Adaptive Tuning - Expectations and Limitations

by John Gerry, ExperTune Inc. www.expertune.com

Adaptive tuning promises to tune loops without disturbing the process. Is this a technology you should consider using in your plant? This paper explores some of the pros and cons of adaptive tuning.

Types of Adapters

Adapters that do not disturb the process can be categorized in three different types. These types are:

1. Auto Regressive Moving Average (ARMA) (self-adapter)
2. Pattern recognition (self-adapter)
3. Pre-programmed adapter

The first 2 are considered self-adapting because they adjust tuning parameters when they detect the process has changed based on a changing process model. The pre-programmed adapter uses pre-set values. It instantly changes tuning based on the process variable or controller output POSITION only.

The self-adapters crunch on a relatively large amount of process data to make a decision. The self-adapters will update only after they have collected enough data to warrant a change. The pre-programmed adapter could change every sample interval based on a changed PV or CO.

Self-adapters are implemented with several hundred to several thousand lines of computer code. The common implementation of a pre-programmed adapter is a characterizer available in almost every control system.

Adapters using ARMA models, attempt to update a time domain model of the process. When the ARMA model is refreshed it is then used to calculate and update PID tuning values.

Pattern recognition adapters look at patterns of the process variable and attempt to update the PID tuning based on the pattern. With pattern recognition the process model is implied.

Pre-programmed adapters adjust tuning parameters based on the value of the process variable, controller output or some other process state. Pre-programmed adapters do not attempt to calculate new tuning parameters based on a new, calculated process model, pattern, or process dynamics. In pre-programmed adapters, the tuning adjustments are based on static pre-set values.

The Challenge of Adaptive Tuners

The promise and biggest challenge for adaptive algorithms is whether controller tuning can be gleaned without disturbing the process. If the process is not disturbed, this means the data used by the adaptive algorithm will contain only load upsets and noise.

Mathematical Analysis of Self-Adapters

Lets first look at the fundamental engineering required to find tuning parameters on a continual basis. First it requires knowledge about the process. We have to know what the process is for the tuning to be meaningful for that process. The process is derived from plant data. Methods of using patterns also infer a process model and are also restricted by the same fundamental engineering basics.

