Improve Process Efficiency Through Regulatory Control Stabilization

How to correctly tune a PID controller to bring the setpoint closer to the constraint with reduced variability.

By Merle Likins, Ph.D., P.E., Solutions Consultant, Yokogawa Corporation of America

Process industry plants must optimize regulatory and advanced control to maximize profitability, while at the same time maintaining safe operation. Regulatory control stabilization is key to achieving these goals, and stabilization can often be improved through closer evaluation of a plant’s regulatory control loops.

Most of these loops are operated by a Proportional-Integral-Derivative (PID) controller, and by better understanding how to tune these loops, plant personnel can improve quality and efficiency while ensuring plant safety. Moreover, achieving regulatory control stabilization forms the foundation for advanced process control (APC) implementation, which can be used to further optimize operations.

In addition to maintaining safe operation, a stable regulatory control system can increase profitability by reducing emissions and energy consumption and increasing the lifespan of equipment. By automating troublesome control loops, the need for manual operations is reduced, saving labor hours and increasing consistency of products.

This article offers an introduction on how to achieve a stable, well-tuned regulatory control system, with emphasis on improved PID control. Information is also provided with respect to PID software tuning packages that can be used to simplify and improve PID control.

Control Loop Basics
In a typical control loop, there is a parameter that needs to be controlled, such as temperature or pressure. This parameter is called the process variable (PV). A sensor usually measures the PV, and this measured variable is provided as feedback to the controller in a closed loop system.

The desired value for the PV, such as 40 degrees F in the case of a temperature control system, is called the set point variable (SV). If the PV is only 30 degrees F, for example, the controller will react to adjust its output to increase the temperature.

A constraint is the limit at which the process can be performed safely and efficiency. For example, the heat at 42 degrees F will destroy the compounds in a liquid during a certain process; therefore, the process must remain below this constraint. However, a process operating at 36 degrees F is so far below the constraint that quality is lost and/or the process becomes inefficient. Thus, the goal is to control the PV and keep it as close to the SV as possible with minimal variability. When variability is minimized, the SV can be moved closer to the constraint, improving operations.

PID Controllers
The PID controller is the most commonly used controller type in process plants, with more than 95 percent of the control loops in a typical plant under PID control. The familiar PID controller can be a great tool for improving quality, energy efficiency and production.

PID controllers offer good performance in a variety of operating conditions and they can be operated in a simple, direct manner. They can be standalone or embedded, and they can be used for one or many control loops. They can also be combined with sequential logic and advanced control functions to form complex automation systems.

The PID algorithm consists of three basic elements: proportional, integral and derivative. The setting for each of these three elements is varied in order to achieve an optimal response, that is to maintain the PV as close to the SV as possible with minimal variability. Each element is linked to a certain task that has a specific effect on the control loop.

A PID controller gets the measured PV data from a sensor, and then determines the difference (error) between the measured PV and the desired SV. It then computes the desired controller output or manipulated variable (MV) based on the error. The MV in turn acts on the final control element (for example, a steam valve) which in turn affects the process to change the PV (Figure 1).

![Diagram of PID controller](image)

Figure 1: **In a closed loop system, processes can be optimized by tuning the controller to reduce the error between the PV and the SV.**

There are many techniques for calculating the tuning constants for PID controllers. Most are based on developing a mathematical model for the combined final control element, process and measurement.

The model for describing the controller is simply the PID Algorithm. If a model can be determined for the process (final control element, process and measurement), then a mathematical model for the entire loop is in place. At that point, tuning the controller becomes a matter of matching the controller tuning constants (P, I and D) to the parameters of the process model, and optimizing the three PID constants to produce a desired response.

In the real world, the controller output is usually not the only signal that affects the system. There may be elements within the environment that cause deviations, such as ambient noise. These elements are called a disturbance. While disturbances must be factored in when controlling the process, they are also intentionally introduced on a temporary basis as an initial step to tune the controller.
Tuning a PID Controller

The process of tuning a PID controller involves five steps:

1. Introducing a disturbance into the control loop
2. Fitting the resulting response to a mathematical model
3. Using tuning correlations to calculate controller parameters
4. Implementing the new P, I and D parameters
5. Documenting the results.

The first step in tuning the PID controller involves inserting a disturbance that creates a new controller output (CO) into the loop. This is called a bump test. The disturbance introduced into the loop must be large enough to force a clear PV response, and the response must be large enough to distinguish it from any noise in the system. Unmeasured disturbances can corrupt the PV data, so larger CO bumps are better. In the example below (Figure 2), a CO bump causes the PV to move significantly.

