

VECTORIZED TEST PROGRAM SETS USING MATLAB AND THE TERADYNE AI-710 ANALOG TEST INSTRUMENT

Cameron McNatt
NAVAIR (ESI)
BLDG 8141 – Villa RD
Saint Inigoes, MD 20684
(301)995-6450

James.McNatt.ctr@Navy.mil

Tom Gaudette
The MathWorks
3 Apple Hill Drive
Natick, MA 01760
(508)647-7759

Tom.Gaudette@Mathworks.com

Abstract - Traditional Test Program Sets (TPS) for Naval Automated Test Systems operate in a serial fashion, collecting and analyzing data one unit at a time. Teradyne's Ai-710 Analog Test Instrument, which is currently being installed into the U.S. Navy's premier test system, CASS, allows for vectors of Input/Output data to be emitted and collected simultaneously. The MathWorks, under contract from the U.S. Navy, designed and implemented test software using MATLAB that controls the Ai-710. This software enables TPS developers to create Vectorized Test Programs Sets (VTPS) in which vectors of I/O data are sent, collected, and analyzed in parallel, increasing TPS efficiency, simulating real-world circumstances, and facilitating the use of the Ai-710.

BACKGROUND

Consolidated Automated Support System (CASS) is the U.S. Navy's premier avionics Automated Test System (ATS). It is a large integrated tester consisting of analog, digital, RF, high-power, and electro-optical instrumentation, power supplies, an operating system, a general-purpose interface, and an interactive user interface. It is semi-autonomous, allowing for various levels of user control. In its primary mode, Naval Avionics Technicians (AT) use it to evaluate, diagnose, and assist in the repair of mission critical electronics equipment for Naval airplanes and helicopters.

Potentially malfunctioning avionics assemblies are identified on an aircraft by a built-in-test (BIT) or by an organizational level (O-level) maintenance exam. These units are removed from the aircraft and taken to the CASS shop for more in-depth

testing. For each unit under test (UUT) or one of its subassemblies, there are corresponding Test Program Sets (TPS) run on CASS, which are sets of functional and diagnostic tests that determine whether the UUT is working properly or malfunctioning, and if it is broken, whether it can be repaired onsite by the Naval AT.

TPS's are designed by government or contracted engineers. TPS engineers use any and all readily available UUT information and their knowledge of electronics to design an effective TPS. However, they are limited to the use of CASS station hardware and software. Legacy station hardware has had a limited number of sourcing and measurement channels. For instance a CASS station had only two arbitrary waveform generator channels and two digitizing channels. More significantly, the programming language in which legacy TPS's are coded, ATLAS, only allows for a single measurement or comparison to be preformed at a time. These limitations constrained TPS design to a linear, step-by-step structure.

PARALLEL TESTING WITH ANALOG INSTRUMENTATION

Parallel analog test replicates the situation a multi-input/multi-output (MIMO) UUT would experience in its 'natural habitat,' the aircraft. By stimulating and measuring some or all of the channels at once, the test designer can observe inter-channel relationships and analog state-based behavior, which would be undetectable though traditional one channel at a time testing. It also increases TPS speed and efficiency by accomplishing several measurements at once.

Recently, the U.S. Navy has integrated the Multiple Analog Capabilities (MAC) kit into its CASS stations. The MAC kit consists of three Teradyne Ai-710 cards and one four-module Bi-410 card. Now, instead of two ARB channels and two digitizing channels, it has 96 of each. A single Ai-710 card consists of 32 channels; each channel containing the functionality of a Function Generator, an Arbitrary Waveform Generator, a Signal Digitizer, a DMM, a Timer-Counter, and a Limit-Detector. Not only can a channel be programmed with the functionality of those instruments at any given time, the channel can be multiple instruments simultaneously, including both source and measure instruments. For example, Channel 1 could source a square wave and also digitize the same square wave. A complex triggering scheme allows for some or all of the channels to be triggered to perform their programmed functions simultaneously, creating a 32 channel parallel analog test instrument (ATI).

Unfortunately, most ATE engineers have not taken advantage of the new parallel capabilities and efficiencies available in CASS. A few TPS engineers invoked the Ai-710 as an instrument capable of saving TPS run-time by reducing instrument switching. However, most engineers have only used the 96-channel resource recently installed in CASS in legacy-style mode – one UUT channel per measurement.

