

# Developing *Test Software* for Naval Automated Test Systems

BY CAMERON MCNATT,  
EAGLE SYSTEMS, INC.

U.S. Naval and Marine Corp aircraft such as the FA-18 Hornet, V22 Osprey, AV-8B Harrier, and EA-6B Prowler must be maintained in a constant state of mission-readiness. When an aircraft electronic or avionic system fails, technicians typically analyze the potentially malfunctioning assembly by running a series of Test Program Sets (TPS) on an Automated Test System (ATS). The U.S. Navy's premier ATS is Consolidated Automated Support System (CASS), which is used in Naval repair depots, aboard aircraft carriers, and in Marine Corp mobile facilities.

In this article, Cameron McNatt, an electrical engineer at Eagle Systems and contractor for Naval Air Systems Command (NAVAIR), describes how he used MATLAB® to create software to aid the development of Vectorized Test Program Sets (VTPS).

VTPS increase testing efficiency by applying, measuring, and analyzing vectors of input and output signals in parallel. They were designed to enable test engineers to simulate real-world operating conditions and to leverage the new automated test equipment recently installed on CASS.

When CASS was fielded to the Naval Fleet in the early 1990s, it consisted of 1980s technology. Limitations in the CASS hardware and software restricted TPS design to a linear, step-by-step approach.

As part of a program to update CASS, the U.S. Navy added three Teradyne Ai-710

parallel analog test instruments to each system. Each of the Ai-710's 32 channels can be programmed as one of six different instruments: an arbitrary waveform generator, a function generator, a digital multimeter, a timer counter, a signal digitizer, or a limit detector. Multiple instruments per channel—including source and measurement instruments—and multiple channels can be used simultaneously. Previously, CASS had fewer than a dozen low-frequency channels. With three Ai-710s, it now has 96 highly synchronized low-frequency source and measurement channels.

The parallel capabilities of the Teradyne Ai-710 enable the test engineer to source signals to multiple inputs and measure multiple outputs of a unit under test (UUT) concurrently. To date, however, few test engineers have taken advantage of the new hardware. There are probably several reasons for this, including the comparatively recent introduction of the Ai-710 to CASS, the serial structure of traditional test program design, and the lack of software to support parallel analog tests.

## Parallelizing Analog Tests in MATLAB

As part of a joint effort, NAVAIR, Eagle Systems, and The MathWorks developed test software in MATLAB to take advantage of Vectorized Test Program Sets and help test engineers make the most of the Ai-710's parallel architecture.

A key consideration in the selection of MATLAB for the TPS development software was its ability to configure and control instruments and other test hardware. MATLAB enables communication with instruments such as the Teradyne Ai-710 using Instrument Control Toolbox, which works with commercial off-the-shelf instrument drivers like *VXIplug&play* and IVI. In this project, Instrument Control Toolbox interfaced with the Ai-710 instrument using a *VXIplug&play* instrument driver supplied by Teradyne. MATLAB enabled us to configure and control the Ai-710 using both a graphical and a command-line interface, making it easy to send and receive data.

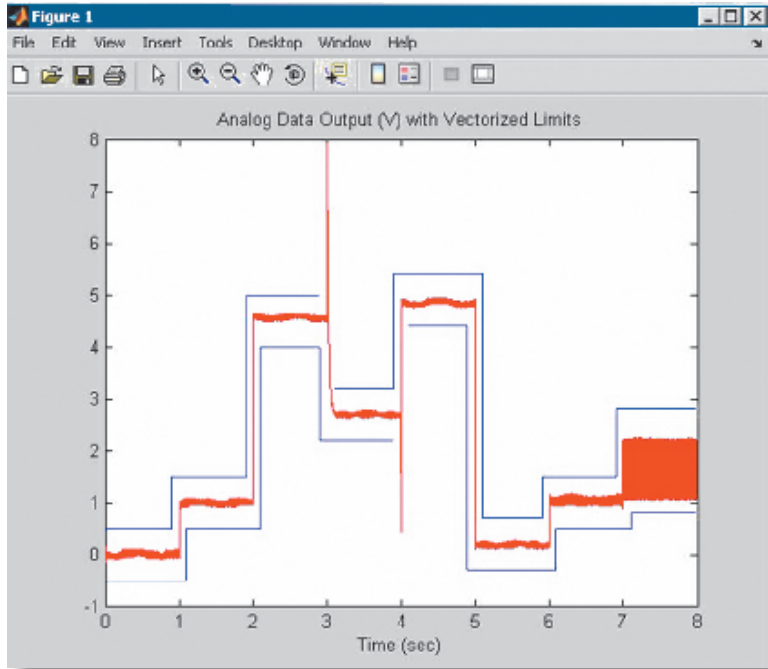


Figure 1. The VDT application developed in MATLAB has a built-in criteria function, enabling the test engineer to run a single vectorized comparison rather than looping through an array of data.

