Rapid Electro-Optical (EO) TPS Development in a Military Environment

James McKechnie[a], Alan Irwin[a], Thomas Gauntner[b]

[a]Santa Barbara Infrared, 30 S Calle Cesar Chavez, Suite C, Santa Barbara, CA 93103
[b]US Army - ARDEC, Picatinny Arsenal, NJ 07806

ABSTRACT

Santa Barbara Infrared, Inc. has deployed IRWindows as an Electro-Optical test development and execution environment for military Test Program Sets (TPS). Advantages of TPS development for EO systems in the IRWindows environment are seen compared to the TPS development in ATLAS. The advantages of the IRWindows environment are:

1. Faster learning curve (graphical user interface is easier than test line interface)
2. Faster TPS development time (real time changes and asset control interface allows for faster development)
3. Asset control panel allows user to control assets real time and monitor all asset functions during development
4. Unit Under Test (UUT) image viewer allows user to set test parameters like Region of Interest more easily and more precisely
5. Continuous mode tests (like MTF allows user to real time adjustments)
6. Open architecture for test modifications

This paper will outline the details of how these advantages are utilized and how not only development time is decreased but also how test execution time can be minimized making traditionally long TPS run times on EO systems more efficient.

Keywords: IRWindows4, TPS Development, E-O Testing

1. INTRODUCTION

Test Program Sets (TPS) encapsulate the hardware, processes, and instructional material necessary for testing specific devices and assets deployed by the military. For this discussion, the hardware consists of the interface devices necessary to connect the unit under test (UUT) to the designated automatic test equipment (ATE) and the processes are captured in the software executed on the ATE. TPS development has traditionally required long development times and in the case of electro-optical (EO) UUT's, long execution times.

In a previous paper[1], the various models of test development were discussed along with their relative advantages and disadvantages. In this paper, a practical example compares the development of a TPS using an older, ad-hoc model to the development of a replacement TPS using a modern system with an alternative development model (UUT Centric). Our goals are the reduction in TPS development time and faster TPS execution time.

IRWindows™ 4 is the latest generation of SBIR's test control application program. The software provides a framework for individual sensor test development, test sequence development, and automated factory testing. It is used to develop a TPS that will replace an older TPS currently in use. The development time and the time for executing the TPS are compared to the original development and execution times of the existing TPS.

Infrared Imaging Systems: Design, Analysis, Modeling, and Testing XXIII, edited by Gerald C. Holst, Keith A. Krapels,
Proc. of SPIE Vol. 8355, 83550C · © 2012 SPIE · CCC code: 0277-786X/12/$18 · doi: 10.1117/12.924992

2. IRWINDOWS DEVELOPMENT ENVIRONMENT

To utilize the EO TPS environment in IRWindows, a set of sequences (meaning a list of EO tests) were created in IRWindows and saved in the IRWindows application. All sequences were created and validated in IRWindows. Once this integration was completed, execution of these sequences was executed from National Instruments' TestStand using function calls. This technique allowed a user to use the EO friendly graphical interface in IRWindows.

An example of the easy to use graphical interface is below (Figure 1). This screenshot shows the list of canned tests (called sequences in IRWindows) the TPS Developer can choose from. The TPS Developer selects the applicable test and adds parameters associated with the UUT.

