Radar Interface Simulation

The Radar communication architecture is divided into multiple network domains based on the functionalities of Radar subsystems and the network load imposed. Real-time configuration of network interface is provisioned in CVDT.

CVDT GUI comprises of 12 pages for configuration of parameter status and values of 12 subsystems respectively. Each respective subsystem page is viewed by clicking on the respective arrow button. Subsystem parameters are configured based on the test cases and 'Update' button is clicked for updating the configurations in the status response to RC, as shown in Fig. 2.

Subsystems comprising of only overall subsystems health status are configured using dropdown menu as shown in Fig. 2. Control command parameters that are not a part of any status response are verified through 'OTHER CTRLS' page. Timer based simulation of Antenna position as well as Output power value during Receiver calibration is also available as shown in Fig. 2.

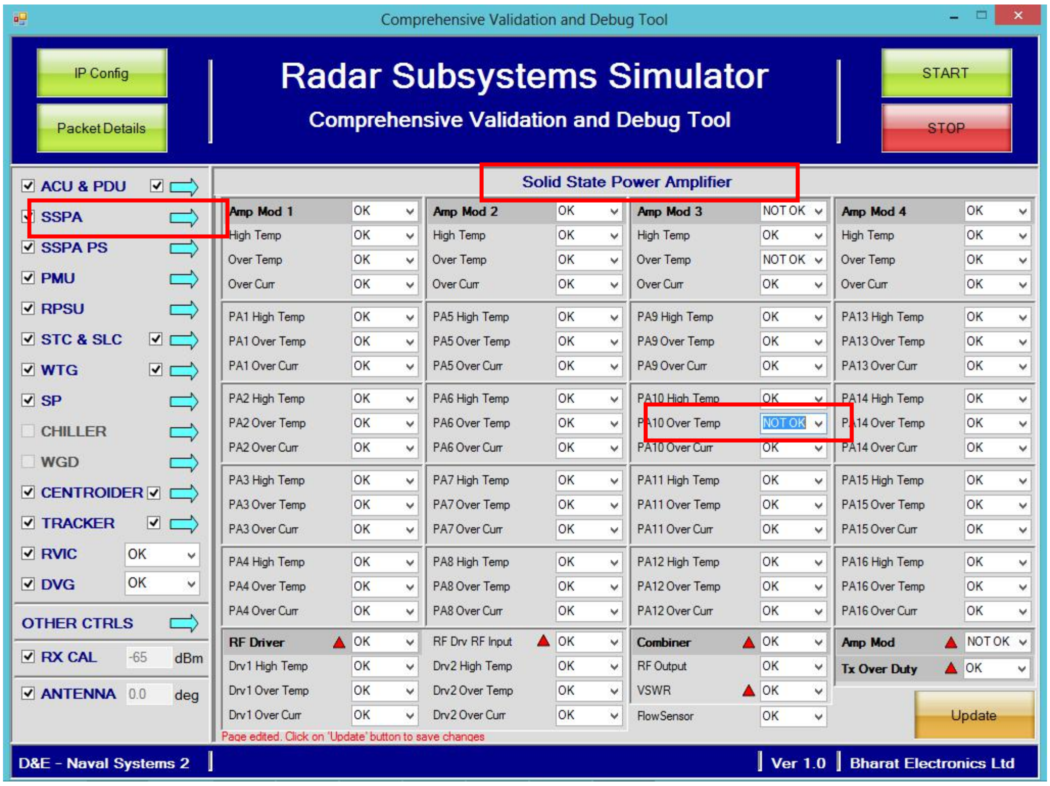

Fig 2. Sample snapshot of CVDT to demonstrate subsystem parameter configuration

Simulation of error conditions

Critical Radar subsystem parameters are classified as prerequisites, interlocks, control parameters and transmission related parameters. RC is designed to initiate error handling scheme and intelligently take automatic appropriate action respectively for each error condition related to critical parameters as classified above. RC also periodically monitors the connectivity of subsystems and is intended to take appropriate action during failure condition.

a. Subsystem connectivity failure: Status response from subsystem is disabled when respective subsystem's checkbox is unchecked to simulate subsystem connectivity failure condition, as shown in Fig. 3.