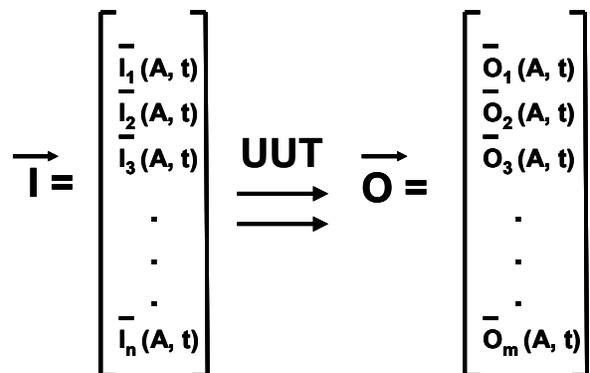
The parallel capabilities of the Ai-710 were not being used for two reasons: 1) the traditional paradigm of step-by-step testing, and 2) the lack of software to support parallel analog tests. Teradyne's *VXIplug&play* Ai-710 includes a multi-level Application Programming Interface (API) that exposes the user to the wide variety of instrument functionality and its complex triggering scheme. The Ai-710 API is quite sufficient to code a completed parallel analog test, but it is awkward to use to develop that test especially for a test designer unfamiliar with the Ai-710 and parallel analog testing. Teradyne also created an Ai-710 Soft Front Panel (SFP), which exposes a first-time Ai-710 user to the multiple-instrument per channel architecture and complex triggering arrangement, but this too was cumbersome to use in real parallel TPS design. In order to make full use of the Ai-710 and facilitate the development of parallel analog measurements, the U.S. Navy with the assistance of the MathWorks consulting team developed the MATLAB-based Vectorized Test Program Set (VTPS) software.

MATLAB is a programming language and algorithm development environment founded in matrix variables. Tremendous amounts of data can be collected, manipulated, analyzed, and visualized using matrix math and MATLAB software. In addition to its built-in engineering and math functions, MATLAB offers a constantly expanding set of functions, tools, and data analysis libraries called 'Toolboxes.' Of particular interest to the VTPS project was MATLAB's Instrument Control Toolbox. The Instrument Control Toolbox with its built-in ability to communicate with *VXIplug&play* drivers made configuring and controlling the Ai-710 simple, allowing users to easily send and receive data from the Ai-710.

VECTORIZED TEST PROGRAM SETS

After establishing a connection between the Ai-710 and MATLAB, and verifying that MATLAB can control and transfer data to and from the Ai-710, we set about writing software that would realize the potential for parallel analog tests inherent in both the Ai-710 and MATLAB. The conceptual test method that we developed was termed Vectorized Test Program Set (VTPS). In a VTPS, the input and output data collected from a UUT is 'vectorized' in two senses: a) each input or output channel is a vector of data in amplitude and time, and b) each test consists of a set of input vectors (**I**) and a corresponding set of output vectors (**O**). Letters in bold represent vector variables. See Figure 1.

Figure 1 – Conceptual Representation of Input and Output Data for Vectorized Test



The output vectors will be a function of the UUT and the input vectors. The output as well as the

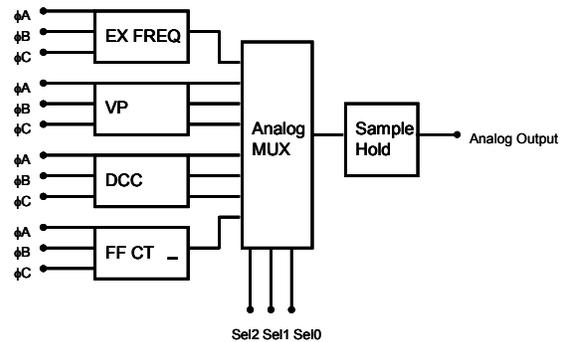
input vector sets can be compared to limits in time and amplitude to create performance or diagnostic tests. And order pairs of input and output vectors sets can be used to develop models of the UUT.

In order to make a generating VTPS practical and easy, we created a MATLAB-based graphical user interface in MATLAB called the VTPS Development Tool (VDT). A VDT user maps UUT input and output pins to Ai-710 channels, determines the UUT input signals, decides the data collection rate and test run time, and triggers the test to start. Both the input and output signals are digitized by the Ai-710, and the selected sets of data are displayed in plots with the same time axis. The user can import the vectors of I/O data into MATLAB for further analysis, or can create a test within the VDT using its criteria function.