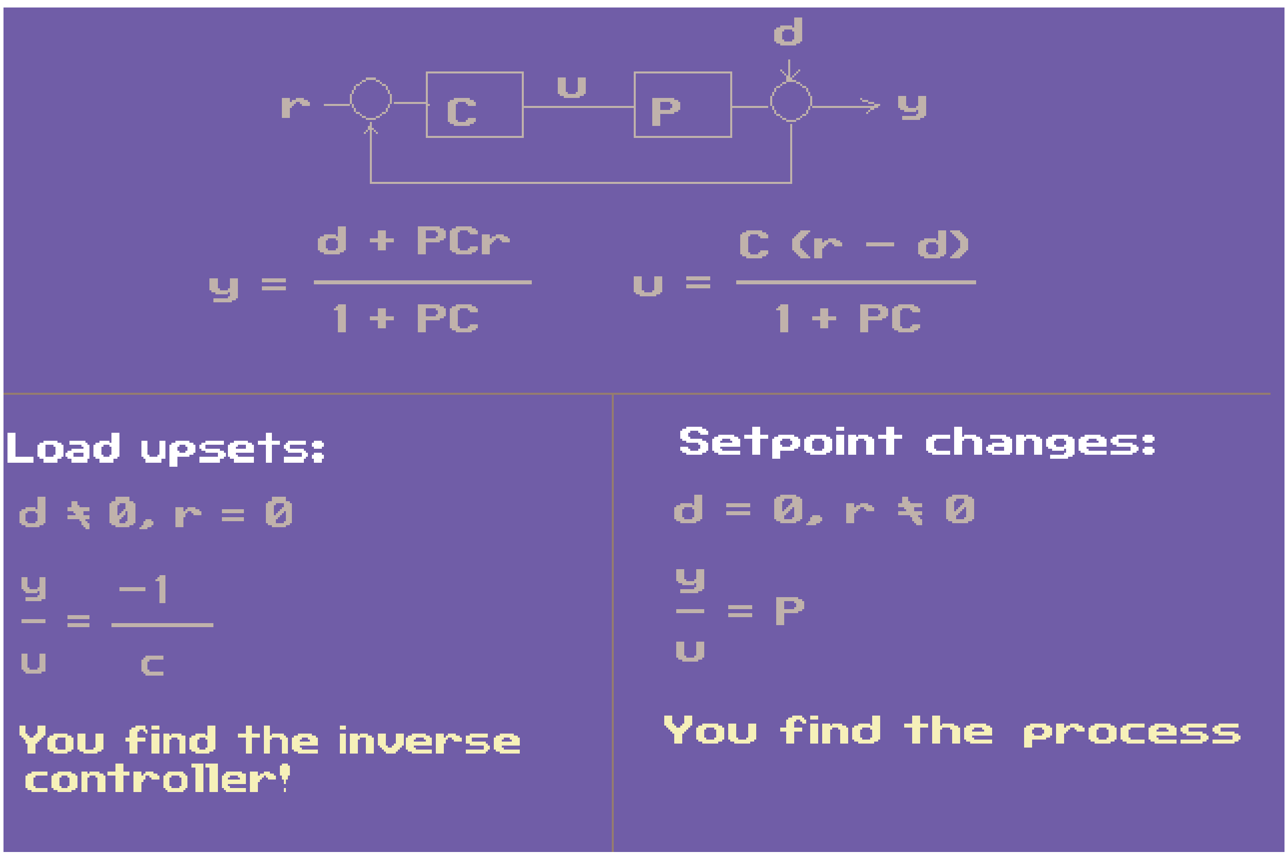

Figure 1: Algebraic statement of process identification.

In Figure 1 the problem is stated algebraically. Given excitation of the process by either set point changes or load upsets, we can solve for the process model,

P = y/u

based on process data collected. Substituting in both y and u and using either setpoint changes (d=0) or load changes (r=0) gives the resultant process model found by using either excitation.

Solving for an equation of the process based on load upset stimulus, the solution is the negative inverse of the controller, as shown on the lower left. Using load upset data drives a self-adapter to tune for the negative inverse of the controller. This is completely useless for arriving at meaningful PID tuning parameters and is at the center of the problem of any of the self-adapters.

If you do similar algebra using set point data, the solution is the process, as shown in the lower right. So, setpoint changes provide useful data for identifying process models, whereas normal operating data, where the setpoint does not change, provides nothing of value.

PRBS Signals

There are some other alternatives to getting good data continuously to model the process. All of them involve bumping the process continually. For example, we could continually excite (change) the setpoint or the output using a pseudo random binary signal or PRBS. While the PRBS sounds real fancy, it is simply changing the setpoint up and down a set amount with a set pattern. If the PRBS changes are large enough they can provide good data for process model identification. Unfortunately "large enough" is roughly making the process variable move 4 times larger than normal process noise. This amount of continuous setpoint change will probably deteriorate performance, quality, and energy usage etc. enough that the potential improvement from continually having the optimal tuning is gone.

Practical Considerations

If there were no load upsets, a PID controller would not be needed. Unknown disturbances that occur require that we have good feedback control to hold the process variable to the setpoint. Yet, because of load upsets and unknown disturbances, as we have seen above, the self-adaptive tuning engine will make unpredictable changes to the tuning parameters. For example, in the presence of an oscillatory upset, the self-adapter may be "fooled" into making the tuning parameters too sluggish. In the presence of very little excitation, the self-adapter may make the tuning parameters too aggressive.

Because the self-adapter's tuning changes are made without human intervention, the self-adapter has the potential to put in tuning that will give:

- Unstable operation: This would cause the controller output to slam into upper and lower limits. This would potentially cause much bad product, poor quality, causing a plant shutdown, or possibly causing safety problems.
- Very sluggish operation. In this case the controller would be very sluggish to respond to upsets. This could result in bad product quality, or a plant shutdown, or again possibly causing safety problems.