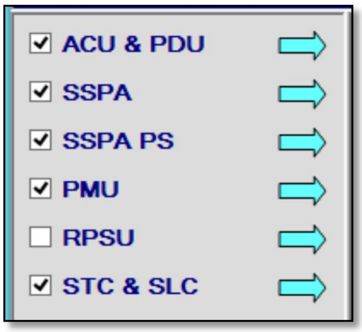

Fig 3. Subsystem connectivity failure simulation

b. Prerequisite failure: Prerequisite parameters as indicated with orange triangle icon are configured accordingly to simulate error condition and the automated action performed by RC to transition Radar to NOT READY state is verified, as shown in Fig. 4.

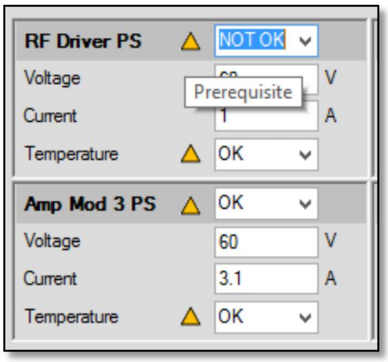

Fig 4. Prerequisite failure simulation

c. Interlock failure: Interlocks as indicated with red triangle icon are configured to simulate error condition when transmission is enabled by RC and the automated action executed by RC to systematically discontinue transmission is verified, as shown in Fig. 5.

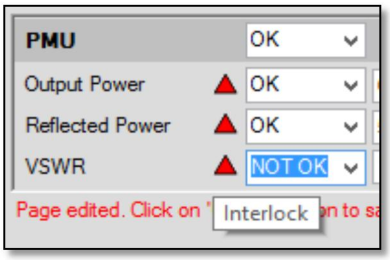

Fig 5. Interlock failure simulation

d. Control parameter mismatch: Control command related error conditions are provisioned to be simulated in two ways and automated action accomplished by RC is verified.
One way to simulate parameter mismatch error condition is on command initiation i.e. checkbox of the respective control parameter is unchecked to simulate rejection of control command, as shown in Fig 6.

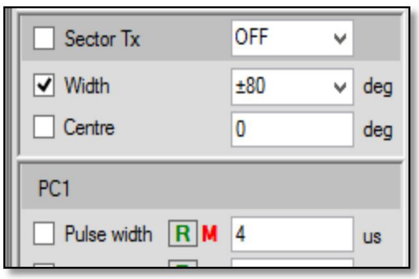

Fig 6. Control parameter mismatch simulation on command initiation

Other way of simulating the error condition is during Radar operation i.e. the value of the control parameter in textbox or dropdown menu is modified to simulate error during Radar operation, as shown in Fig 7.

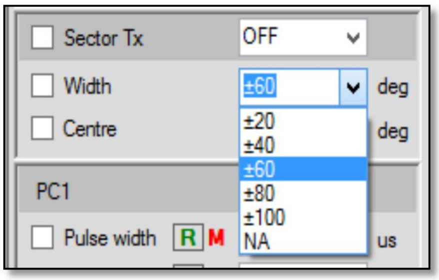

Fig 7. Control parameter mismatch simulation during Radar operation

e. Transmission related parameter error: Parameters significant to transmission as grouped with blue background are tightly coupled. These parameters are crucial and are intended to follow a specified sequence which is verified by varying their status and simulating required test conditions, as shown in Fig. 8.

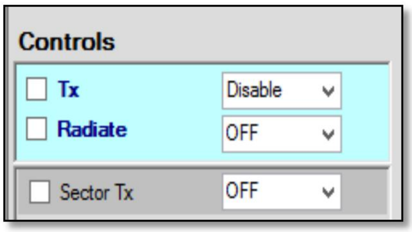

Fig 8. Transmission related parameter mismatch simulation

f. Shared control parameters error: Control commands shared between several subsystems as indicated with 'M' icon are modified based on the test case to simulate error and validate action performed by RC as shown in Fig. 9.