In order to demonstrate the capabilities of VTPS and the VDT, we worked with a demonstration UUT, the A7 Analog Buffer Circuit Card Assembly (CCA), a subassembly of the Generator Control Unit for an AV-8B Harrier. The A7 routes conditioned AC power supply information to a microprocessor CCA upon request. The analog input signals to the A7 are 400 Hz 3-phase AC signals, which undergo various signal conditioning to extract information such as peak voltage, DC voltage level, frequency, and current limit. This information is stored as voltage levels, which are meaningful to the microprocessor CCA. The microprocessor selects which information should be routed to the output by switching an analog MUX via digital TTL inputs. Although there are

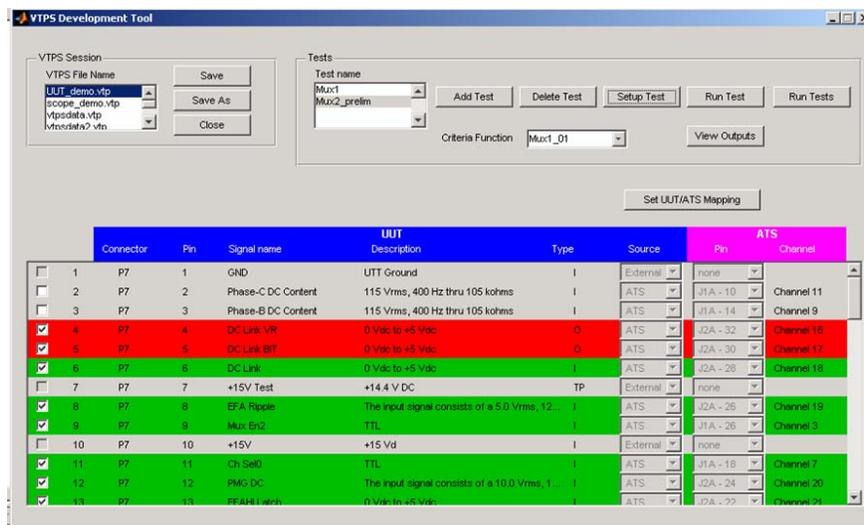
several other inputs and outputs, the A7 channels of interest are the analog power supply inputs, the digital switch control inputs, and the single analog output to the Microprocessor CCA. A conceptual block diagram of the A7 can be seen in figure 2. Some additional inputs, outputs, and channels are not shown in figure 2 for simplicity.

Figure 2 – Block Diagram of the A7 Analog Buffer



The first step in creating an A7 VTPS with the VDT is mapping A7 input and output pins to Ai-710 channels based on the interface connector from the UUT to the instrument. From this point on, each A7 channel is referred to by its UUT channel name rather than its Ai-710 channel name. For example, I would setup a signal input on ϕB VP instead of channel 17. Figure 3 is a screen shot of the VDT showing the mapping of several analog buffer channels.

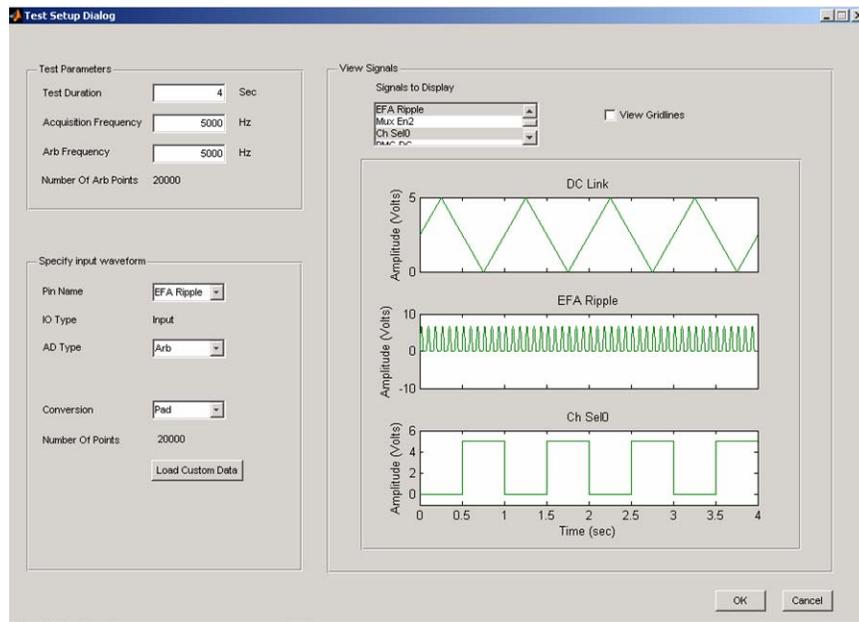
Figure 3 – VTPS Development Tool Showing UUT Channel Mapping



Next, we set up a vectorized test for the A7 Analog Buffer by selecting the relevant UUT input channels to which we wanted to source to and the relevant UUT output channels from which we wanted to collect data. In the *Test Setup* window we set the source signals, the sample rate, and the test run time. The source selection consists of standard function generator signals including DC, sine, square wave, and ramp wave, analog arbitrary waveforms, or digital arbitrary

waveforms. Figure 4 shows a screenshot of the *Test Setup* dialog box. It is important to note here that all the signals are setup on the same time axis. They may have different frequencies, but they are all sampled at the same rate and run for the same amount of time. This means that for all the *I/O* amplitude data vectors there is a universal time vector. Once the test has been setup, it is run with a push of the virtual *Run Test* button.