The "solution" to these problems is to put limits on the PID parameters that the self-adapter is allowed to download. When the self-adapter comes up with values that will drive the process unstable or make it very sluggish, we reject its tuning parameters. However, accurately knowing the limits or constraints on the tuning parameters requires a fairly intimate knowledge of the process. For users with this kind of knowledge, an Adaptive algorithm is probably not needed.

Valve Problems

Probably the most common cause of cycling in control loops is a mis-behaving valve. Valve hysteresis and or valve stiction will usually cause cycling.

Self-adaptive algorithms do nothing to diagnose or adjust tuning when stiction or hysteresis is present. The cycling caused by these problems will drive the self-adapter to keep backing off on tuning paramaters - the self-adapter must assume that the tuning parameters are responsible for the cycle.

More information:

How To Measure And Combat Valve Stiction On Line, ISA, Houston, September 2001
Stiction: The Hidden Menace, Control Magazine, November 2000
How Valve Performance Affects the Control Loop, Chemical Engineering, September 2000

Prior Process Knowledge Required (Pre-test)

Self-adaptive tuning algorithms require initial knowledge of the process to work. Usually this means knowing at least the process dead time and the process type. But often a complete initial process model is required. Self-Adapters therefore have a pre-tune feature that requires one or multiple process bump tests and a fair amount of algorithm setup.

Summary of Self-Adaptive Setup

To compensate for the drawbacks of poor adaptations, the self-adaptive algorithm requires a signficant engineering setup effort requiring:

- intimate process knowledge to set initial dead time and process type
- one or multiple process bumps
- intimate process knowledge to set limits on PID values

Non-linear processes

Where self-adaptive controllers may appear to really help is in processes with non-linearities. A non-linear process will have different response across the operating range. It will be sluggish at one end of the range and comparatively faster or even oscillatory at the other. The reason for the difference in response is a difference in process gain at different operating regions. The classic case of a tough non-linear process is pH control. But non-linearities can also be caused from other process equipment: valves, heat exchangers, pressure loops, etc.

The challenge in a non-linear process is that the control loop can move into a different range very quickly, causing a fast, sharp change in gain of the loop. When this happens the PID control loop is either too sluggish or it is too responsive (and perhaps oscillates).

The problem for self-adapters is they cannot change the PID tuning fast enough to be responsive to a non-linear process that has moved quickly into a different range. The self-adapter must wait for a period of time to examine the new response and attempt to come up with meaningful values, and all the time it is restricted by the problem of having to do this on load upset or noise data. During this time, your process quality is poor; you are wasting energy, etc.

More information:

How to Linearize Your Process: http://www.expertune.com/present.asp?name=Linear
pH Control Webinar: http://www.expertune.com/pastwebinars.html

Adaptive Controller Alternative

An alternative to the problems and lengthy setup of the self-adapters is to use Advanced PID Optimization software that will diagnose, recommend and solve control loop problems. The software should:

- diagnose outside oscillations
- identify valve problems
- solve non-linearities with a characterizer
- provide tuning parameters
- provide simulation and robustness plots

Since a control loop is comprised of more than just the tuning parameters the analysis software should include:

- Oscillation analysis to pinpoint the cause of plant wide oscillations
- Valve stiction testing
- Valve hysteresis testing
- Control loop linearity test

Once these checks are made, then, and only then does it make sense to tune the controller.

Since self-adapters require a pre-test, it makes sense to use powerful analysis software that should be able to use data from either a:

- CO step change in manual step
- CO pulse change in manual (fast test)
- setpoint change in automatic
- setpoint pulse made in automatic
- combination of manual and automatic change

The software should do this without prior knowledge of the process dead time, lag, or process type. Using the plant data alone, the software should provide a range of tuning options, safety margins, what-if simulation scenarios, robustness plots etc.

The ideal software is an Assessing Performance Monitor that continually scans and monitors the process for these types of changes. The software then automatically models and tunes the controllers alerting the user, at the user's discretion, of the new parameters. Wholesale tuning changes are possible.