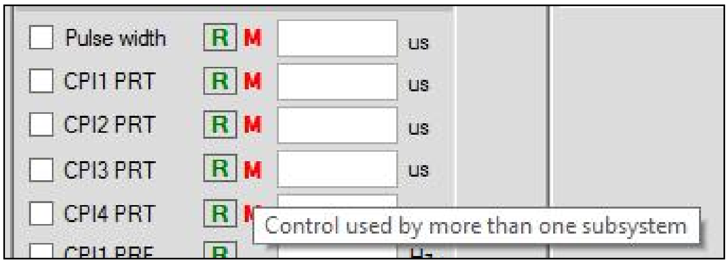

Fig 9. Shared control parameter error simulation

g. Range related error: The Radar is configured for Long range (long pulse) and Medium range (short pulse) operation. Certain parameters have varied values based on the range selection as indicated with 'R' icon are modified as required by the test scenario to simulate error as shown in Fig. 10.

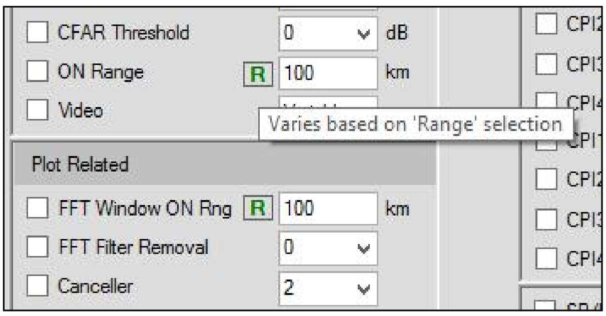

Fig 10. Shared control parameter error simulation

Real-time error message display

Control command sent by RC is validated in real-time by CVDT with the set of acceptable values and error message is displayed immediately to facilitate quick identification of error, as shown in Fig. 11.

Fig 11. Error message display

Packet display utility

In-built packet display utility provides time stamped details of the UDP datagram sent by RC as a part of Control command or Status query to ease troubleshooting, as shown in Fig. 12.

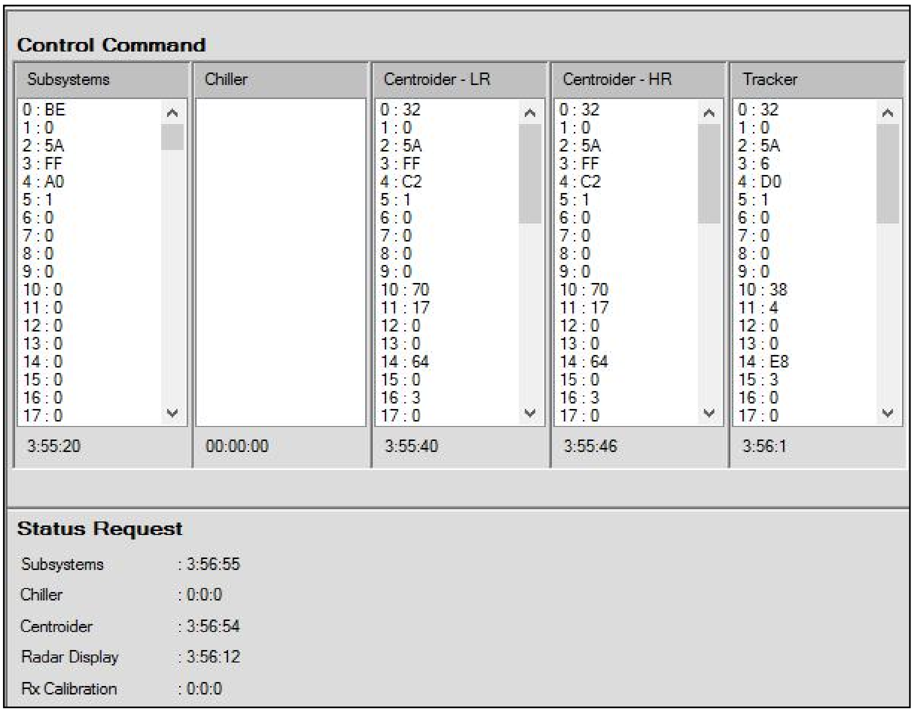

Fig 12. Packet Display Utility

IV. Salient Features

Some of the salient features of CVDT are listed below:

a. CVDT provides flexibility of enabling and disabling subsystem interface through GUI and facilitates systematic validation of each subsystem interface as well as entire Radar interface with RC
b. Icons-based CVDT enables quick and self-reliant identification of important Radar subsystem parameters without document reference and effectively reduces the software testing time [10]
c. Wide range of hints in the form of tooltips and text indications makes CVDT self-explanatory.
d. CVDT ensures meticulous validation of RC by simulation of any combination of errors which is difficult to simulate in the integrated system
e. Packet display utility in CVDT provides useful debug interface and eases troubleshooting without depending on any external packet capturing tool.
f. CVDT is developed in modular form which is easily maintainable and reusable.

V. Conclusion

CVDT is a self-reliant, self explanatory and user-friendly tool which plays a vital role in comprehensive and rapid standalone verification and validation of Radar Controller software used for operation and health monitoring of mission-critical system. CVDT ensures the robustness and appropriateness of RC with flexibility to simulate exhaustive test scenarios and facilitates seamless integration of RC with the Radar system, and ultimately ensures safe operation of the Radar system. CVDT is developed in modular form with easily configurable software modules. Considering that parameters across solid state Radars are generic in nature, with modifications or additions, this tool can be easily extended for validating Radar Controller applications of other similar Radars.

VI. Acknowledgment

Extremely thankful to Mrs. Dharani B, AGM, D&E/Naval Systems-2, Bharat Electronics Ltd, for constantly providing technical support, guidance and valuable inputs for performing innovative software improvements, in the capacity of Project Manager.

References

[1] Anusha M, Sharath L and Hanamant V M, “Design and Development of Radar Controller for 2D Air Surveillance Radar,” 11th International Radar Symposium India – 2017 (IRSI-17), December 2017.
[2] Inder Singh Chauhan, Jang Bahadur Singh and Justin Sagayaraj M, “A Generic Robustness and Stress Testing Tool for Radar Software,” 11th International Radar Symposium India – 2017 (IRSI-17), December 2017.
[3] Shayistha Banu A, JB Singh, Jithesh V and Justin Sagayaraj M, “IVV Process for airborne Radar software – A case study,” 11th International Radar Symposium India – 2017 (IRSI-17), December 2017.

[4] Kiran Jammalamadaka and Nikhat Parveen, "Holistic Research of Software Testing and Challenges", International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-8, Issue- 6S4, April 2019
[5] Nahid Anwar and Susmita Kar, "Review Paper on Various Software Testing Techniques & Strategies", Global Journal of Computer Science and Technology Volume XIX Issue II Version I, 2019
[6] Komal, "Review Paper on Software Testing", International Journal of Engineering Research & Technology, NCAEM Conference Proceedings, ISBN: 978-93-83758-09-8, 2013
[7] Himanshi Babbar, "Software Testing: Techniques and Test cases", International Journal of Research in Computer applications and Robotics, Vol. 5 Issue 3, Pg.: 44-53, ISSN 2320-7345, March 2017
[8] Muhammad Abid Jamil, Muhammad Arif, Normi Sham Awang Abubakar and Akhlaq Ahmad, "Software Testing Techniques: A Literature Review", 6th International Conference on Information and Communication Technology for The Muslim World, 2016
[9] Kazimierz Worwa, "Logistical Aspects of the Software Testing Process", Vol. 6, No. 2, pp. 155-164, DOI: 10.21008/j.2083-4950.2016.6.2.5, Poznan University of Technology, April 2016
[10] Shruti N. Pardeshi, "Study of Testing Strategies and available Tools", International Journal of Scientific and Research Publications, Volume 3, Issue 3, ISSN 2250-3153, March 2013