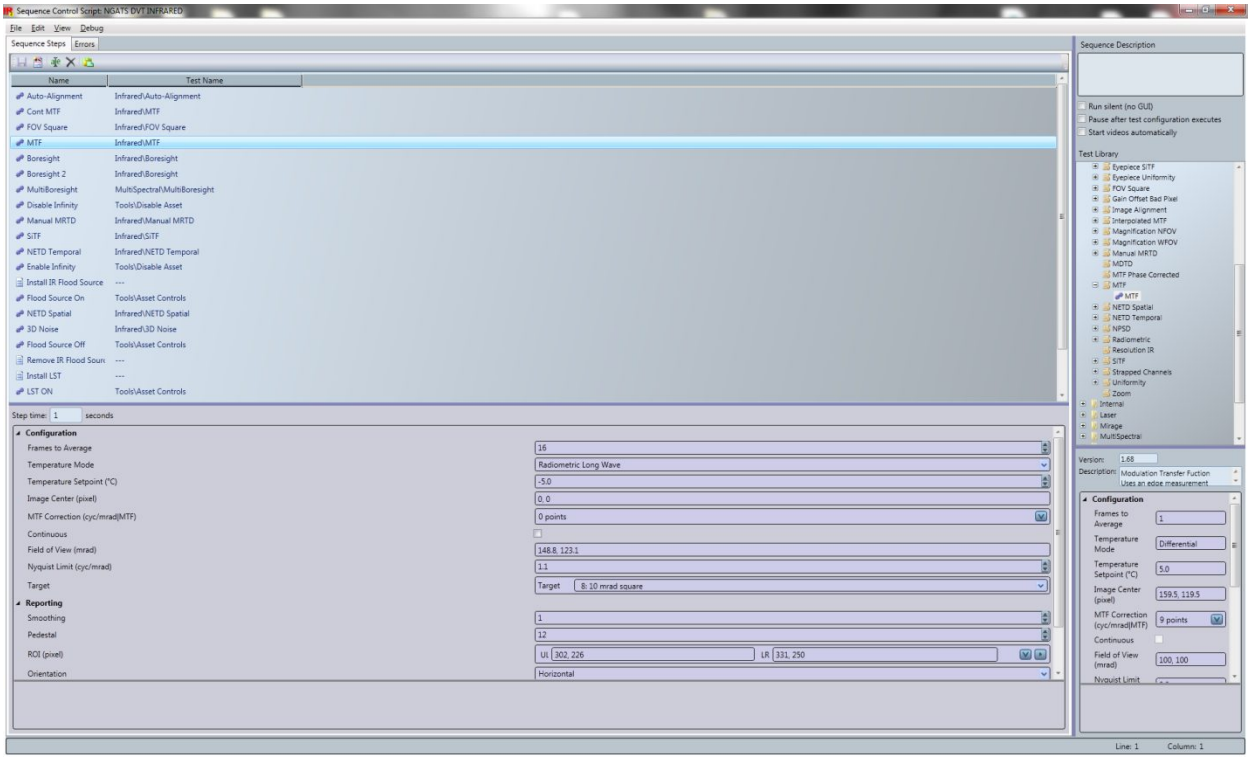

Figure 1: IR Windows Sequence Editor

The ATLAS interface currently utilized in the IFTE V5 is a text interface. Parameters sent to the hardware to control target projector functions and video parameters to fine tune the region of interest to perform the EO tests are provided in a text format. Feedback to the TPS developer on where the target appears and detailed portions of the video the tests are being performed are not available. To "fine tune" the particular sequence the operator is required to change parameters in the test, rerun the test, and evaluate the results. There is no intermediate feedback during the testing to help the developer.

IRWindows tests provide a more intuitive graphical interface to the TPS developer providing a real time video capture of the UUT scene allowing the developer to fine tune important test parameters. This feature significantly reduces

development time. This also allows the TPS developer to change parameters on video previously captured also reducing development time.

