

Automated Testing of Actuators for Medical Exam Tables

Author:

Gregory C. Cala, Chief Technology Officer, Data Science Automation
Ben Rayner - Senior Measurement and Automation Architect, Data Science Automation

NI Products Used:

NI 9201 8 channel, 12-bit, +/-10VDC, 500 kS/s, AI Module
NI 9225 3-Ch +/-300V Analog Input
NI 9227 4 channel current input, 5Amp, ISO, 50k, 24bit
NI 9474 8-Ch 24 V, 1us, High-Speed Sourcing DO
NI 9234, 4 Input, 24-Bit, 51.2 kS/s, SW Selectable IEPE & AC/DC
cDAQ-9178, CompactDAQ chassis (8 slot USB)
NI 9476 32-Channel 24 V, 500 us, Sourcing Digital
NI LabVIEW 2010

Categories:

Advanced Control Systems
Automated Test
Life Sciences

The Challenge

Develop an application to monitor and control the testing of test linear actuators as they exit a production line. The application must be enforce safety requirements during all phases of operation, detect out-of spec conditions and perform graceful recovery.

The Solution

DSA created an adaptive solution that implemented all the client requirements while allowing the easy addition of future client devices to the test procedure.

Introduction

Data Science Automation (DSA) is the premier National Instruments Alliance Partner. DSA integrates commercial off-the-shelf (COTS) components from automation technology vendors to create custom, adaptive automation solutions for a diversity of research, manufacturing, government and business operations to:

- acquire, analyze, present and manage data
- design, simulate, test and validate products
- monitor, predict, control and optimize processes
- invent, draft, prototype and build machines

for maximum productivity, quality, profit and understanding.

DSA is a certified member of the Control Systems Integrators Association (CSIA) and staffs multiple National Instruments Certified Training Centers with more certified LabVIEW Architects than other integrators.

DSA was chosen by the client because of our ability to compound their investment through an adaptive solution while empowering them through education. The client was also attracted to DSA because of our extensive certifications, responsiveness, diversity of experience and success with other medical device manufacturers.

Touch Panel Interface

A touch panel was chosen in order to allow operator input and control of the test station by users wearing gloves. Gloves worn to protect the operator's hands while handling the linear actuators often can introduce typographic errors when using a standard keyboard. A unique solution was implemented

by harnessing a commercial off-the-shelf (COTS) touch screen that effectively combines a graphical display with mouse functions while utilizing the graphical controls and indicators native to LabVIEW. This approach simplified the user interface requirements to simply plugging in the touch panel. No extra code was required to support the interface. This was leveraged after the initial deployment when it was determined that a 7-inch control panel was too small (barely visible in Figure 1) to allow viewing from a distance. The 7-inch display was replaced with a compatible 10-inch display, again with no code changes required to implement the change.



Figure 1. Test Station front view.

Safety Requirements

The safety of the operators using the test station was top priority when designing and building the system. A light-curtain and abort buttons were used to kill all power to the actuators and drives when the curtain was broken or the abort button actuated. The safety interlock circuits were also monitored by the PC via the NI-9476 to distinguish between actuator failures and safety considerations. An

interruption of the safety functions can be recovered by the operator but actuator failures are handled by the software.

Match load to DUT, Verify and allow Expansion

The test station is capable of using an assortment of loads to test the actuators. Heavy loads are available for the larger actuators while lighter loads can be used for the small units. To properly test an actuator, the proper load had to be used during testing. In addition to choosing the proper load required for an actuator, the application had to be able to verify the proper fixturing was used for the selected device under test (DUT). Additional sensors were used to ensure the DUT was fixtured correctly by verifying that only the hardware required for the DUT were absent from the storage rack. This check guaranteed that an under-sized pin would not be used to test a high capacity actuator.

Driven by the need to allow for new DUTs in the future, the application reads the required load and hardware requirements from file along with jpg images of the actuators themselves. The operator need only tap the image of the actuator to be tested on the touch panel and the application would adjust accordingly. New actuators could be added to the collection of testable DUTs by simply editing a configuration file. An additional set of checks were implemented to protect the integrity of the test station itself. Since some of the actuators are able to exert forces that could damage the test station, sensors were integrated that verified the proper set of weights was selected and the enabling hardware had actuated as required.

