

So Many Tuning Rules, So Little Time

Gregory K McMillan

Introduction

Considerable time and effort has been devoted to developing and touting tuning rules. The developers and their followers of tuning rules are adamant that theirs are the best. The user can be left confused as to what tuning rule to use and how. Often not stated are the revisions and practices needed in the applications of the rules for maximum achievement. Industrial consultants who have extensive field experience in the application of these rules are the best bet along with the use of software to identify the open-loop dynamics. These industry-wide consultants using process identification and auto tuning or adaptive control software can provide a wide spectrum of solutions and the education for the user to develop the tuning skills needed. Outstanding industry consultants have been interviewed by me in my [Control Talk Columns](#). Examples in alphabetical order are James Beall, Mark Coughran, Sigifredo Nino, Michel Ruel and Jacques Smuts. I don't include myself because I am too mature (i.e., too old) to work in the field. In a way this frees me up to be more objective. I am not trying to sell my services.

Industry-wide consultants can extend tuning rules to deal with all types of situations and objectives. They can also address the root causes of poor performance by improving the design and installation of the control strategy and the field measurements and valves, and getting the most out of the PID controller by the intelligent use of PID features. Most notable is the use of output tracking to eliminate manual operator actions and external reset feedback to eliminate oscillations from discontinuous measurement and valve response and to enable directional move suppression. The control literature has not sufficiently documented the expertise of these consultants and tends to give a misleading view from a focus on a specific simplified process and performance metric. Consultants who have experience in several different industries will have a better perspective. For example, consultants who work primarily on gas unit operations with heat integration in hydrocarbon plants and inline liquid blending unit operations in pulp and paper plant may be focusing on minimizing the movement of the PID output. These processes suffer from interactions and short-term variability in the product since there is no attenuating effect from a large process time constant in the unit operation or in a downstream volume.

The control literature to date usually uses for test cases balanced self-regulating processes, also known as moderate self-regulating processes, where the primary time constant is about the same size as the dead time. Many of the tuning rules for pole-zero cancellation such as internal model control and lambda were based on achieving the best setpoint response for these processes. The tuning parameter is a closed-loop time constant that is the time after the dead time for a process variable to reach 63% of the setpoint change. Originally, the closed-loop time constant was thought of as a multiple of primary process time constant also known as the open-loop time constant.

Setpoint response for continuous processes is usually only a consideration during start-up. Setpoints are also changed during real-time optimization, but these changes tend to be so slow and gradual that setpoint response is not an issue. The greater problem is the

So Many Tuning Rules, So Little Time

Gregory K McMillan

ability to deal with disturbances. If there were no disturbances, there would not be much of a need for feedback control. Process engineers could home in on the best flows with the process flow diagram as the starting point.

A PID tuned for best disturbance rejection can achieve a best setpoint response by the use of a “two degrees of freedom” (2DOF) PID structure or a setpoint lead-lag. There are also techniques, such as a bang-bang type of logic in the online Control Global article “[Full Throttle Batch and Start-up Response](#)” that can minimize rise time and overshoot. The whole premise of tuning rules to achieve a best setpoint response largely goes away.

Industry-wide consultants have modified the pole-zero tuning methods to deal with a wide variety of disturbances and process dynamics. In industrial applications there are self-regulating processes with low and high time constant-to-dead-time ratios. There are also processes with zero internal process feedback, termed integrating processes, and processes with internal positive feedback, termed runaway processes. Furthermore, a lead in the same direction or the opposite direction (e.g., inverse response) can exist.

The performance metric predominantly appearing in the literature is the integrated absolute error (IAE) for a step disturbance. The studies typically dealing with self-regulating processes’ total judgments based on relative values of the IAE are dismissive of tuning rules and even PID control. Unfortunately, the tests are often made without the practical modifications found and practiced today. The prime example is the use of a closed-loop time constant as a multiple of the primary time constant and the continued use of self-regulating process tuning rules for large time-constant-to-dead-time ratios.