Figure 2: Data from the bump test should show a dynamic cause and effect from the change in the controller output (CO).

Self-Regulating Process vs. Integrating Process

Any number of model types are possible for describing a process, but before choosing a model, one must first determine if the process is self-regulating and integrating (non-self-regulating). This determination is essential in tuning a controller because the different processes require different mathematical models.

After a change in the CO is introduced, a self-regulating process will eventually find a steady state as long as the CO and disturbance variable remain steady, as the process has an internal method of regulating the PV that balances the change in the CO. For example in a flow loop, if we open the valve an additional 10%, the flow will increase to a new value and will stay at that value until the valve is moved again. Most processes are self-regulating.

However, in an integrating (non-self-regulating) process, the PV will keep changing unless the CO is returned to its original position. With an integrating process, the PV will continue to rise or fall in a linear fashion in relationship to the CO change. This means an integrating process can be difficult to control.
An example of an integrating process is changing the liquid level in a tank. If more flow is introduced into the tank, the level will keep rising; if more flow is let out, the level will continue decreasing.

Tuning control loops for self-regulating processes tends to be easier than tuning integrating processes. Negative consequences can happen very easily and quickly if an integrating loop is tuned incorrectly. The inability to achieve equilibrium after a change in the CO is introduced in an integrating process could lead to problems, for example a liquid overflowing or running dry which could cause equipment (pump) damage, environmental damage or significant safety issues.

Self-regulating processes require both proportional and integral (and occasionally derivative) control modes for good performance. Despite their nomenclature, integrating processes rely much less on the integral control mode. Different tuning rules must be applied because the process models are different.

Using Modeling
After the bump test is performed and the type of process involved is determined, the next step involves fitting a model to the data. Modeling is one of the best ways to understand process data. A first order plus dead time (FOPDT) dynamic process model is usually sufficient to describe the process response and allow calculation of good tuning constants.

In this model the PV is the measured process variable and the controller output (CO) signal is the manipulated variable. The FOPDT model is simple (low order and linear), thus it only approximates the behavior of the real process. It’s represented by the following equation:

\[ \tau_p \frac{dPV}{dt} + PV = K_p \cdot CO (t - \theta_p) \]

Lumped Parameter Process Model
The lumped parameter model is an FOPDT model. This model describes the process response with simple linear equations comprised of three parameters: Gain \( K_p \), Time Constant \( \tau_p \) and Dead Time \( \theta_p \).

For the lumped parameter model, two forms of models are required depending on the nature of the process control loop: the self-regulating process (flow, temperature, etc.) and the integrating process (level).

Since an integrating process is more difficult to control, a different model must be used to describe this type of process, described by this equation:

\[ \frac{dPV}{dt} = K_p \cdot CO (t - \theta_p) \]

Tuning Correlation
There are many different tuning correlation methods used to calculate the PID tuning constants.

Ziegler-Nichols and Cohen-Coon are the two most popular techniques for calculating tuning constants. These two techniques emphasize speed of response. Internal Model Control (IMC), also referred to as the “Lambda” rules, offers a robust alternative that balances speed of response with controller stability.
or robustness. IMC tuning can be used for linear and non-linear processes, and it produces a smoother FOPDT response than other techniques.

IMC tuning is based on the concept that ideal control is possible with an exact model of the process. However, a mismatch between the model and the actual process can occur because of external disturbances that affect the process, which can lead to faulty results. As a result, IMC was designed to have methods for compensating for disturbances and modeling error, including filters and compensators in the higher frequency domain where many errors occur in other models.

Like other tuning procedures, a step test must be performed with IMC to determine the process characteristics. After determining the lumped parameter process model, a desired closed loop time constant ($t_{CL\text{taucl}}$) for the control loop must be selected.

If the closed loop time constant is too large, a slow control loop will result. Therefore, a smaller $t_{CL}$ value will create a faster control loop. But if the closed loop time constant ($t_{CL}$) is set to be shorter than the FOPDT process time constant ($\tau_P$), the advantages of IMC tuning will disappear.

Generally, the value for $t_{CL}$ should be set between one and three times the value of $\tau_P$. In many cases, $t_{CL} = 3 \times \tau_P$ is optimal to achieve a very stable control loop. Therefore, after determining the FOPDT process model, the IMC technique has one single tunable parameter, the closed loop time constant. The controller speed is made more aggressive or more conservative by changing the closed loop time constant.

