Machine Control and Data Acquisition of a Torque Converter Test Stand Using LabVIEW™

by
Anthony R Gaskell
Project Engineer
Quantum Controls

Category:
Automotive-Power Train

Products Used:
LabVIEW
SCXI™ I/O
PCI-MIO
NI-DAQ™

The Challenge:
Controlling a machine that tests automobile clutch torque converters by causing them to slip and/or lock-up, all while collecting pertinent data at speeds up to 2000 HZ and storing it to a PC hard drive for post-test analysis.

The Solution:
Using the inherent bitmap and boolean control features of National Instruments LabVIEW to create a graphical monitoring screen, as well as using LabVIEW with SCXI hardware to implement the control system for the machine and high-speed data acquisition.

Introduction
An Automotive component testing firm needed two automated test stands to perform Clutch Torque Converter Slip tests. LabVIEW combined machine control, data acquisition, data logging, and user interfacing into one compact software application. The SCXI hardware provided a high degree of flexibility in I/O setup, allowing for future addition of signals to/from the control system.

The graphical user interface (GUI) had to be suitable for operators who are heavily involved in mechanical design and possess limited Windows experience. LabVIEW was used to implement a graphical monitoring screen that features dynamic objects resembling mechanical equipment and pneumatic diagrams.

Machine Control
The torque converter machinery can be simplified conceptually to a closed fluid system that flows automatic transmission fluid (ATF). A pump circulates the ATF through the torque converter, and a proportional valve is used to control the amount of pressure/flow in the hydraulic lines. The ATF reservoir tank contains BOTH a heater and a heat exchanger cooler for controlling the ATF temperature. Throughout the machinery, flow meters, pressure sensors, and thermocouples are located to help monitor the ATF's vital statistics.

The torque (read from the converter using a 0-24mV load cell connected to an SCXI-1102 analog input channel) is directly proportional to the applied ATF pressure on the converter. The PID Control Toolkit for LabVIEW was employed to help determine the output to the proportional valve, which controls the system ATF pressure. An SCXI-1124 analog output module was used to supply the 4-20mA signal to the valve. The machine operator specifies a torque setpoint in the auto cycle recipe screen. This value is used as the setpoint for the PID VI (PID.VI), with the input torque used as the process variable.

The ATF temperature loop is more complicated than the simple torque/pressure control loop described above. The ATF temperature is affected by two different variable outputs – the tank heater and the heat-exchanging cooler. The temperature setpoint is given a deadband around it in which no action is taken by either the heater or cooler. When the ATF temp rises above the deadband, the heat exchanger cools the fluid down. Conversely, when the temperature drops below the deadband, the heater activates to warm the fluid up. Two separate PID loops were implemented for this single application task. The operator is given fields on a password-protected screen to change the deadband width of the temperature PID control loop, as well as change any of the parameters for EACH of the three PID loops. The screen shown in figure 1 is used by operators to change the given fields.