Pre-Programmed Adaptive Control

The most sure, solid, and effective use of Adaptive control is to use pre-programmed Adaptive control. This, in effect, provides different tuning parameters depending on the state of the process. When the process moves to a different region, the tuning instantly changes to the set pre-programmed, and known values. There is no wondering what the pre-programmed adapter will do since the values are chosen based on the controlled variable or process variable.

The practical way to implement a pre-programmed adapter strategy is with a characterizer. Most control systems include characterizer blocks that let the user enter series of xy points that define the characteristic. The characterizer can be used either on the output or input of the PID controller depending on where the non-linearity is. The characterizers' non-linearity compensates for the varying process gain. It provides a smooth linearization of the control loop. Analysis software provides a characterizer builder that creates the code for this characterizer block. The block makes the overall process a linear one.

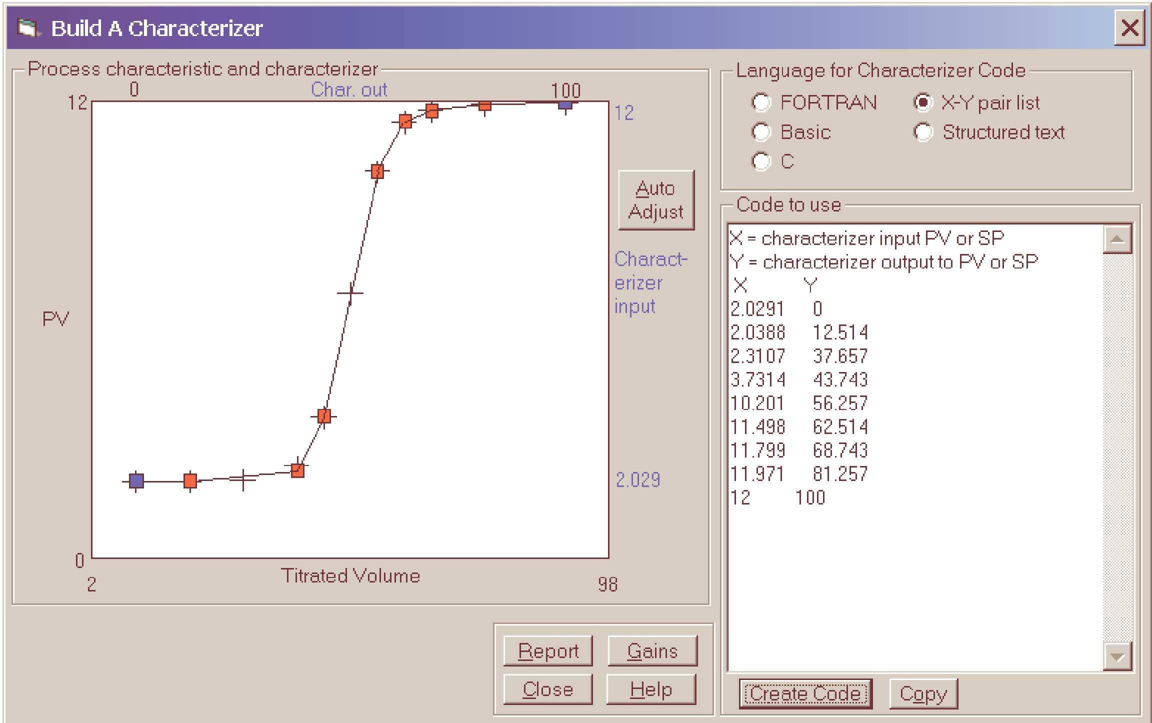

Figure 2: Characterizer code from analysis software

Conclusions

The high risk, long set-up times, and general ineffectiveness of self-adaptive algorithms make them a poor choice when compared to using Advanced PID Optimization software that will optimize the entire control loop and provide for a pre-programmed adaptor.

Of the differing adaptive controller techniques, pre-programmed adaptive algorithms provide the greatest possible benefit. This is because:

- The adaptation is known and pre-set based on the process
- There is no risk of downloading erroneous settings
- The adaptation is instantaneous
- The adaptation is simple to implement and available on most DCS systems and controllers.