DESIGN OF A FUEL CELL TEST STAND

USING COMPACT FIELDPOINT FOR PID AND DISCRETE CONTROL

Client

A major fuel cell company.

Problem Scope

The control of a fuel cell test stand requires a combination of slow safety checks and faster PID control loops. Our customer also wanted precise and reliable control of the fuel cell test stand which required the capability to implement control algorithms more advanced than simple PID. As system designers, we wanted a modular architecture that would allow us to have multiple control loops, operating at different loop rates, running on one controller.

Our first major decision during the design of the new test stand was to select the best programmable controller. We decided to either use a PLC with some dedicated PID controllers or a PAC-like Compact FieldPoint (cFP) running LabVIEW Real-Time (LVRT). We knew that Compact FieldPoint had the capability to run more advanced control algorithms and to run multiple parallel loops, but since cFP is a new technology, we were concerned about the ability of Compact FieldPoint to control our process. Tight tolerances on process variables were needed, with some at less than 0.5% of full range, and the digital safety I/Os had to work reliably. Finally, the solution had to communicate with the test PC.

Viewpoint's Solution

Viewpoint Systems, Inc. worked closely with Micro Instrument, an automation vendor, building the test stands. Micro has extensive experience with PLCs and standalone controllers for control of repetitive motion, safeties, and other “environmental” parameters such as pressure and temperature. From this experience, it was clear that a PLC could deliver the discrete I/O control and that standard PID loop controls would hold the required process variable tolerances.

However, one of the major reasons for our customer to consider cFP was the ability offered by LVRT to run smarter PID loops. For example, by programming cFP, we could develop control loops that were aware of imminent system state changes and could act accordingly to eliminate overshoot or alternatively allow for faster approach to a setpoint. Since the customer did not know in advance what these “smart” controls would entail, it was clearly a benefit to have the full power of LabVIEW to develop such controls. Providing this functionality with a PLC would be cumbersome or impossible.

PID Control Loops

Our first task, then, was to prove that cFP with LVRT could deliver the required control tolerances. In this system, PID control was of two forms: PWM and continuous control.

First, we'll discuss the PWM control. Due to a non-optimal set of hardware to use for our initial testing, the PWM control was created with a standard digital output module. The PWM period was chosen to be 100 ms and implemented with a highspeed 1 ms loop that would create the PWM pulse by setting the output on or off according to the PWM pulse width. This provided a PWM pulse width granularity of 1%. Inside the 100 ms loop we read the process variable, performed a PID calculation using the LabVIEW PID toolkit to compute a new PWM pulse width, and passed that new width to the inner 1 ms loop, to build the PWM pulse via a digital output.

Even with this custom, software timed PWM, we achieved similar or better control than was available with our existing PLC controller. Ultimately, we used the cFP-PWM-520 module, rather than this inner 1 ms loop, enabling the LabVIEW Real-Time process to concentrate its CPU on the more advanced control tasks. For instance, in LabVIEW we implemented gain scheduling for the PID parameters and included some nonlinear control based on process error to change the minimum and maximum outputs allowed by the PID controller. These programming enhancements boosted our PWM control capabilities beyond the PLC in terms of reaction time and stability.

In the continuous control case, an analog input value was passed through a parallel PID calculation and then was passed to an analog output. This loop was designed to run at a 100 ms loop rate. Again, we achieved similar or better control than with our traditional controller.

Safety and Tooling Control

Our second task was to validate the capabilities of the cFP to perform discrete control. A typical PLC application will use discrete I/O to watch and manipulate the state of the machine. These discrete I/Os tend to be 24 VDC or 120 VAC, and we used Compact FieldPoint digital I/O modules to implement this functionality.

For our application, we needed to watch several safety sensors to monitor the test stand health. The loop time on these sensors did not need to be extremely fast, and we chose a 25 ms loop time. On some PLC applications, the reaction time to discrete inputs may need to be on the order of milliseconds. In our tests, we found that the cFP was able to achieve a 2 to 5 ms response time with careful application of LabVIEW coding techniques and appropriate digital I/O hardware. However, with the nearly 30 parallel loops we used (analog and discrete combined), a loop time of 25 ms was more realistic. In our case, had we needed the 2-5 ms response, we would have needed to include another cFP controller.

Communication with the PC

Our final task consisted of communicating the state of the test stand to the master PC. In addition, we needed to update new control setpoints to the various PID loops on the cFP controller.

On most PLCs, such communication between the PLC controller and a PC is done via a proprietary communication protocol, such as Modbus or Profibus, or via standards such as Ethernet. So, normally, the PC has a driver written to talk with the PLC or a standard layer such as OPC is used. In either case, to update a setpoint value held in a PLC memory location, the PLC tag associated with the memory is read from or written to via the driver or OPC interface. Then, any ladder logic program on the PLC can use an updated value by polling the PLC memory.

On the cFP controller, there are several options for moving data to and from the PC. First, since LabVIEW RT is running on the cFP controller, we could use a number of communication methods, including remote VI server, raw TCP/IP, and the FP tag I/O VIs used in conjunction with “memory” tags created with the ‘FP Publish.vi’ function. (This VI automatically creates tags that appear in FieldPoint Explorer.)

For reasons of both performance and ease of programming, we chose to use the “memory” tags. With this approach, we were able to read the process variable values from the cFP as well as write new setpoints and PID coefficients.

Results

The final solution used a single Compact FieldPoint controller to control the following independent parallel loops:

- 7 for PWM based temperatures control
- 2 for continuous pressures monitoring
- 4 for discrete solenoid and sensor monitoring and control
- 15 discrete safety loops

Each control loop is performed in a separate LabVIEW “while” loop. Control loops of similar type were programmed into a single VI so that we could tweak performance by placing the control VIs into separate threads.

The PWM control used inputs on the TC-120 module and outputs on the PWM-520 module. As stated above, we used the PID toolkit to perform the PID calculations. In addition, we incorporated some of the “smartness” the customer wanted by switching control methodology and PID coefficients depending on system state. Again, each temperature loop was in a separate LV “while” loop. The PWM temperature PID loops ran with a 500 ms cycle. These loops were all coded into a single VI that was set at ‘above normal’ priority.

The pressure control was done also within separate loops, programmed in fashion similar to the temperature loops. The inputs here came from the AI-110 module, and the AO-200 module drove the outputs. These loops ran at a 100 ms cycle and were all coded into a single VI that was set at ‘above normal’ priority.

The discrete control loops were coded into a third VI. These loops watched sensor input and reacted by closing or opening solenoids. We also switched solenoids based on thresholds on some analog levels. Multiple DI-300 and DO-400 modules were used. These loops ran with a 25 ms cycle and were coded into a single VI that was set at ‘time-critical’ priority.

Finally, we had a VI set at ‘normal’ priority used for communication with the PC.

In summary, the cFP with LVRT performed extremely well and surpassed the capabilities offered by standard PLCs for our requirements. The customer now has the future “smart” control capabilities that they desired with a strong foundation of performance equal or better to what we could have obtained with a PLC or standalone controller.