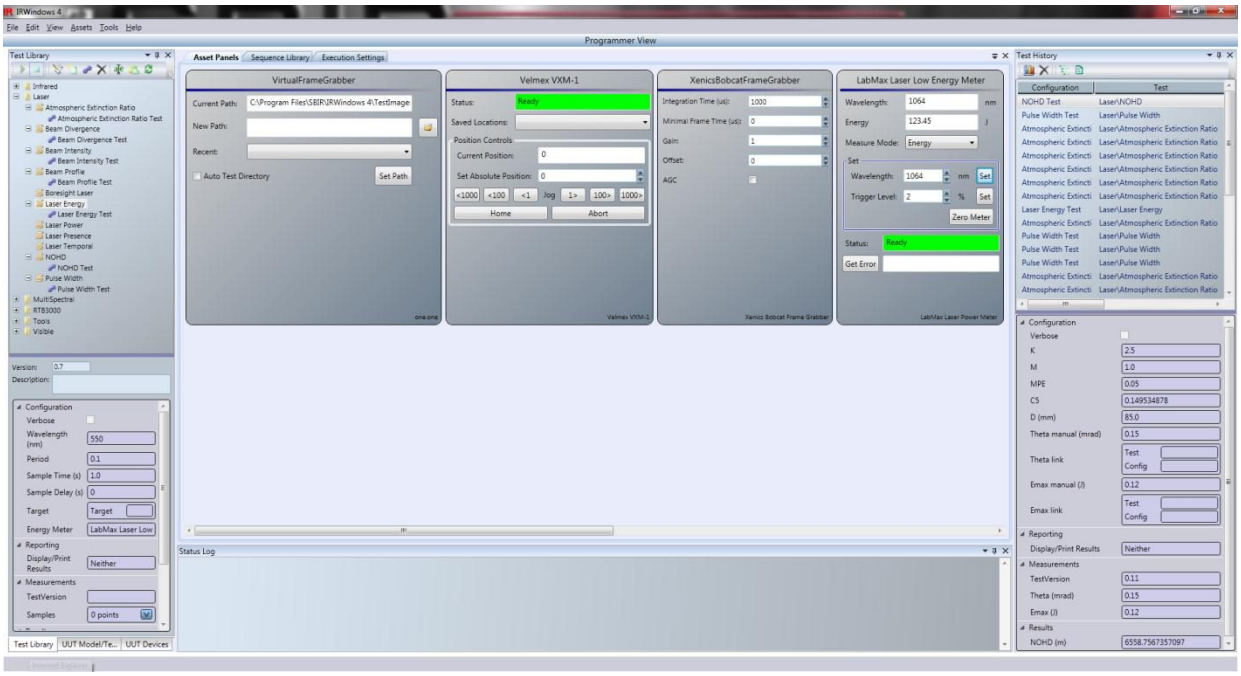

Figure 2: IRWindows Asset Control Screen

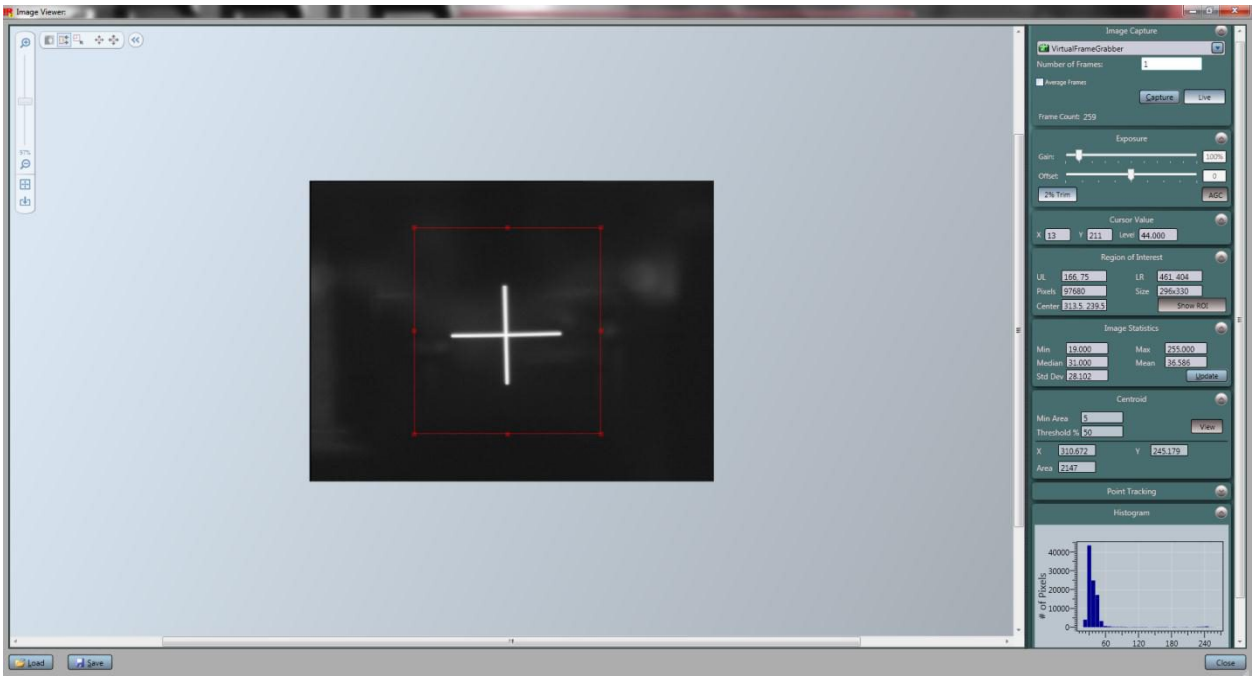

Figure 3: IRWindows Screenshot of an Image Capture

3. EXAMPLE

Modulation Transfer Function (MTF) is used to find the position of best focus of an infrared imaging system. IRWindows allows the developer to view the captured image, fine tune the region of interest the MTF is being calculated as well as other parameters (like frames to average and image filtering) to best fit the MTF value. Additionally, a continuous mode is offered allowing the operator to adjust UUT parameters (like focus) to maximize the MTF value. This significantly decreases test development time compared to simple text inputs. Also, the TPS developer has the option of including the continuous mode in the TPS, allowing the operator to best focus the system during TPS operation and then executing a MTF for captured results.

Examples of a screen shot showing the image viewer with the target for MTF is below (Figure 4).

