Originally appeared in HYDROCARBON PROCESSING® December 2006, pg 118. Posted with permission.

Y. ZAK FRIEDMAN, CONTRIBUTING EDITOR

Zak@petrocontrol.com

Audit your APC applications

I was one of the first control industry whistleblowers, pointing out that half of the advanced process control (APC) applications implemented over the years do not work. We APC engineers have the responsibility to straighten out this situation or we would soon be unemployed altogether. One of my clients, a site APC coordinator, asked me to give a presentation about how I would help audit APC applications, all of which were implemented by well-established APC vendors. That's not difficult, I replied. All we have to do is methodically go through the following steps:

1. Review the unit design and economics. Do not begin your audit before you have a full understanding of the unit. You are about to verify whether or not the APC improves the unit economics and will not be able to accomplish this job without fully understanding how this unit should be driven.

2. Review the DCS configuration for the unit. It may be embarrassing, but many DCS designs do not address the unit objectives and sometimes are plainly infeasible. APC cannot work effectively on top of such DCS configurations.

How can APC overcome such a situation? The best way is obviously to reconfigure the DCS. But what if red tape and politics preclude DCS reconfiguration? APC can cope with bad DCS designs by opening DCS loops, and creating new control (*CVs*) and manipulated variables (*MVs*). That is not ideal. You may have to accept it but the audit should not overlook the aberration.

3. Review the selection of *MVs*, disturbance variables (*DVs*) and *CVs*. Check that:

- Each *CV* has at least one *MV* that affects it.
- The *CVs* are not in conflict. An example of *CVs* in conflict is two temperature *CVs* or two inferential models that do not agree.
- There are no *MVs* "flapping in the breeze" with no reasonable logic to set their position.
- If there are DCS idiosyncrasies, they are adequately addressed.

4. Review the checking logic for the application. For every application there is a set of conditions that, if not satisfied, the application should be turned off.

5. Review inferential calculations. While favoring first-principles inferential models, I would accept any model that trends well against the lab. Be sure to look not at the inference signal that is biased daily but primarily at the unbiased signal. Biased signals tend to obscure the disagreements. Accurate agreement between unbiased inference versus lab is not very important but trending up and down together is. If the two do not trend together, there is no predictability and no biasing method would help.

Should you conclude that the inference model is not reliable, would you continue using this application? I would say that you must turn off the inference *CV*, stop any maximization and attempt only to stabilize the unit. Right there about two-thirds of the benefits are lost.

6. Review the steady-state application performance. The tests in items 1–5 are for verifying that an application has a reasonable design, and if you find that the design is not reasonable, save your time and investigate no further. If the design is acceptable, then item 6 begins looking into application performance. When the unit runs at constant load, we demand that:

- The application is stable. Sometimes instability occurs because of poor DCS tuning, but that is no excuse. DCS tuning is part of the APC and it should be stable.
- *CVs* and *MVs* are driven to their optimal positions.
- *MVs* should not generally be pegged against limits, or if they are, those limits must correspond to legitimate constraints, not an attempt by the operator to reduce the application freedom.
- *CV* targets should represent real unit constraints, or sometimes reflect upstream and downstream constraints.

It would be encouraging to see that the performance demands are satisfied. If they are, that can be directly translated to financial benefits. Process variability of the unit without APC is known or could be estimated, and the cost of variability is also generally known.

7. Review the dynamic application performance. We now come to the ultimate test of an APC application. Can this application work during the typical unit disturbances? There are large benefits associated with controlling the unit during transition periods, quickly stabilizing it at the new desired steady state, for example, dealing with feed switches, coker drum switches, mode switches, severe weather, etc.

That was what I said to the client and then realized that one cannot test an APC application without being an expert. We ended up conducting the audit together, not a simple job, about two days per major application. After the test, we had to face the difficulty of publishing the conclusions without alienating people. My client later obtained management approval to upgrade many of the applications.

A success story you might say, except I am discouraged to report that only 30% of the applications made it to item 6, and 10% passed item 7. Still, I thought this story would be of interest to the readers because of the need to face the music: audit and repair control application is very widespread. **HP**

Y. Zak Friedman is a principal consultant in advanced process control and online optimization with Petrocontrol. He specializes in the use of first-principles models for inferential process control and has developed a number of distillation and reactor models. Dr. Friedman's experience spans over 30 years in the hydrocarbon industry, working with Exxon Research and Engineering, KBC Advanced Technology and since 1992 with Petrocontrol. He holds a BS degree from the Israel Institute of Technology (Technion) and a PhD degree from Purdue University.