IMC has one drawback in that the controller’s integral time is set to equal the process time constant. A process with a very long time constant means the controller will also have a very long integral time; and long integral times lead to very slow recoveries from disturbances.

Properly tuning a PID controller isn’t a simple process, but it’s one of the best methods for improving productivity, quality and safety in a process. By achieving a stable regulatory control system through improved PID tuning, the SV can be safely moved closer to the constraint while reducing the variability of the PV, thus reducing inefficiencies in the process.

However, collecting data and performing all of the modeling mathematics can be difficult and time consuming. Fortunately, advanced software can simplify PID controller tuning and reduce the possibility of error. The sidebar describes how one such software package can be used to automatically generate optimal tuning parameters, and to document the results of the tuning session.

Whether PID loop tuning is performed manually or with assistance from loop tuning software, the resulting improvements in the performance of each control loop will lead to significant overall performance gains throughout the process plant.
SIDEBAR: csTUNER Loop Tuning Software

This sidebar shows how to use csTUNER, Yokogawa’s loop tuning software. Other loop tuning software packages are available on the market may provide similar functionality, but csTUNER has been designed to interface seamlessly with the Yokogawa Centum family of control systems and offers features not available with other packages.

The csTUNER software enables automatic data collection, direct downloading of new controller constants, and documentation/archival of the tuning sessions. csTUNER facilitates effective tuning through its intuitive graphic displays. Its patented Non-Steady State (NSS) Modeling enables end users to accurately model both transient and noisy process data, delivering significant economic benefits to manufacturers across the process industries.

Through analysis of the documented process data, csTUNER users can further improve the performance of business-critical production processes. Offering a natural product extension to the Yokogawa’s CENTUM family of distributed control systems, csTUNER software uses identification algorithms and tuning correlations while providing a direct interface to the CENTUM DCS.

csTUNER is started from any controller faceplate. The screen has five (5) tabs across the top of the display. These tabs correspond to the five steps required to tune the controller.

- Perform "Bump" test and collect data (Grab Loop Data)
- Fit model to data (Fit Model)
- Use tuning correlations to calculate PID constants (Tune)
- Implement/Test (Implement Tunings)
- Document results (Report)

When csTUNER starts, it automatically begins collecting set point (SV), process variable (PV) and controller output (MV) data. Figure 3 shows the first csTUNER display with trending of the flow rate response as the outcome of a change in the controller SV.
Figure 3: This csTUNER trend screen shows the flow rate response resulting from a change in the set point.

An important feature to note is that the response does not need to settle at a steady state value, as the identification algorithm will identify the model without waiting for a steady state. This feature greatly reduces the time required to tune a control loop.

Figure 4 shows the next step, modeling the response to the a first order plus deadtime model (FOPDT). In this example, two models are fitted: one based on the bump step up, and the second based on the step down. csTUNER collects the time and date information for the data and shows the model parameters, as well as some statistical indices to demonstrate the appropriateness of the fit. At the top of the display is the composite model (average) of the two model fits. Any number of models may be created.
Figure 4: This screen shot shows the modeling response to FOPDT.

Figure 5 highlights the third tab (Tuning) on the csTUNER display. This indicates the current tuning and current response to a SV change along with the predicted response showing a new set of tuning constants. It also depicts the stability plot. A slider allows changing the closed loop time constant, which speeds up or slows down the response, and the predicted response is shown in real time. The current and new tuning constants are also shown.
Figure 5: The current tuning and current response to a SV change with the predicted response for a new set of tuning constants.

When the control engineer determines the new tuning constants they are downloaded directly to the DCS controller block from the fourth tab (Figure 6). The new and current constants are displayed. The results of the new controller constants can be tested and displayed before concluding the tuning session. The display also enables the reloading of the initial tuning constants.
Figure 6: The new and current constants are displayed for testing before completing the tuning session.

Finally, on the last tab, csTUNER produces a report showing the data collected, bump tests, models produced and old and new tuning constants (Figure 7). This report is automatically archived to document changes to the control system. In addition, the entire tuning session is also archived to help evaluate new models, test different controller constants, and perform other actions to optimize loop control.
Figure 7: A report of the data collected, tests, models and new tuning constants is created.

The csTUNER package offers fully integrated PID optimization software, and it seamlessly integrates with the CENTUM DCS to enhance existing software investment. csTUNER applies the DCS standard tag structure to access critical loop information, including existing tuning coefficients, controller spans and associated algorithms; and it automatically logs data for more in depth analysis.