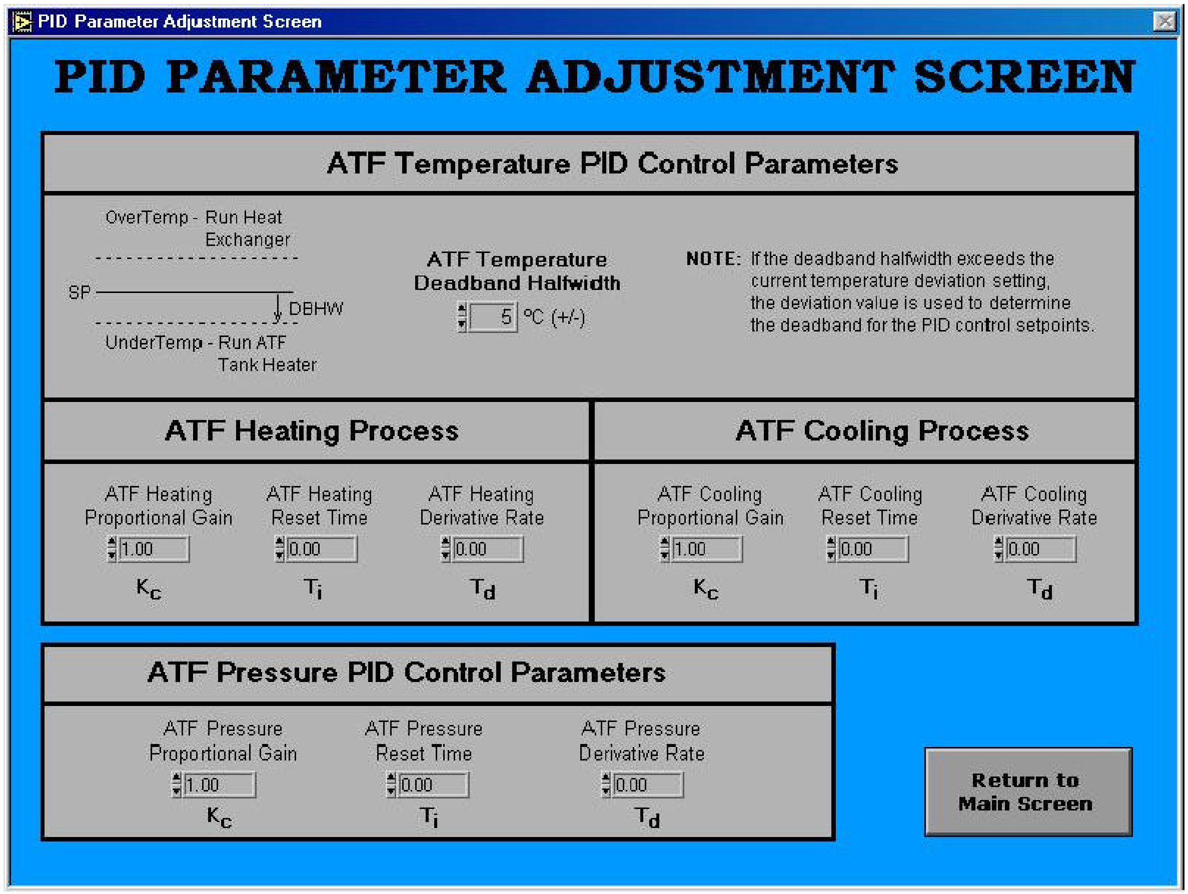

Figure 1. Operators have the ability to change PID parameters programmatically.

The ATF heater was variably controlled using a 4-20mA signal similar to that used for the pressure proportional valve. The heat exchanger, however, was controlled by a digital on/off output (powered from an SCXI-1160 output module). This output was variably controlled using a pulse-width modulation scheme given a reasonable duty cycle for temperature control.

Data Logging

The end user of the system wanted process data to be logged to the hard drive of the control PC. This data would then be viewed directly through a text editor or Microsoft Excel on the host computer. The data was not required to be stored in a format that could be read back by LabVIEW. Provisions were included, however, to put the data in a format easily read by LabVIEW if the end user decided to add that capability to the system in the future.

User Interface

The user interface front panels are used by operators whom possess a great deal of mechanical knowledge and limited PC/Windows experience. For machine monitoring the operator has a choice of five different full-screen strip charts to choose from, each representing a different pertinent grouping of analog input signals (i.e. torque and pressure plotted vs. real-time). Basic boolean push buttons are used to switch between screens. Standard data entry fields and boolean buttons are also used for the manual mode screen. The auto recipe setup screen uses the standard Windows dialog box for recipe filename selection, since all operators are familiar with this type of interface.

A separate monitoring screen was added to the interface that allows the more mechanically inclined operators a chance to get a good overview of the entire system and view the different states of equipment and signals. The screen resembles a pneumatic diagram of the overall torque converter machinery. The different elements of the diagram change color based on their on/off state or fault condition. The screen shown here exhibits the graphical monitoring screen as viewed by the operators.

