

May/June 2012

Your Letters

Process simulation—another benefit

Kudos to Platt Beltz for his excellent article about the many benefits of using process simulation for operator training (January/February 2012 *InTech*). There is, however, another major benefit of process simulation that often gets overlooked. Process simulation can also be a very useful tool for control system software development, testing, and verification.

Control system programmers typically write better software when they can test their code as they go. Code that is tested throughout the development process typically has less bugs, is easier to manage, and has a better chance of achieving what the process designer intended. When used as part of the overall software development process, process simulation can be an effective development tool and value-added feature.

Complex control algorithms can be verified and debugged before they hit the plant floor. HMI screens with their many dynamic screen elements, which can be notoriously hard to test, can often be easily “test-driven” with the aid of an appropriately written process simulator. Control system check-outs and software factory acceptance tests become much easier when the control code has a process simulator to work with.

Is doing simulation to aid software development hard to do? The answer is no. With today’s PLCs/RTUs, simulation can usually be implemented in code by using I/O redirection and simulation subroutines. Most modern DCS systems typically have some degree of simulation capability and/or connectivity for third-party simulation tools. Open communications protocols like OPC also allow control code and process simulators to be connected in various ways. The key with using process simulation as a development tool is to develop only the level of sophistication needed and no more. When used wisely, the benefits of using process simulation tools during the software development cycle can greatly outweigh the costs.

Graham Nasby

Starting at the process

Regarding Dick Caro’s September/October “The Final Say”: if good control is important, good maintenance is more important. In my experience, a “noisy” control loop is more likely due to issues with the actuator or measurement than the control loop tuning. You should never go into a situation assuming the issue is in the controller tuning and ignore the rest of the loop. Unfortunately, engineers are often considered “bought and paid for,” while maintenance requires parts and labor that is a “cost” to the plant; so if a process is performing badly, the engineer is told to fiddle with the gains. Poorly implemented instrumentation or an incorrectly-sized, worn-out, or sticky valve will lead to overshoot, hunting, and other problems that cannot be “tuned-out” unless you want to abandon good control of the process. When the actuator’s performance is degrading, dumbing-down the controller to smooth performance will just make overall loop performance worse. Before you improve a loop, you have to start at the process and work your way back towards the controller. Then you can worry about adjusting tuning parameters. You may find your problems have “magically” gone away (by restoring the rest of the loop to the state it was when the original tuning was correctly set).

Jeffrey

Grid synchronization issues

If the “Grid” was truly “Smart” (September/October “Talk to me”), it would not render local power production from wind and solar useless in the event of a grid failure. “Grid tied” local power systems only pretend to decentralize power. If we want autonomous power for our water agencies, homes, businesses, and schools, “Grid tied” has to be changed to “Grid optional,” a hybrid system of standalone or shared power. Let’s work on synchronization issues and truly make a smart grid that does not put people in the dark even when they have solar panels on their roofs.

Mark