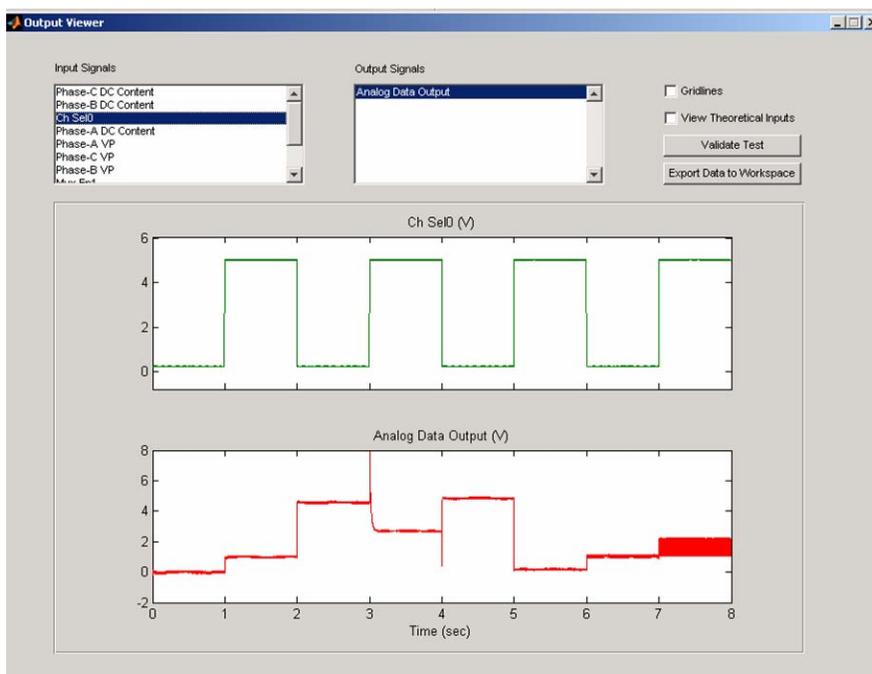
Figure 4 – Configuring the Test in the VDT *Test Setup* Window



After the test is complete, we can view the input and output signals on the universal time axis in the *Output Viewer* window as seen in figure 5. One or more inputs or outputs can be viewed simultaneously. The input signals seen in this window are not the same as the inputs seen in the *Setup Test* window, although they should be very similar. The inputs displayed in the *Setup Test*

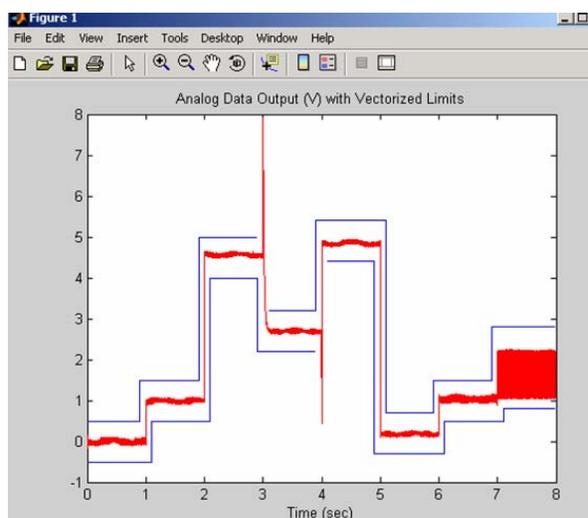
window are mathematical plots of the signals that the Ai-710 will be setup to source. However, the inputs displayed in the *Output Viewer* window are plotted digitized data, which has been sampled at the same rate as the outputs. This utilizes a very unique feature of the Ai-710, the ability to source and measure on the same channel simultaneously.