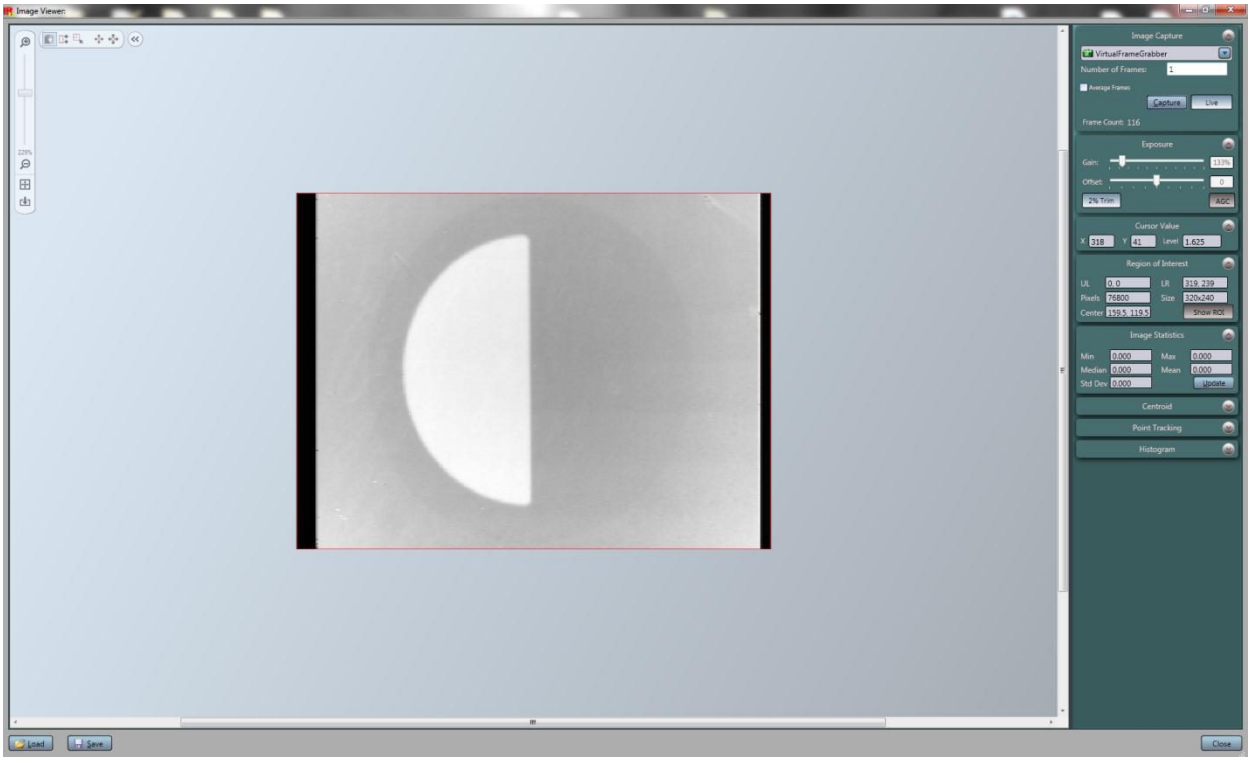

Figure 4: Image Viewer Screen Shot with MTF Target

The resultant MTF is shown below in a graph (Figure 5). The TPS developer can view the results in either a graph or table format and make changes on the captured image to fine tune the results.