Control Actuator While Loading, Unloading and Testing

Once an actuator has been mounted in the test station and the safety checks have been verified, the operator can initiate the test by actuating a palm button. Once started, the loading of the actuator is carefully monitored using a LVDT capable of plus or minus 24-inches of travel that is coupled to the load bearing element of the test station. If the position signal from the LVDT does not change within a configurable time limit after starting the actuator (Figure 2), an error is declared and the actuator is retracted. This error is most often triggered by failing actuators or improperly fixtured actuators.

Provided the actuator assumes the load (as indicated by the LVDT reporting the carriage has reached the start position) testing can begin. Testing consists of cycling the actuator between the start and end points for a configurable number of cycles. Time limits were used to ensure the actuator responded to the motion requests properly. Similar checks were implemented to ensure safe unloading of the DUT after the test completed.

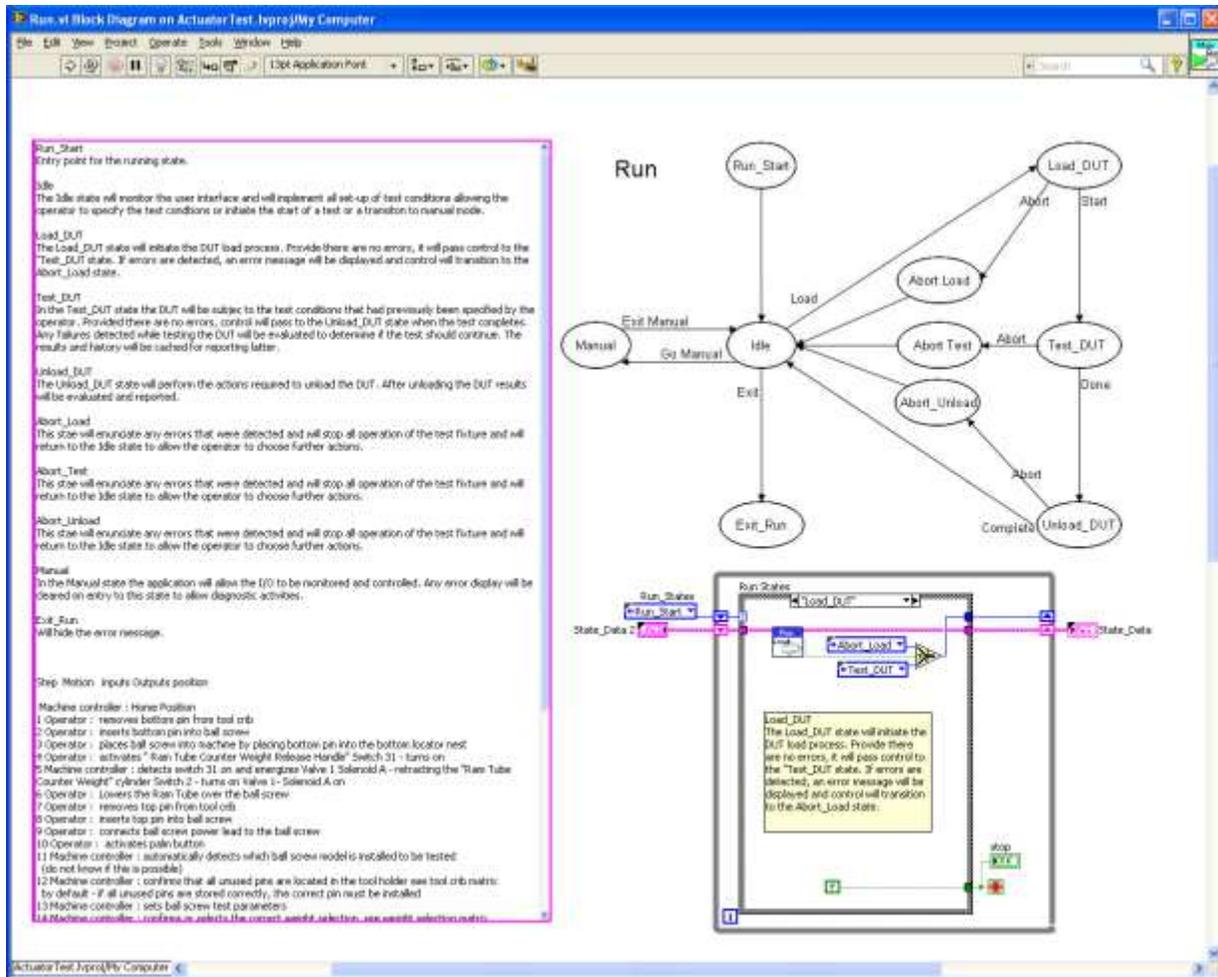


Figure 2. Run Mode code and design shows transitions from idle through loading and testing.