In traditional TPS testing methodology, only a single UUT input and output pair is typically examined in each test. With VTPS, input and output data is specified as vectors in amplitude and time, and sets of vectors are sourced to UUT inputs and measured from UUT outputs in parallel. Each output vector is a function of the input vector and the UUT. MATLAB is an excellent mathematical and algorithmic language for analyzing these large sets of data and for understanding the relationship between the vectors' sets of output and input data.

With MATLAB, we can test parallel input and output signals. Once we have applied the input vector to the UUT and gathered output data, we can compare the output vector data with a high-limit vector, a low-limit vector, or both (Figure 1). In MATLAB, this vector comparison is written as a single operation; no looping through array data is required. Because we also collect input data as vectors, we can perform a similar comparison to ensure

that the input is within specified test parameters.

To simplify the development of test programs, we created a graphical application in MATLAB called the VTPS Development Tool (VDT). With this application, users can map the inputs and outputs of the UUT to Ai-710 channels. They can also

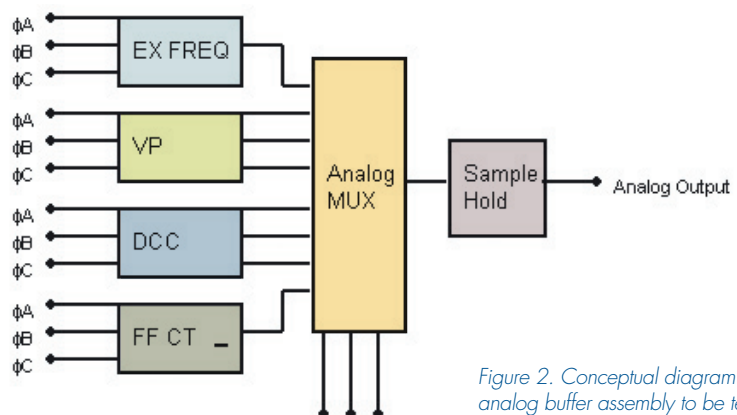


Figure 2. Conceptual diagram of the A7 analog buffer assembly to be tested.

determine the input signals, set the data collection rate and test duration, and initiate the tests. The application displays the selected test data in multiple plots with a universal time axis.

Working in the MATLAB environment, test engineers can further analyze the input and output data. Alternatively, they can specify a criteria function within the application to create a true VTPS—one that produces a pass/fail result.

### Testing the A7: A VTPS Demonstration

To demonstrate that Vectorized Test Program Sets are a practical alternative to traditional TPS testing, we developed one for the A7 analog buffer circuit card assembly (Figure 2). Used on the AV-8B Harrier, the A7 assembly routes information on several AC signals to a microprocessor. The microprocessor selects the information to be routed by switching an analog multiplexer (MUX). The primary channels of interest on the A7 are the AC inputs, the MUX control inputs, and the single analog output.

With a traditional TPS, each input of the A7 analog buffer assembly is tested separately. In this scenario, the tester sets up each channel sequentially, a task that can consume up to 50% of testing time.

Testing the A7 buffer assembly with a

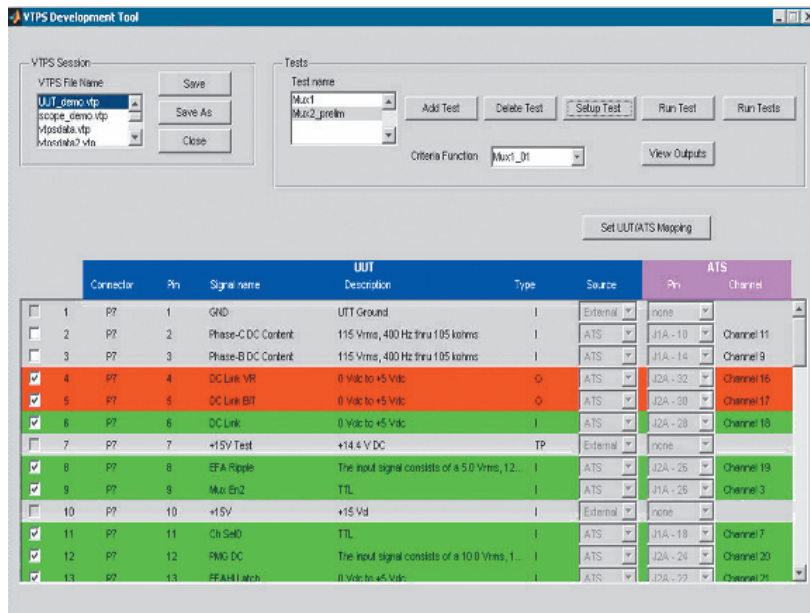


Figure 3. The VTPS Development Tool enables users to select and configure inputs and outputs.