Figure 5 – Analyzing and Visualizing the Results of a VTPS



Once a baseline of input and output data has been extracted, we can use what we have learned to create tests using vectors of limits. The vectors output data can be compared to a high limit vector (**HL**), a low limit vector (**LL**) or both. For example: If $HL > O > LL$, then pass, else fail. In MATLAB, this is a single comparison, rather than looping through an array of data as one would do in other programming languages. Because data is also collected on the input channels, a similar comparison can be performed to verify that the input is within specification or to provide additional diagnostic information. This comparison can be coded in MATLAB and loaded into the VDT as the *Criteria Function*. Figure 6 is a plot of an output from the A7 and its high and low limit vectors. The MATLAB lines of code for this *Criteria Function* are seen next to it. Coding a *Criteria Function* into the VDT turns a vector collection of data into a genuine Vectorized Test Program Set.

Figure 6 – A Comparison of Acquired Test Data vs. High and Low Test Limits



```

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

```

The vectorized data-collection strategy also provides an excellent framework for a mathematical or data-based model of the UUT that can be used for prognosis or in a higher-level system model. Because the input and output data

is sampled at the same rate, the time vector can be factored out of either side. The resulting output matrix is a function of the input matrix. See figure 7.

Figure 7 – A Mathematical Model of a UUT

$$f_{\text{UUT}} \left(\begin{bmatrix} I_{11} & I_{12} & I_{13} & \dots & I_{1k} \\ I_{21} & I_{22} & I_{23} & \dots & I_{2k} \\ I_{31} & I_{32} & I_{33} & \dots & I_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & I_{n3} & \dots & I_{nk} \end{bmatrix} \right) = \begin{bmatrix} O_{11} & O_{12} & O_{13} & \dots & O_{1k} \\ O_{21} & O_{22} & O_{23} & \dots & O_{2k} \\ O_{31} & O_{32} & O_{33} & \dots & O_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ O_{m1} & O_{m2} & O_{m3} & \dots & O_{mk} \end{bmatrix}$$

Depending on the UUT, several iterations of data collection will provide enough information to generate a mathematical function or a matrix of mathematical functions representative of the UUT. Or, if no mathematical approximation can be found, the fast and repeatable VTPS will allow a user to collect and log sufficient sets of data to create a look-up table of outputs for certain sets of inputs. MATLAB and several of its toolboxes would be very useful in either of these UUT modeling approaches.

CASS to work in their accustomed environments, hopefully facilitating the use of Vectorized Test Program Sets in production TPS on CASS, and lay the groundwork for VTPS on other automated test systems.

CONCLUSION

Although it occurs frequently on analog electronics assemblies under operation, parallel analog stimulation and measurement is rarely used today to test those UUTs. The ATS world is constantly moving in the direction of decreased size, shorter run-times, and increased effectiveness. The recently integrated MAC kit in CASS added 96 channels of parallel analog stimulation and measurement capability in three C-sized VXI cards. However, a tester is only as good as the tests it runs. Without the experience or software to perform a parallel test, the potential advantages of the Teradyne Ai-710 Analog Test Instrument will not be fully realized. A vectorized approach using the Ai-710 could dramatically decrease TPS run-time as well as provide a novel testing technique. The VTPS Development Tool was produced in Phase I of the U.S. Navy/MathWorks Vectorized Test Program Set project, assisted in the development of a demonstration-quality VTPS and successfully proved that a VTPS can be created with the Ai-710. MATLAB test software was chosen to develop the VTPS Development Tool due to its parallel measurement and analysis capabilities and support for controlling Teradyne's Ai-710. The future of VTPS depends on making it accessible to TPS developers and subsequently to the United States Naval Fleet that it was designed to support.

The VTPS program we created was project to demonstrate the feasibility of VTPS in general and the potential for MATLAB and the Ai-710 to run a VTPS. This program has not yet been used on the "big blue" benches of CASS; it was run on an Ai-710 in an external VXI card cage connected to a stand-alone PC. The VDT controlled one Ai-710, and all signals sourced and measured on the A7 demo UUT were solely from the Ai-710. This provided significant limitations in producing a veritable VTPS on the A7 Analog Buffer. The Ai-710 has 32 I/O pins, but the A7 has up to 56 pins that could be wired to an instrument at once, and the Ai-710 can source a maximum of 12 V, while several A7 channels required 115 V.

A second generation VDT is being planned. In this next stage of the VTPS project, we are planning to remedy some of the limitations listed above and improve the visualization capabilities, the criteria function, as well as other features of the VDT. In VTPS Phase II, we would like to load the VTPS Development Tool onto a CASS station, and use it to control all three Ai-710's as well as other CASS assets. This would provide VTPS developers all of the instruments available on