PC Based Application

To reduce project cost and allow for simplified upgrades in the future a standard Desktop PC (Figure 3) was selected as the platform to run the application. Since most of the hardware produced by National Instruments ships with drivers to configure, monitor, and control their hardware from a Windows based PC the integration time was minimized. LabVIEW was selected as the development environment to take advantage of the native multithreading capabilities and thereby greatly simplify the code required. Multithreading allowed the various code modules of application to execute independently e.g. the digital input logic only contained code related to acquiring the state of the Boolean sensors and related input signals. This modular approach yielded very clean code that was unencumbered by operations handled elsewhere.



Figure 3. Test station rear view shows fully populated Compact DAQ Chassis

Monitor Vibration

One of the less obvious checks performed during testing required determining how noisy the actuator was during operation. The amount of noise produced was monitored (using an NI-9201) by using a 3-axis accelerometer physically bonded to the base plate to which the actuators were coupled. Independent limits were implemented to allow production engineering staff to dictate the amount of vibration that was acceptable on a model by model basis. If during testing the limit was exceeded, the actuator would fail the test. The vibration signals were also saved as part of the report data to allow design engineers to evaluate various lubricants and components used in the actuator drive train and allow for product enhancements.

Monitor Efficiency

Knowing the amount of weight used for the actuator and the displacement of the load, the amount of work performed by the actuator was evaluated. During testing the current drawn by the actuator (NI-9227) as well as the applied voltage (NI-9225) allowed the application to determine the amount of energy used during testing. The ratio of the measured work to the ideal work calculated using the measured voltage and current provided an indication of the actuator efficiency. All of the measurements required for the efficiency calculation were logged to file to provide design engineering personnel the ability to evaluate how the efficiency varied for various parts of the test.

Log All Measurements

All of the analog signals acquired during testing are logged to a unique file for each actuator. New files are generated based on the actuator type and are maintained during testing. The file includes all of the signals while the test is active along with summary information that indicates the pass/fail status of all tests and includes meta-data describing the DUT tested. Mapping the drive to the corporate network allows the IT department to perform nightly backups as well as transferring test files for use by production and design engineering. The logging was implemented in two phases to ensure no data loss while also providing easily readable text files as the final reports. The first phase of the logging activity utilized binary files to record the measurements as quickly as possible. The speed requirement

was driven by the high frequency sampling used by the accelerometers to monitor vibration. After the DUT was unloaded the second phase was entered which read the data recorded as binary data and evaluated the limits as well as the maximum and minimum values recorded for all analog signals.

