Test and Evaluation of Cognitive EA systems - Requirements for Future Test Systems

Dan Pleasant

Keysight Technologies, Inc., Santa Rosa, CA, 95403, USA

***Abstract* — While the world has been hard at work developing cognitive AI-based approaches to electronic attack, little thought has been given to the ways in which such EA systems must be tested and their performance evaluated against various threats. The common parametric test techniques that have been used in the past will not be sufficient. This paper examines the problem and finds that there are two major areas of interest: First, a closed-loop test system is required; and second, a simplified method of evaluating the effectiveness of a jammer must be implemented. The paper will present potential test system architectures, highlight the gaps in the available test technologies, and discuss at a high level the considerations that must be taken into account when evaluating EA system effectiveness.**

***Index Terms* — Cognitive EW, T&E.**

I. Introduction

Near-future designs of machine-learning based cognitive electronic warfare systems are posing new types of challenges to test systems. Traditionally, test systems have usually employed some variety of stimulus-response testing, wherein a signal is applied to a system under test (SUT) and the SUT's response is captured and analyzed. Knowing what the response should be, it's easy to evaluate the SUT's performance and determine whether it is working properly. But cognitive systems violate the basic premise of stimulus-response testing because there is no predictable response to a given input, and in fact the SUT's response may change over time as it 'learns' more about the stimulus. Therefore, the performance of a cognitive system must be evaluated in a different way.

This paper outlines a system design that could be used to test a cognitive electronic attack system (jammer). The system would be potentially useful in a number of scenarios – for evaluating jammer performance against various threat radars, for training, and even for mission planning. This candidate design can then be evaluated against the operational requirements of such a system to determine whether it would successfully meet its goal, and to understand further requirements that may become apparent. This paper will show that a standalone test system is not sufficient, new technology will have to be developed, and the SUT will have to be a full participant in its own testing.

II. Scenario

To test a cognitive jammer, it is necessary to emulate a particular scenario. A simple scenario is illustrated in Fig. 1. The signals in this scenario will be artificially created and applied to the jammer's input(s). The radar system must be emulated in order to evaluate the jammer's performance.

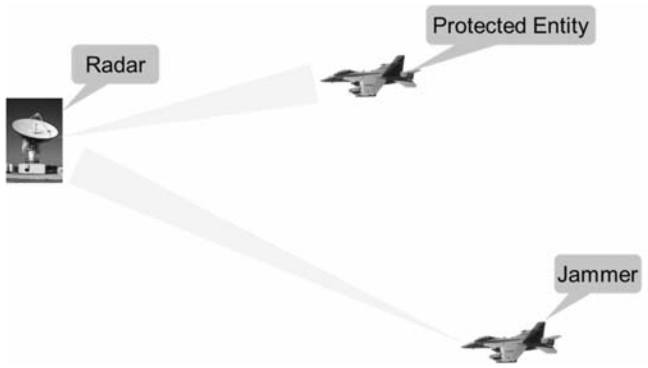

Fig. 1 A simple scenario that will be emulated by the candidate test system.

Fig. 1 illustrates a simple scenario that contains only a single radar, a single jammer, and a single protected entity. More advanced systems will require multiple elements of each type, requiring the system to be able to scale to larger scenarios. This puts some additional requirements on the test system that will be discussed briefly later in this paper. In the meantime, this simple scenario can be used to make some important observations.

III. Example test system

A high-level block diagram of the proposed system is shown in Fig. 2. This is not the only possible design for such a system, but it utilizes some state-of-the-art measurement hardware and makes a good example. Any other system design would encounter the same issues, so the example in Fig. 2 will serve to illustrate.