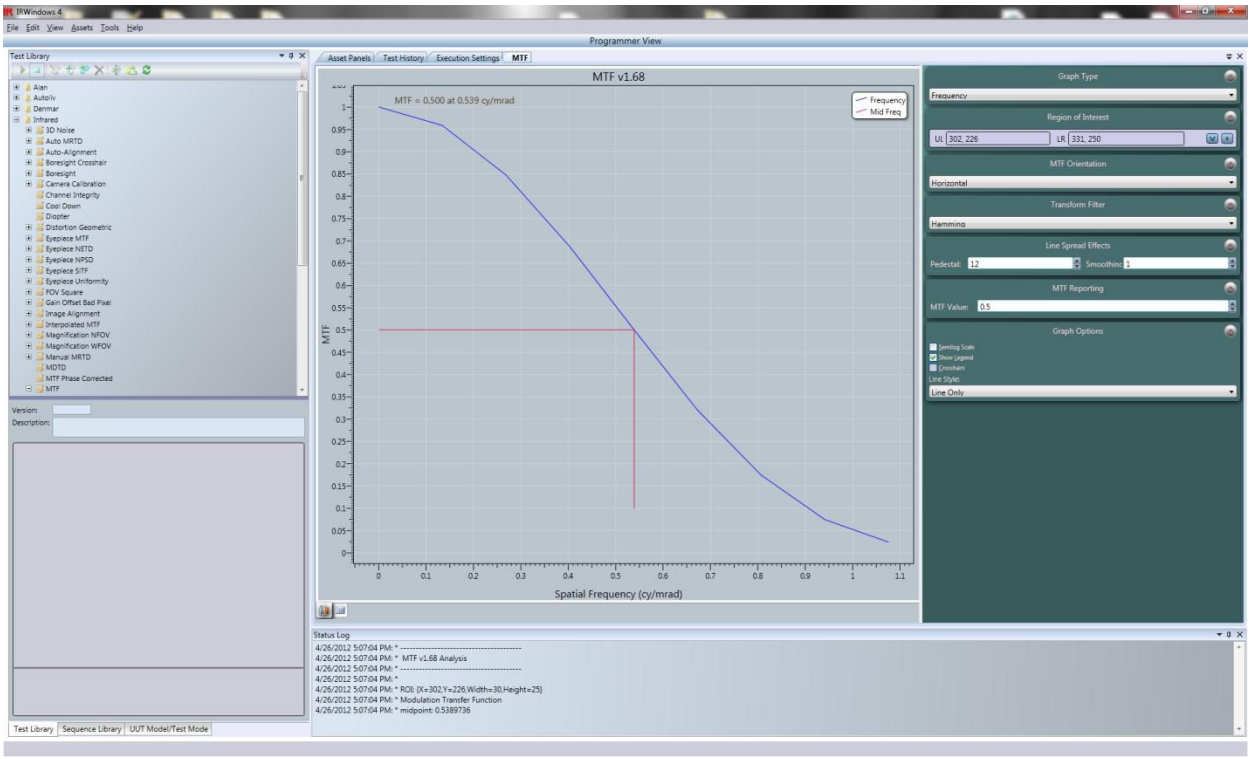

Figure 5: IRWindows Screenshot of MTF Results

Simply put, a TPS developer is able to set up a target projector set of parameters, capture UUT video, perform a test, and then fine tune important parameters with this information. Once this fine tuning process is complete, the test can be validated. This approach is significantly more efficient than the ATLAS approach where a test is performed, executed, results returned with no feedback of how the values were calculated.

4. ADDITIONAL FUNCTIONS

There are two additional functions that IRWindows provides. One is continuous mode. IRWindows has multiple tests that can be utilized in continuous mode. The 2 most important of these tests is a continuous boresight and continuous MTF. These tests allow a TPS user to fine tune UUT parameters (such as focus in the MTF case) in real time while providing test result feedback. This feature allows flexibility in both TPS development time as well as TPS run time.

Another feature in IRWindows is the open architecture of the test algorithms. IRWindows allowed a skilled developer to modify the test algorithms within IRWindows. Modifications of IRWindows algorithms does not affect the base line test, it simply creates a duplicate test with the parameters changed by the user.

A screen shot of the open architecture is below (Figure 6).