VTPS executed in the MATLAB application is much more efficient, and presents the A7 with operating conditions closer to those in its real-world operating environment. Employing multiple channels on the Ai-710, the application enables users to connect all I/O pins of the A7 to the Ai-710 instrument once at the start of the tests, eliminating the need to disconnect and reconnect shared inputs. Each input waveform generator can be started at the beginning of the test, minimizing setup delays.

The output of the A7 assembly must switch within a specified time frame in response to a different MUX select. In a traditional TPS, it is difficult to measure this switching time because control over instrument timing characteristics is limited. It is much easier with the Ai-710 and the VDT application, which is designed to show those timing relationships and interactions between channels.

The first step in using the VDT is to map UUT I/O pins to Ai710 channels (Figure 3).

Next, we set up the stimulus on the channels of interest. On the input chan-

nels, we decide the type of signal, the signal parameters, and the time limit of the test. Our signal stimuli are displayed on a single time axis, giving us a better understanding of the interchannel timing relationships.

We also select the output channels that we want to measure. After a test is run, the application displays the digitized input and output together on a universal

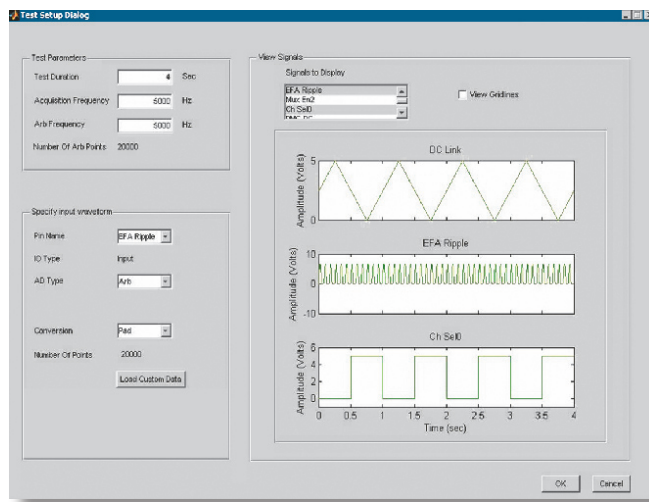


Figure 4. The VDT application displays input and output waveforms on a universal time axis.

time axis to enable users to visualize the actual measured timing relationships and gain insight into the UUT's performance (Figure 4).

By specifying a criteria function in the application, we can create a VTPS that produces a pass/fail result for the test run. The criteria function compares the output vector against high- and low-limit vectors. For example, a criteria function could be coded in MATLAB as

```

if (all(~(fData > LimHigh)) &&
    all(~(fData < LimLow)))
fStatus(jdx) = 1;
else
fStatus(jdx) = 0;
end

```

MATLAB matrix and vector capabilities make comparison straightforward.

### Next Steps

Promoting the ability of VTPS to reduce testing time and improve test effectiveness is a multiphase effort. The first-phase demonstrations on a standalone Ai-710 were well received by U.S. Navy personnel. MATLAB has proved a viable and practical software environment for developing efficient parallel test systems due to its ability to control test hardware and analyze VTPS.

In the second phase we will demonstrate the MATLAB based VTPS Development Tool and VTPS on CASS with multiple Teradyne Ai-710 instruments. We are planning enhancements to the application's visualization capabilities and criteria functions. Other improvements will enable VTPS to be used to debug

the assemblies and isolate the cause of assembly failures. The future usage of VTPS with these test systems depends on making them accessible to test engineers and to the U.S. Navy. ◀◀

---

### For More Information

- NAVAIR [www.navair.navy.mil](http://www.navair.navy.mil)
- Test and Measurement  
[www.mathworks.com/res/testandmeasurement](http://www.mathworks.com/res/testandmeasurement)

## Resources

### VISIT

[www.mathworks.com](http://www.mathworks.com)

### TECHNICAL SUPPORT

[www.mathworks.com/support](http://www.mathworks.com/support)

### ONLINE USER COMMUNITY

[www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)

### DEMOS

[www.mathworks.com/products/demos](http://www.mathworks.com/products/demos)

### TRAINING SERVICES

[www.mathworks.com/training](http://www.mathworks.com/training)

### THIRD-PARTY PRODUCTS

[www.mathworks.com/connections](http://www.mathworks.com/connections)

### WORLDWIDE CONTACTS

[www.mathworks.com/contact](http://www.mathworks.com/contact)

### E-MAIL

[info@mathworks.com](mailto:info@mathworks.com)



©1994-2007 by The MathWorks, Inc.  
MATLAB, Simulink, Stateflow, Handle Graphics, Real-Time Workshop, SimBiology, SimHydraulics, and xPC TargetBox are registered trademarks and SimEvents is a trademark of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.