Know your process better to control it better

Hans Eder, President, ACT, Brussels, Belgium

This second in a series of articles by Hans Heinz Eder of ACT will help you better understand economic-driven, proactive Advanced Process Control strategies. In it, Mr. Eder explains that, when properly applied, APC is so much more than simply the application of any one technology.

It is a widely-accepted fact that sound knowledge of the various controllers and technologies used in a system is a prerequisite to achieving good control performance. There is also little debate over the contention that the control system and its features have to be well known in order to develop and implement new controllers and control schemes. Yet many control professionals have relatively little knowledge about the processes they have to control, and sometimes even try to tune existing controllers or develop new control strategies without sufficient process knowledge.

However, sound process knowledge is one of, if not the most, important factors in achieving effective control. The better we know the behavior and specifics of the process, the better able are we to choose the right control scheme and the right controller type, and to find the best suited tuning for the given situation and performance requirements.

Controlling a process requires knowledge of various disciplines

Let us start with a general statement: To achieve good control performance we must have sound knowledge regarding:

1. The process,
2. Process control, and
3. The control system.

Item 1, knowledge of the process, is quite likely the most important of the above three. Included in this process knowledge should be the type of process behavior we have to deal with as well as the static and dynamic process parameters involved. And remember: when I speak about knowledge, I mean true, quantitative knowledge.

Item 2, knowledge of process control, is the most obvious: Without a thorough understanding of the fundamental behavior of controllers, and the various approaches used to configure and tune them, we can never (or least not in an acceptable amount of time) achieve reasonable control performance.

Item 3, the control system, is fairly easy to understand. Without an in-depth knowledge of the distributed control system (DCS) and its features, we are in no position to even attempt to implement the controller or control scheme. To accomplish these tasks, we must know not only what controllers are available, but also what their specific design features are in our control system (since these features can vary quite substantially from vendor to vendor). At the same time, we need to be aware of the other supporting features, modules, or algorithms we have at our disposal, and know how to establish the best communication between the user and the controller or a complete application.

The only item above that is not self-evident is number 1, process knowledge. Therefore, we need to examine this in more detail and prove why it is perhaps the most important factor of all. To do this, we need to answer the following questions:

1. Why is it so important? What benefits are derived from knowing, for example, that we are dealing with a process of third order with some ten minutes of deadtime?
2. What exactly do we need to know? Is it sufficient to know the process type in general terms (e.g., self-regulating or integrating)? And why do we need, as we mentioned earlier, quantitative information?

Let's now answer some of these questions.

Process type determines choice of controller

When we develop a new control scheme, we need to select the controller type that is most suitable for the process to be controlled. This suggests that we need to know, at the very least, the process type—for example, whether or not it is self-regulating. Furthermore, we should have an idea about the order of the

> *"The better we know the behavior and specifics of the process, the better able are we to choose the right control scheme and the right controller type, and to find the best suited tuning for the given situation and performance requirements."*

process and its behavior in qualitative terms (fast or slow, with no, short, or long deadtime, and so on).