978-1-5386-9549-4/19/$31.00 ©2019 IEEE

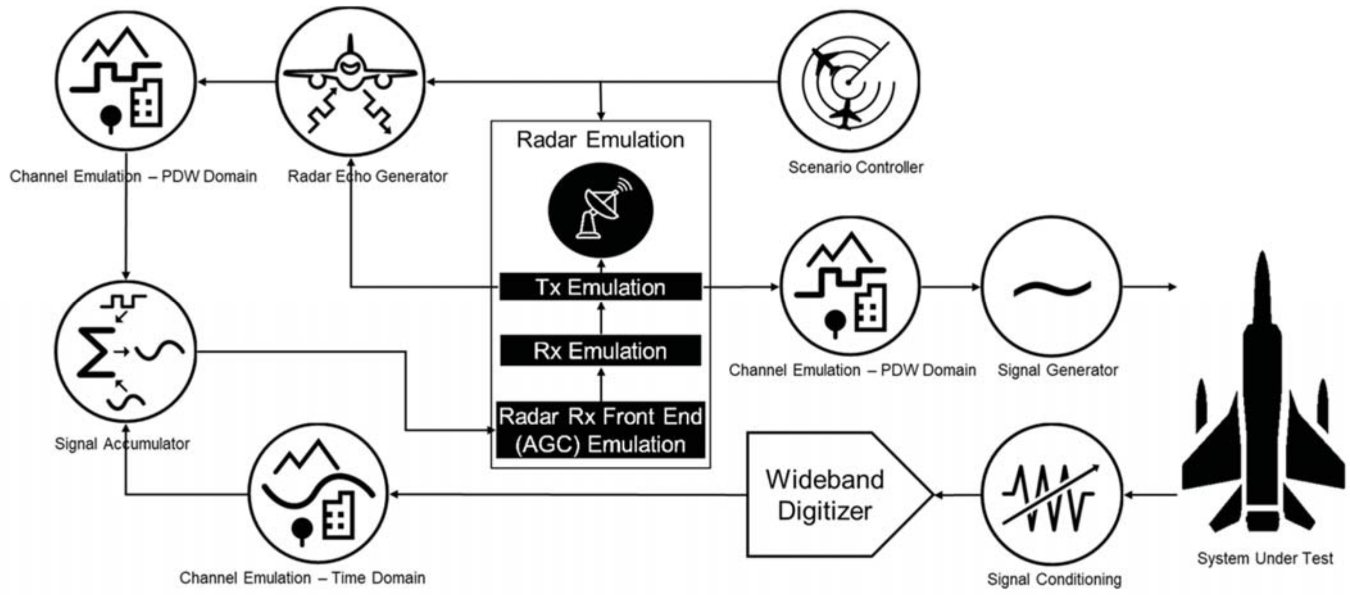

Fig. 2 A candidate test system to evaluate jammer effectiveness.

The overall operation of the system shown in Fig. 2 is described in the following paragraphs. The function of each block and its relationship with the SUT will be considered at each step. This 'thought experiment' will highlight the types of challenges that will be faced by the implementors of such a system.

A. Scenario Controller

A scenario controller is a software application (often supplemented by a GPU) that keeps track of the actions of all the objects in the emulated scenario – their positions, velocities, aspects, and relationships with one another. Such applications are broadly available although many of them are not architected for closed-loop operation and can only create a set of signals that would be applied to the SUT, without the ability to emulate a radar in real time.

It is here that we first encounter challenges in the implementation of this test system:

- The SUT may function and maneuver autonomously, without the knowledge of the scenario controller. This presents the need for a two-way communication path between the SUT and the scenario controller. The scenario controller needs to know where the SUT is at any given time, and the SUT needs to know about the surrounding terrain and other emulated objects so that it can properly maneuver.
- Timing is key – precise timing between all system elements is critical, including the SUT.

B. Radar Transmitter

Radar emulation is central to the operation of this test system. Depending on the purpose for which the system is to be used, radar emulation may be high-fidelity (emulating a specific radar model) or more general-purpose in nature. In any event it is important for the radar emulator to be reconfigurable so that it may be used to emulate more than one kind of radar.

In the system architecture used here, the radar transmitter emulator's output is a series of pulse description words (PDWs), which describe the pulses that the radar wishes to transmit. The PDWs will change according to the mode of the radar, and the radar emulator must transmit certain information (such as antenna beam direction) to other test system elements.

Radar emulation will likely be implemented with a combination of software and FPGA-based firmware, requiring the ability to reconfigure FPGA contents at will. This will require new technology development.

C. Channel Emulation (PDW domain)

The PDW-based output of the radar transmitter emulator must be scaled to account for the signal path between the radar and the SUT. For simple cases, this may be a simple delay and attenuation factor. More advanced systems will include models for multipath transmission, general scattering, atmospheric absorption, and other factors that may affect the transmitted signal. Channel models must be applied to the PDWs to account for these effects.

But the SUT must participate in channel emulation, and there are other logistical problems:

- The test system knows the beam direction of the radar but not the beam direction of the SUT. This requires a communication path between the SUT and the transmit channel emulator in order to properly scale the SUT's input signal.
- The SUT antenna might be an array. Full, element-by-element testing of phased array antennas is possible

but expensive. Alternatively, the signal may be applied to the SUT at a point behind the antenna if a physical connection is available.

- The SUT might be trying to execute a direction-finding algorithm, requiring a test setup specifically design to enable direction-finding (with a calibration routine), or the ability for the test system to 'push' direction information into the SUT's controller via a special test mode.

D. Signal Generator

Fortunately, a number of suitable signal sources have become available in recent years. Such sources accept the PDW output of the channel emulator and create RF signals to match, which can be applied to the input of the SUT.

E. Signal Conditioning

The SUT's output must be assumed to be unpredictable and of varying power levels over time. It is important to prevent the SUT's power, which may be quite high, from damaging the input of the downstream receiver circuits. At the same time, the receiver's input power should be in a range that allows the digitizer to operate near full scale so that optimum signal fidelity is maintained.

Signal conditioning can be achieved through the user of a fast gain control circuit (with power protection at the receiver input to handle any transient signals). Preferably, a software interface with the SUT would allow the test system to determine appropriate attenuation levels – obviously requiring cooperation from the SUT.

