Originally appeared in HYDROCARBON PROCESSING® August 2009, pg 13. Posted with permission.

Y. ZAK FRIEDMAN, CONTRIBUTING EDITOR

Zak@petrocontrol.com

APC designs for minimum maintenance—Part 3

This is the third in a series of three editorials about designing advanced process control (APC) applications for an environment of little to no maintenance. Having lamented for years about APC applications failing due to lack of maintenance, we have to accept that a personnel shortage is here to stay. If we want our APC applications to survive in a minimum-maintenance environment we have to make them as robust as possible.

My June editorial proposed design rule 1: Avoid cluttering the multivariable predictive controller (MVPC) control matrix. Associate each control variable (*CV*) with one, or a maximum of two manipulated variables (*MV*s). The July editorial continued with rule 2: Avoid nearly redundant *CV*s, and rule 3: Where near redundancy is necessary, restructure the *CV*s to improve matrix conditioning. This editorial adds two rules that affect how robust an application is and how much maintenance it would need to remain useful.

Design rule 4. Keep the problem as linear as possible. MVPCs rely on linear models of the unit response to *MV* changes. However, the real world is nonlinear. Unit response changes with time, throughput and operational mode, and if those changes are not passed on to the MVPC the controller becomes confused and could begin to cycle between two or more unoptimal solutions.

Throughput-related nonlinearities are inherent in almost any application because MVPCs mix extensive variables, such as flows, with intensive variables, such as temperatures. If you halve the throughput then the response gain of a temperature *CV* to a flow *MV* would double. Commercial MVPCs permit gain scheduling on the fly, which offers a solution to the nonlinearity problem, although I have not personally seen massive use of this gain-multiplying feature.

Can we make the matrix linear by designing intensive manipulated variables, using for example yields instead of flows? While product quality *CV*s are typically intensive, constraint *CV*s, such as flooding or valve positions, are extensive. Restructuring the *MV*s would hence eliminate some nonlinearities but introduce others.

Should we make automatic use of gain multipliers? Gain multipliers adds complexity, but also robustness. Intuitively I think we should begin to design applications with automated gain scheduling. In a proper maintenance environment the APC engineer could manually scale the gains as needed and make the application run, whereas in the absence of maintenance, when the MVPC control models drift the application would likely be turned off.

Are process gains easily predictable? Throughput-related gains are predictable because they are inversely proportional to the throughput, but for other nonlinearities we need more elaborate rules. Still, one cannot design effective APC for a unit without a detailed understanding of the unit, and that includes changes between operational modes.

Design rule 5. Use high-quality inferential models. I come now to a topic near and dear to my heart. APC makes money by pushing the unit toward constraints, but such a push has value only if accompanied by precise product quality control. In the early years APC relied on simple inferential models (for speed of response) plus onstream analyzers (for accuracy) to achieve quality control. But analyzer reliability, never very strong to begin with, has further deteriorated over the years. The lack of maintenance environment that decimated APC has surely also reduced analyzer reliability. We are now at a point where we must design inferential models to work without being corrected by analyzers, especially for applications that are expected to survive in an environment of minimal maintenance.

I have written several articles against the practice of developing inference models via regression analyses.[1] A very good process engineer can perhaps specify model inputs correctly, conduct a series of test runs and identify a working model. He/she would need extensive lab support for the initial development as well as for continued testing and redevelopment upon process changes. While I have never been impressed by regression models, with continued lab support and periodic regression analyses maintenance of such inferences might succeed, but in an APC-neglect environment I cannot envision weak inferential models surviving very long.

What then is the type of inference model that can survive in a low-maintenance environment? I have made a career out of developing first-principles models,[2] and think that the more the model is based on process engineering principles the more reliable the inference is. Even the best inferential models need occasional recalibration, but that recalibration is fairly simple, involving a change of bias or multiplier. Actually, the most common cause of inferential model failure is an erroneous input measurement. An engineer must still be there to first identify the problem, and second to nag the instrument maintenance team, which is also short of people, to repair the offending measurement. **HP**

LITERATURE CITED

1 Friedman, Y. Z., "Choosing inferential modeling tools," HPIn Control, *Hydrocarbon Processing*, January 2006.

2 Friedman, Y. Z., "Where do you take those inferential models from?," HPIn Control, *Hydrocarbon Processing*, November 2008.

The author 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.