

Ultra Efficient Fuel Delivery System TestCell

By

Nils Gokemeijer
Project Consultant
Data Science Automation
United States of America

Category: Aerospace/Defense

Products Used: (MUST BE VERIFIED)

Realtime Control System

- PXI-8170 Running RealTime OS
- PXI-8211 Ethernet Module
- PXI-8421/4 RS485 interface card
- PXI-6030E MIO card
- PXI-6704 Analog Output Card
- PXI-1000B Chassis
- 5B01 16 channel Signal conditioning backplane
- 5B40 Signal conditioning units
- 5B41 Signal conditioning units

Data Acquisition and User interface system

- PXI-8176 Running Windows 2000
- PXI-8421/4 RS485 interface card
- LabVIEW 6.1 RT
- PXI-6040E MIO Card
- PXI-1000B Chassis
- SCXI-1001 12 Slot chassis
- SCXI-1102B 32 channel signal conditioning
- SCXI-1102C 32 channel signal conditioning

The Challenge: Develop a Jet-Engine Simulator and data logging system to test a newly developed Ultra-Efficient Fuel Delivery System. The test system needs to simulate the behavior of a jet engine flying at a particular altitude, a particular speed (mach-number), and with the power level angle in a certain position (throttle).

The Solution: Using LabVIEW RT we developed code to control Pump Drive Speed, Heat Exchanger valve position, Back Pressure valve position, and Actuation Flow valve position. Meanwhile, the LabVIEW application on the windows system displays and logs all 70 channels of data at 1000 samples/sec.

Abstract: Two National Instruments PXI systems are used to control and monitor the UEFDS testcell. The control loops on the RT system execute 100 times per second and reads thirteen input channels which are over sampled 10X. The RT system controls three valves and the pump drive speed. The LabVIEW for Windows Application on the second PXI system allows the user to monitor and log all 70 channels of data collected at 1000 samples/sec through the SCXI DAQ System, the 30 parameters collected on the RS485 network connected to the unit under test, and the status of the RT system.

The UEFDS Testcell In order to be able to test the newly developed fuel pump, a test system needed to be built. The system was to monitor and log all 120 channels of data, and provide deterministic feedback control to allow for accurate simulation of a jet engine flying at a particular altitude, speed and throttle. Data Science Automation was contracted to develop the software. The Windows system was used to monitor, display and log all 120 channels at 1000 samples/sec. In addition, the system would monitor and log results from the RT system send over by way of UDP packets and information send from the fuel pump by an RS485 link. An SCXI chassis was used for signal conditioning of all DAQ channels.

Deterministic Control Piggy-backed onto this system is the LabVIEW RT system, which monitors the 12 critical channels, also at 1KHz. Signal conditioning for this signals was provided by 5B series modules. The data from these channels is buffered and averaged 100 times per second. A hardware timed loop executes 100 times per second, recalculating and updating the four outputs. The math necessary for the simulation is all calculated in LabVIEW and contains, in addition to some elementary math, a formula node, four PID loops, 9 rate limiters, a derivative and 12 look-up tables.

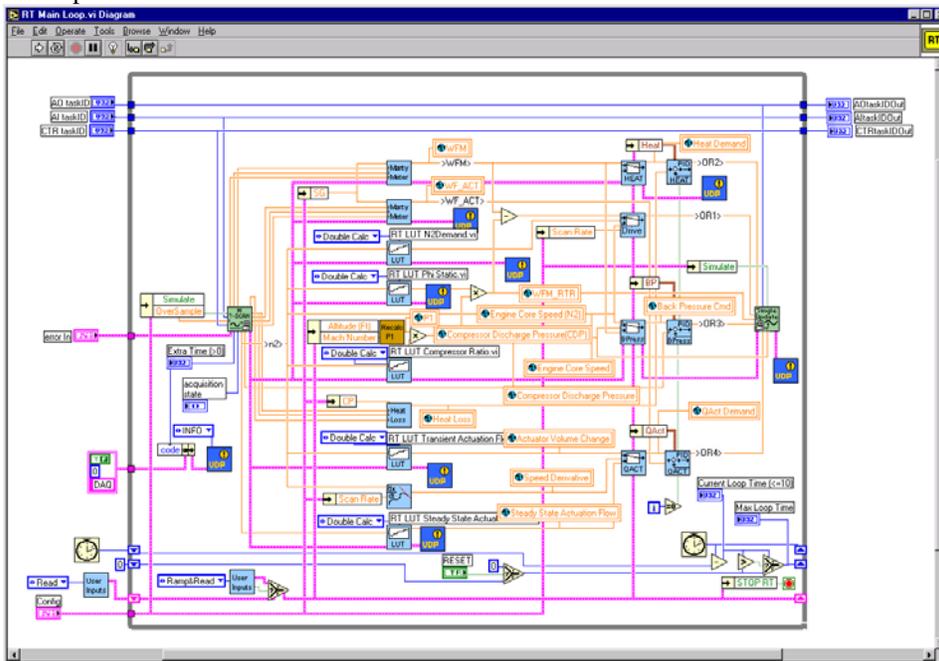


Fig 1: The real time loop that executes 100 times per second and recalculates the 4 output parameters.

The global variables are used to monitor interim values during the refinement phase and to pass data to a second loop. The second loop only executes 20 times a second and it passes critical information to the fuel pump over the same RS-485 network that the windows system uses to monitor the status of the fuel pump.

The system also contains a simulate mode, such that the application can run on a regular desktop computer for testing and development purposes.

PID Control It was a significant challenge to get the PID controls to work reliably and fast. Without over-sampling, the bit-noise in the analog inputs made the derivative unusable. The automatic calculation of the delta t in the PID VIs also caused problems. Although there was no hardware jitter, a software jitter of 1ms would cause a 10% change in delta t at our loop-rate of 100 Hz. Oversampling 10X and wiring in the delta-t really helped, although the noise on the signal still made use of the derivative term difficult.

RS-485 Communication The communication with the unit under test also posed significant problems. The data generated by the unit under test was too much for the RT system to analyze and log, while the commands needed to be sent to the unit deterministically. It was decided to use an RS485 network, such that the RT system could write commands to the unit under test, while the windows system would be receiving all the data to be logged. The lack of wiring and naming conventions posed additional challenges, who would have thought that Tx(+) on the RT system should be connected to Rx(-) on the testcell? RS485 proved to be quite robust however, even at the wrong baud-rate and the wrong polarity, we were still receiving data!

The powerful features of Measurement and Automation Explorer allowed us to test and calibrate each channel independently without the need for any customized software. Distinguishing hardware problems from software problems was therefore easy. The rapid development capabilities of LabVIEW in combination with the robustness of the PXI system allowed us to create a reliable system in a very short time-frame. The powerful testing and debugging capabilities of LabVIEW RT provided the assurance that we would be able to make the deadline and we did.