There is an additional, and unusual, requirement on the signal condition solution needed for this test system. To enable proper signal scaling downstream, the signal conditioning system must communicate its attenuation setting to the rest of the test system. The signal conditioning system must therefore include the appropriate communications mechanism while also participating in system timing functions.

F. Digitizer (receiver)

Assuming successful implementation of signal conditioning at the SUT's output, the receiver design may be straightforward. The block diagram in Fig. 2 assumes the use of a wide-band digitizer of a type that has become recently available off-the-shelf. This type of design does not require a tunable downconverter and may make it easier for the system to handle frequency-hopping SUTs. A more traditional downconverter-based design (with a narrowband digitizer) could also be used, in which case the test system must be able to follow frequency changes by the SUT. In either case, an interface with the SUT that allows the test system to more easily follow frequency hops is desirable.

G. Channel Emulation (time domain)

The SUT's captured output signal must be modified to account for channel effects over the path from the SUT to the radar. Although conceptually identical to the channel emulation already discussed, the implementation here is different because the input signal consists of time-domain sampled data (probably baseband IQ data), not PDWs. This requires considerably more processing power.

Commercial channel emulators are available which can perform this function, but their inputs and outputs are usually RF signals, and they are designed for mobile phone applications, so they are not useful in the context of the test system examined in this paper. The underlying technology is available but must be re-implemented in the proper form.

H. Radar Echo Generator

The total set of signals in the simulated scenario must include radar echo returns. In the very simple case illustrated in Fig. 1, there are two echo returns: one for the jammer and one for the protected entity. These echo returns must be mathematically generated in software, with channel models applied. These signals are represented as PDWs to keep data bandwidths low, and the channel models are like the functionality already described for the radar's transmitted signals except for the addition of round-trip effects rather than just one-way. The SUT is not involved in this step.

I. Signal Accumulator

The captured signal from the SUT must be combined with the emulated radar echos, using precise timing. The signal accumulator block represents a new (but straightforward) technology that must be developed. Note that the echo signals are in the PDW domain while the SUT's output is captured in time domain, so the signal accumulator must translate PDWs into time domain. In addition, the SUTs signal will not arrive at the same time as the emulated radar echo signals, so buffering and careful timing is necessary. Once accomplished, the output of the signal accumulator is a time-domain signal that represents the RF signal that would be captured at the radar's antenna.

J. Radar Receiver Emulation

It is important to properly emulate the radar receiver, including automatic gain control at the receiver's inputs. (Many jamming techniques rely on the jammer's ability to manipulate the radar's AGC so that actual echos are driven down into the noise floor while the jammer's signal is stronger.) Radar receiver algorithms must also be implemented. However, the necessary fidelity of radar emulation depends on the specific use case – some systems will require detailed, high-fidelity emulation of specific radars, while others only need general-purpose radar signals of different types and do not need to exactly match any particular model of radar.

IV. Jammer Effectiveness Evaluation

A survey of the industry indicates that there is no standard method for evaluating the effectiveness of a jammer. In fact, jammer effectiveness is at its core a mission-specific measurement. If a jammer can "fool" a radar for five minutes, is that enough? It's easy to see that the answer depends on the specifics of the mission.

More generally, jammer effectiveness should be evaluated on a statistical basis – especially since the jammer may 'learn' from one simulated run to the next, and its behavior will be modified by noise, different navigation paths, and its own cognitive algorithms.

Clearly, a standard measurement technique would be helpful. Such a standard would help to avoid confusion and duplication of effort across the field. For now, no such standard exists. This should be the topic of future papers and industry discussions.

V. System Scalability

It is important to consider the issues that will arise when the type of test system in Fig. 2 is scaled up to include multiple radars, jammers, and protected entities. Noting that all radar signals must be applied to all jammers, all jammer signals must be applied to all radars, and all radar echos must be received by all radars, it can be seen that the complexity of signal and data connections in the system will expand factorially as the system is scaled up. This presents architectural problems that must be resolved to prevent the system from becoming unmanageable

As a subject for future investigation, it is proposed that ring and star architectures be implemented. In these types of architectures all of the necessary data is broadcast to all every part of the test system. System blocks can then listen for the data they need, ignoring the rest. Such architectures are often used to create scalable systems – but these architectures are not common in test and measurement applications, and further investigation will be needed.

VI. Conclusions

Testing cognitive jammers – or any other cognitive system – will require the same level of innovative thought that's required to design the cognitive systems in the first place.

Cognitive systems will have to be full participants in their own testing, with software and hardware interfaces included that allow the test system to communicate with the SUT in a peer-to-peer fashion. Designers of cognitive systems will be well-served to consider these requirements early in the design and development process. Otherwise, expensive modifications will be needed in order to test them later.

Likewise, test system designers will be forces to implement new technologies and new architectures – and the costs of such systems must be included in budgets at an early stage.