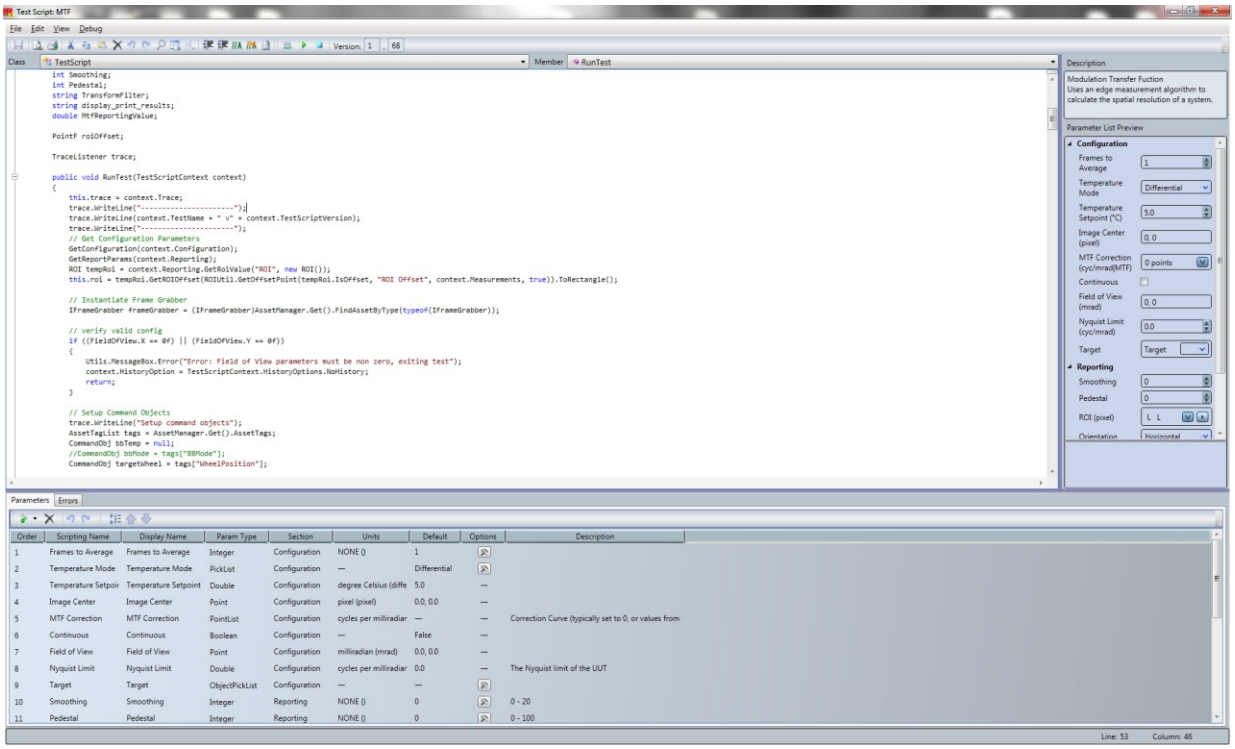

Figure 6: IRWindows Screenshot of Open Architecture

For Autotestcon on October of 2011, 5 tests were created using the NGATS test system under IRWindows control and testing the CROWS (Common Remotely Operated Weapon Station) TIM (Thermal Imaging Module) UUT. The EO TPS was developed in IRWindows and executed from TestStand with the results saved to a common SQL database.

Below is a screen shot of an EO TPS being executed remotely (Figure 7).