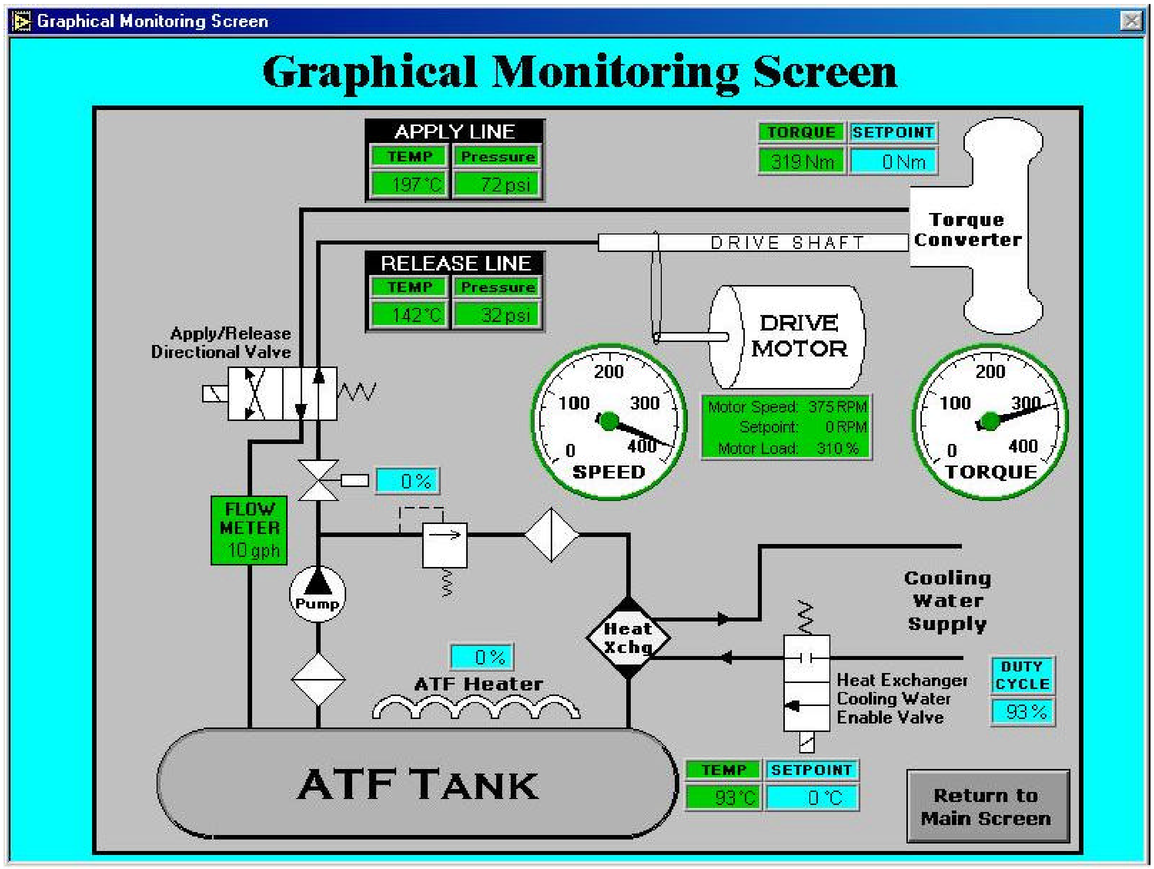

Figure 2. Mechanically familiar operators see symbols that change color based on state.

The changing symbols and icons on the graphical monitoring screen were designed to help easily determine the state of the torque converter equipment. The "edit control" feature of LabVIEW proved to be an important tool in the design and creation of these components on the front panel. Most items on the screen (including the Drive Motor and solenoid valves) were first drawn in AutoCAD as part of a drawing set. The pieces were then exported from AutoCAD 2000 into Microsoft Paint, using the inherent cut/copy/paste features of Windows. Fine-tuning of each object was completed, and two copies of each object were saved. One copy consists of the component in its resting (OFF) state, with white used as its main color. The other copy of the BMP shows the object in its active (ON) state, with its object color as green (or red, if fault object being displayed). A boolean indicator was edited, importing the resting BMP as the FALSE case and the active BMP as the TRUE case. By utilizing the "edit control" features of LabVIEW, it is possible to create ANY type of display item desired.

Conclusion
LabVIEW proved to be the best choice for implementing this system. The system required a PC, since process data was to be logged to a hard drive for future analysis. LabVIEW integrated ALL different aspects of the control system (machine control, data acquisition, data logging, and user interfacing) into one single programming platform. It can sometimes prove difficult to get different programming environments and hardware to communicate to one another. But the use of LabVIEW eliminates this costly programming, since everything is incorporated into ONE programming environment.