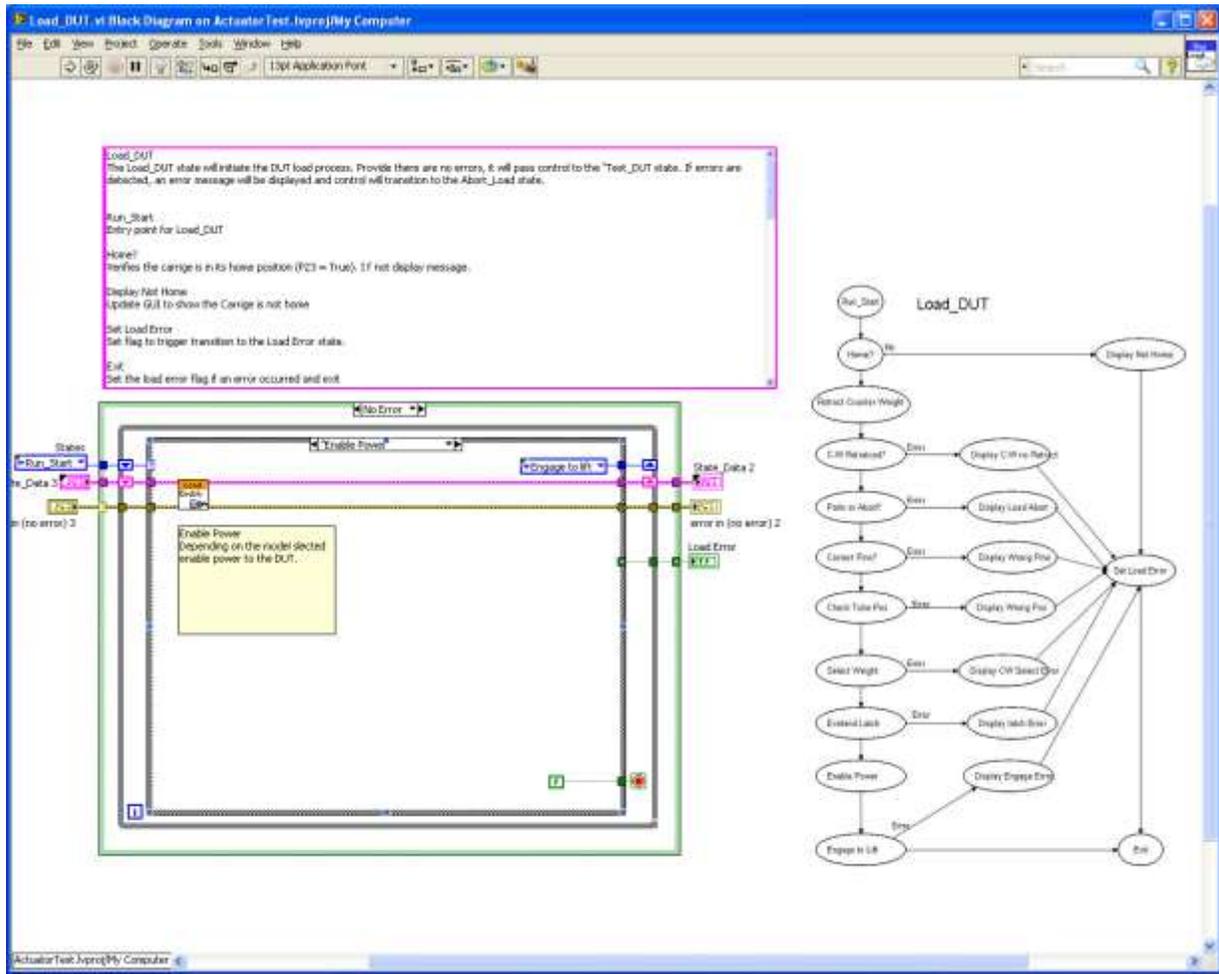


Figure 4. Load DUT code and design illustrates how LabVIEW's graphics can include documentation

Failure Recovery, Aborting Test and Manual Control

In an ideal world no DUT would fail, no test would be aborted and manual control would not be required. The test station was developed to operate in the real world and needed to provide functionality to allow operators to clear a failing DUT and restore proper operation. Wherever possible and safe, the application will detect failures and act to correct them. For example, if the load was not assumed by the actuator within a reasonable amount of time, the load attempt will be aborted and the test station restored to its idle state (load in home position all latches released) and the operator notified the actuator failed (see figure 4). Similarly, if a time-out is detected during testing, terminate further cycling attempts and initiate an unload sequence. In the event the unloading and failure recovery fails to complete properly, the operator can abort the operation manually. The manual operation permitted an actuator to be backed-off and allow them to be disengaged once the load was released safely. If all else failed, a hydraulic jack was built into the test station to lift the load and allow the actuator to be removed.

The manual control functions served another purpose aside from failure recovery scenarios. Manual mode allowed for functional testing of the many inputs and outputs of the test station. Once manual mode is invoked all of the digital states are clearly displayed and accessible to the operator. When any one of 15 sensors failing could prevent proper operation, the display of individual digital states are very helpful in determining the cause of the problem.

Serial Numbers

When an actuator passes, a serial number is assigned and associated with the report of the DUTs testing. The serial number is then printed on a part number label that is affixed to the actuator to clearly identify it as it passes through its life cycle. A small COTS label printer was integrated and used by the label generation code to produce the label. The serial number was then marked as used in the test station's configuration file allowing it to pick-up where it left off the last time an actuator was tested. The file also allowed production engineers to over-ride the increment of the serial numbers used during non-production testing.

Summary

Using LabVIEW and National Instruments hardware greatly simplified the work and reduced the time required to develop the application. The unique graphical development environment allowed the documentation to be included in the source code and served to ease development and troubleshooting. Knowledge transfer between the developer and those who will support the application on a long term basis was minimal. The ease of reading and changing the code permitted a graduate of LabVIEW Core I & II to send an email saying "I forgot to tell you about X, but I figured it out and fixed it myself."

Contact Information

Ben Rayner, 724-942-6330, www.DSAutomation.com