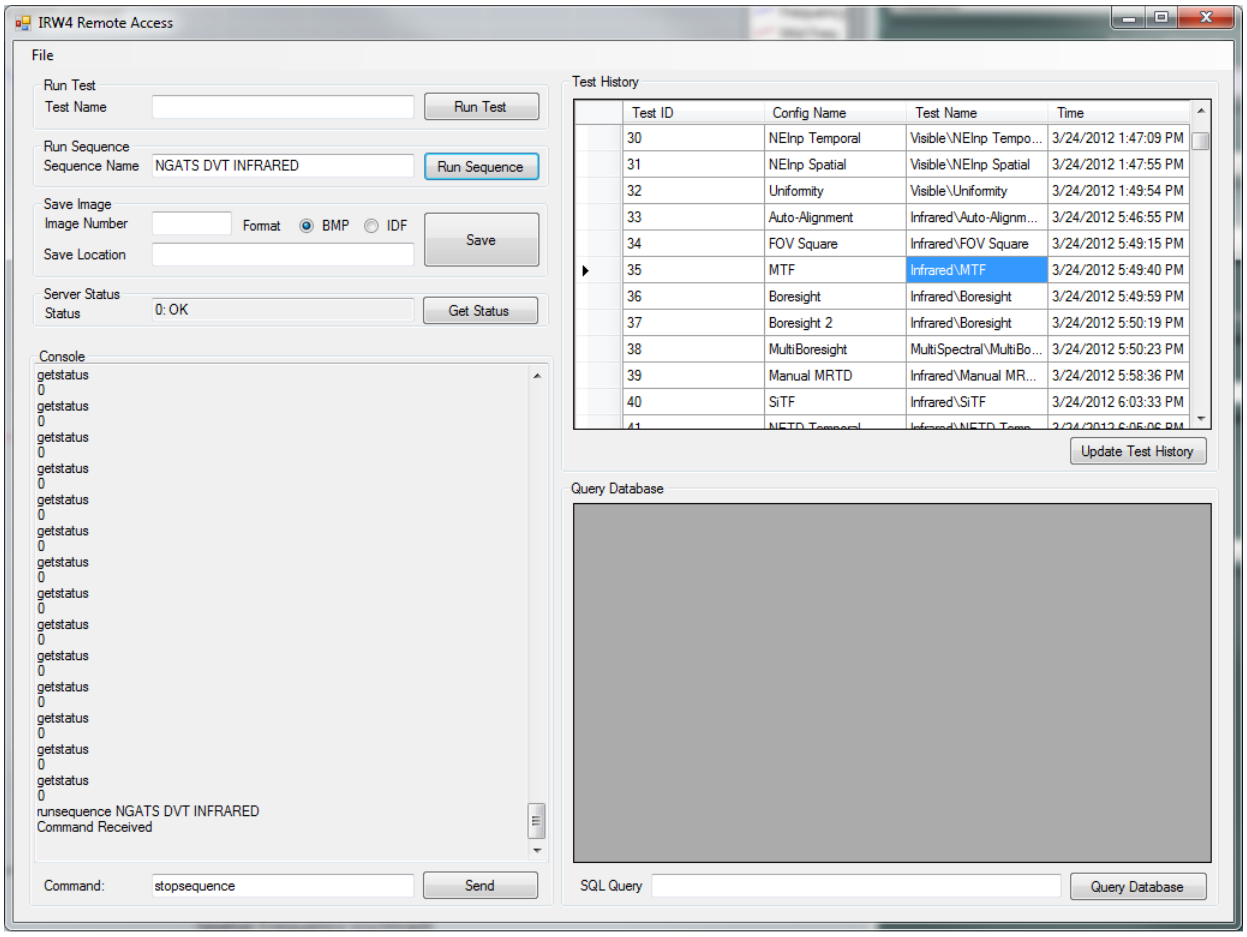

Figure 7: Remote Access Screenshot

5. DEVELOPMENT TIME COMPARISON

This section compares a TPS created on the IFTE V5 station under ATLAS control to a similar TPS created using IRWindows and executed via TestStand. There are 5 tests included in the TPS:

Boresight
Boresight – Continuous
Manual MRTD
MTF
MTF – Continuous

Below is a comparison of execution times on the NGATS station under IRWindows control compared to the IFTE V5 station under ATLAS control.

Table 1: Test execution time comparisonbetween IRWindows4 and IFTE V5.

Test	Execution time on a NGATS station under IRWindows control	Execution time on a IFTE V5 station under ATLAS control
Boresight	Approx. 45-60 sec	Approx. 2-3 min
Manual MRTD (with the caveat that it can take as long as you want it to)	Approx. 2 min	Approx. 5-6 min
Modulation Transfer Function	instant	Approx. 2-3 min

Note: Continuous Boresight and MTF aren't comparable since there is no correlating functionality on V5.

TPS Development Time comparison:

CROWS TIM TPS on IFTE V5 --> 9-12 months
CROWS TIM TPS on NGATS/IRWindows --> 5 weeks

TPS Runtime comparison:

CROWS TIM TPS on V5 --> 45 min
CROWS TIM TPS on NGATS/IRWindows --> 15 min (including additional testing)

6. SUMMARY/CONCLUSIONS

By taking advantage of updated test development models and an environment designed and developed to operate EO tests, the TPS developer working with IRWindows has many advantages:

1. Faster learning curve (graphical user interface is easier than test line interface)
2. Faster TPS development time (real time changes and asset control interface allows for faster development)
3. Asset control panel allows user to control assets real time and monitor all asset functions during development
4. Unit Under Test (UUT) image viewer allows user to set test parameters like Region of Interest more easily and more precisely
5. Continuous mode tests (like MTF allows user to real time adjustments)
6. Open architecture for test modifications

We were able to demonstrate these advantages in the case of the CROWS TIM TPS development. IRWindows 4 running on the NGATS EO interim solution test station shows significant savings in TPS development and execution time. Additionally, this application was able to take advantage of IRWindows remote interfacing so that NI TestStand could serve as the executive controlling the overall operation.

7. REFERENCES

[1] Irwin, A., LaVeigne, J. and Nehring, B., "A common architecture for TPS development", Proc. SPIE 8014, (2011)