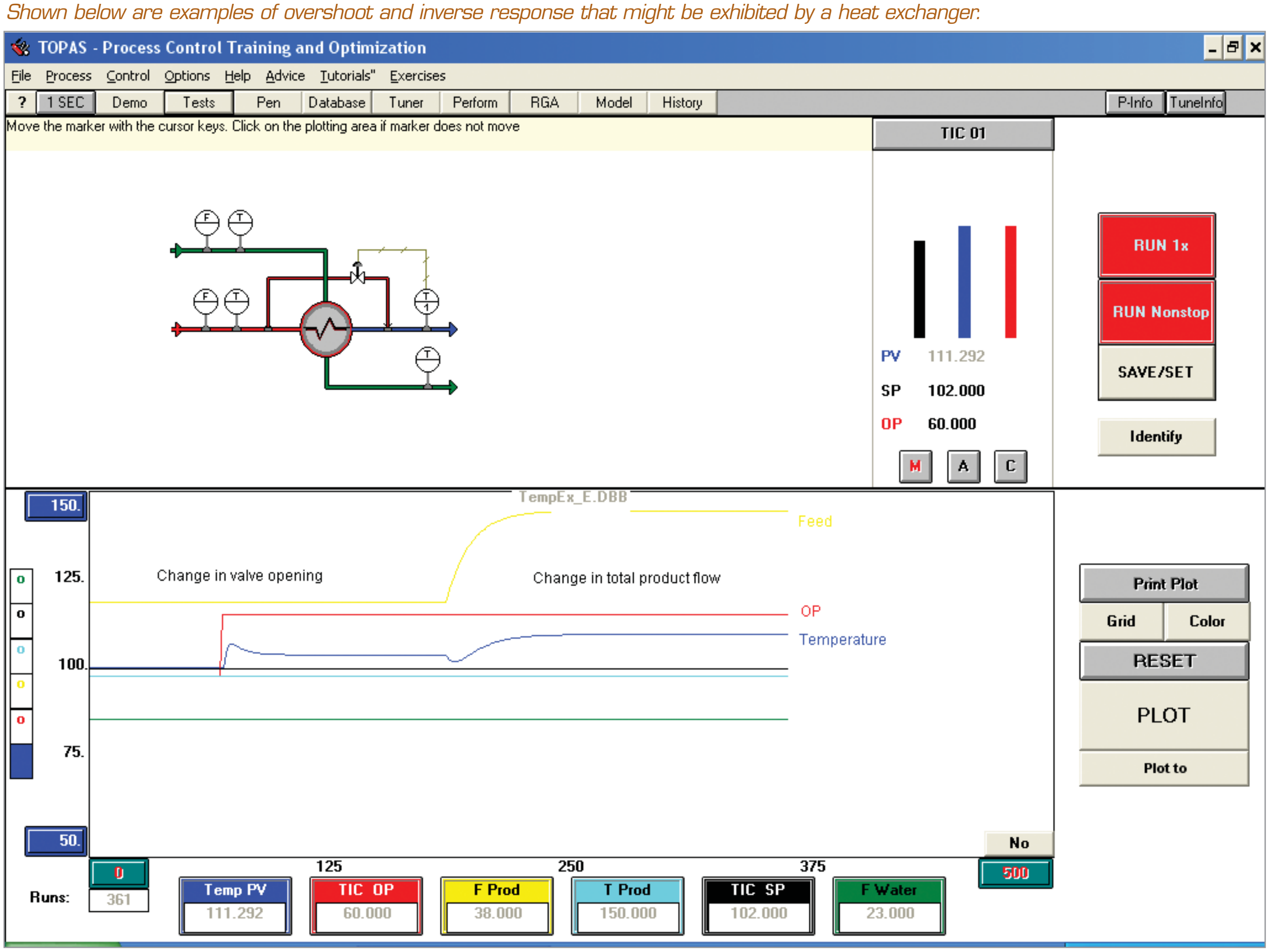
With a self-regulating process, when a change is made in the input to the process, the process output will come to a new equilibrium—that is, to a new steady-state value after a period of time. In contrast, with a non-self-regulating process, a change in input will result in a process output that keeps on changing until a limitation is reached. In the special case where this ongoing change in the output is linear, we are dealing with an integrating process.

So, what does process type have to do with the selection of the right controller? In answering this, in most cases we will use a standard PID-controller as a frame of reference, or parts of it such as P or I or PI. If we take the I-controller alone and analyze its behavior with different process types in a closed-loop, we see that it is certainly possible to control a fast, self-regulating process such as a flow loop with an I-controller alone. However, we also see that it gets much more difficult to tune the controller as soon as some deadtime is present. It is the ramping of the I-controller's output during the entire deadtime that gives us rapidly increasing problems with increasing deadtime. Nonetheless, for easy fast processes an I-controller could be considered a good or at least reasonable choice.

In case of an integrating process, the situation is completely different. An I-controller working in closed loop with an integrating process delivers a perfect oscillator. No matter how hard we try to tune it, it will always swing and our tuning will just affect the amplitude and period of the oscillation. Consequently, this controller is unsuitable for use in an integrating process—a fact that is too often overlooked and the reason for permanent swings in a system.

For a loop consisting of a self-regulating process and a P-controller, the PV will never reach the new setpoint upon a setpoint change. Therefore, the P-controller should never be used by itself for this type of process. On the other hand, a loop consisting of a pure integrating process and a P-controller exhibits the behavior of a first order system with no permanent deviation from the setpoint—a perfect behavior from a control point of view.

Shown below are examples of overshoot and inverse response that might be exhibited by a heat exchanger.



As long as the process is exhibiting a smooth, consistent transient behavior, our selection of a controller shouldn't be too difficult; a PI controller, for example, will likely be able to do the job. Unfortunately, there are quite a few processes that exhibit open loop behavior with overshoot or, even worse, with inverse response. These types of process conditions might be found, for example, in a heat exchanger that has to cool a hot gas or liquid—where control is achieved by changing the flow in the bypass around the exchanger by manipulating the bypass valve.

When the temperature controller is in manual mode and the bypass valve is opened more, the temperature will first rise swiftly, then drop somewhat since the reduced amount of material traveling through the exchanger will be cooled more. However, this second effect takes a bit of time and both effects combined cause an open loop overshoot (see Fig.).

If, on the other hand, the total product flow is increased—with the controller still in manual mode—in the first moment an increased amount of cooled material will be pushed out of the exchanger, causing a slight drop in the temperature; the temperature will then rise since the exchanger cannot cool down the increased material to the same temperature as before. In this case, we see an inverse response (see Fig.).

Any reactive controller, such as a PID or fuzzy logic controller, will simply follow the behavior of the measurement or its deviation from the setpoint, respectively; thus, during the time when the process moves in the "wrong" direction, the controller will react in the wrong way. Therefore, for a case such as this where there is a strong overshoot or very large dip during the inverse response, we would have to consider a controller that "knows" this transient behavior and, consequently, won't get confused by the process reaction. Clearly, the solution here is a model-based controller.

Process type also has an impact on controller tuning

From what we have said thus far, it is quite obvious that the tuning of, say, a PI controller is quite different for loops with self-regulating processes than those with integrating processes. In the latter, the integral action needed to bring the PV back to the setpoint in case of disturbances must be very weak to avoid oscillations. Under these circumstances, the main work will be done by the P-controller. On the other hand, for fast self-regulating processes, quite strong I-action can be applied.

From our heat exchanger example (see Fig.) we can also easily deduce that the tuning of a controller that has to deal with overshoot or inverse response will be quite different from that of a controller for a first or second order system. Regarding the order of the process, we also know that higher order systems will tolerate a stronger D-action than a true first order system because the initial response of the higher order system is much smoother.

We also know that interactions among the variables can threaten the stability of a loop. Therefore, we also need to have an idea as to the strength of these interactions because the stronger they are, the more we need to detune the individual controllers in order to maintain stability.

Quantitative process knowledge is key to proper controller selection and fast, effective tuning

General information about the process is certainly needed, but is not sufficient to achieve optimum control. The best way to understand this is by looking at a well-known, important indicator, the so-called controllability ratio. This is the ratio of the deadtime to the time constant. It's called the controllability ratio because it is a good indication of how easy or how difficult the process will be to control: The higher the value of this ratio, the more difficulties we can expect. In most cases, high values stem from long deadtime, which is a widely-recognized barrier to reaching sound performance. However, the controllability ratio can also assume large values when the deadtime is not all that long, but rather, the time constant is very small. The difficulty of such a situation is often not so obvious.

The controllability ratio also provides help in deciding on the controller type to use: For a value of a up to 2, the PID-controller is generally well suited. For values between 2 and 3, we know that we still can use the PID, but that tuning will get more and more cumbersome and time consuming. In fact, in this range, it might be faster to set up a complete model-based controller than to tune a standard PID. Finally, for figures higher than 3 we always need to resort to other controller types, such as a model-based predictive controller, expert system. or the like.

"It's been said and written many times that a great advantage of the PID controller is that it can be used and tuned just by trial and error—without sound process knowledge. In fact, I see this as a major disadvantage."

Just remember: To be able to calculate the controllability ratio and make the right decision as to the best controller type, we need to know the process parameters in quantitative terms. We will also need this quantitative process knowledge to choose the right sampling and/or control interval in digital systems—decisions which, once again, are mainly dependent on the process dynamics.

Yet another reason why quantitative process knowledge is so critical is that, with the process parameters known, we can use one of the many methods to calculate the PID tuning parameters. We can do this for the current situation, and also estimate how the tuning needs to be adapted when we foresee changes in the process, such as changes in the throughput, operating point, and so on.

Process behavior plays vital role in control

By analyzing the steps involved in choosing the right controller and setting it up properly, it becomes quite clear that process behavior plays an important if not vital role in achieving effective control. If we choose a controller type that is not suited for the process in question, no tuning effort whatsoever will give us the required performance.

It's been said and written many times that a great advantage of the PID controller is that it can be used and tuned just by trial and error—without sound process knowledge. In fact, I see this as a major disadvantage! Likewise, I see one of the key advantages of model-based predictive control is that we are forced to obtain all that important information—both qualitatively and truly quantitatively. And as those discover that follow this course of action and acquire the needed process knowledge, they will achieve results that invariably justify the extra effort.

About the author

Hans H. Eder is the founder and president of ACT, an Austria-based company delivering technical and management consulting, training courses, seminars and other special services, plus software products for operations optimization and automation, mostly to large international companies. A former advanced control engineer, APC manager and CIM Advisor with EXXON/ESSO, and CIM advisor with Exxon, he is a recognized expert in the economics of control. Mr. Eder has more than 25 years of experience in model-based predictive control.

Some of his many accomplishments include the design of an award-winning Windows tool called TOPAS for training and optimization, the development of a model-based predictive controller called AMC that fits into every DCS, and the development of a model-based batch reactor control scheme (feedback & feed forward) in a single day.

Mr. Eder received his masters degree in mechanical engineering from T.U. Vienna. He can be contacted at: ACT, Brussels office, Madeliefjeslaan 13, B 3080 Tervuren, Belgium (++32)-2-767-0895; hans.eder@act-control.com; http://www.act-control.com.