Lambda tuning consultants know that lambda should not be a multiple of the time constant, but instead set relative to the dead time, and that processes with a large time-constant-to-dead-time ratio are near-integrating and should use integrating process tuning rules. To achieve the IAE performance touted in the literature, lambda should be set less than the dead time despite the recognition that a lambda range from 3 to 5 dead times is appropriate for industrial applications to deal with the inevitable nonlinearities and non-idealities. To compete with studies of ideal linear systems, a lambda equal to about 0.6 times the dead time can approach a minimum IAE with a smooth response (no faltering and negligible oscillation). Such studies depend upon an unrealistic situation of perfect measurements and valves and accurate complete knowledge of fixed open-loop dynamics. The absolute minimum IAE response typically has some oscillation albeit well damped. The effect of sensor lags and a valve’s pre-stroke dead time and velocity-limited exponential response with backlash and stiction is not considered. Despite the idealism, knowledge of the low limit of lambda is useful for gamesmanship, avoidance of dismissal of the lambda tuning and understanding of the proximity to the best IAE.

Since the 1990s, it has been known in lambda tuning that one should switch from self-regulating to integrating process tuning rules for near-integrating processes when using pole-zero cancellation tuning rules. Most of the literature does not recognize this switch. For integrating processes, the tuning parameter becomes an arrest time, which is the time to start to reverse a process excursion after a load disturbance. This lambda is much more

So Many Tuning Rules, So Little Time

Gregory K McMillan

in tune with minimizing IAE for disturbances. The arrest time should be set equal to 0.6 times the dead time for minimizing IAE. This requirement is readily visualized since dead time places a limit on how soon an excursion can be reversed. Integrating process tuning rules and the concept of an arrest time is even more important in runaway processes to prevent a large deviation (e.g., high temperature excursion in a highly exothermic reactor) from causing an acceleration to the point of no return (e.g., activation of reactor relief system). Runway processes are characterized by a positive feedback time constant, but are treated as integrating processes because open-loop tests cannot be safely done to reveal the acceleration.

Integrating process tuning rules also recommend the use of derivative action with a rate time set equal to a secondary time constant to compensate for its extremely detrimental effect on near-integrating, true-integrating and runaway processes. Unfortunately other pole-zero cancellations methods do not switch to near-integrating tuning rules, and even when using integrating process tuning rules, the tuning parameter is the closed-loop time constant based on a setpoint response.

Documentation by Bill Bialkowski (developer of lambda tuning) of the near-integrating approach and relating lambda to dead time appeared in the 1999 5th Edition of *Process/Industrial Instruments and Controls Handbook* that I compiled for McGraw Hill. Unfortunately, I did not realize the significance of these aspects of lambda tuning until I took a closer look about three years ago at the problem with lag-dominant processes.

In my career at Monsanto and Solutia, the most critical loops were composition, pressure and temperature control of batch and liquid continuous unit operations such as distillation columns, crystallizers, evaporators, biological and chemical reactors, and neutralizers. These all had a near-integrating, true-integrating or runaway process response. I don't remember any dead-time-dominant loops because the temperature control of extruders had thermal lags associated with heat transfer zones and thermowells; the sheet thickness control had thermal lags from the use of heaters instead of actuators to change die bolt clearance; and the temperature control of plug flow gas reactors had thermowell lags. The disturbances were nearly always process inputs, and there wasn't much heat integration. There was little concern about moving the PID output too fast. In fact for near-integrating, true-integrating and runaway processes, you need to overdrive the PID output past the final resting value to reverse the direction of an excursion. Furthermore, driving the PID output to an output limit was important to reach a temperature setpoint sooner to reduce batch cycle time in a chemical reactor.

Tight pressure control at Monsanto and Solutia prevented the propagation of disturbances in headers and supply lines. The most disruptive disturbances were fast liquid flow changes from a level controller or an operator and a channeling of gas flow fluidized beds. Temperature disturbances tended to be slower except in some utility systems. Composition disturbances tended to be slow due to large liquid volumes. Minimizing peak error was important to prevent shutdowns, and minimizing integrated error was important to prevent the accumulation of off-spec product.

So Many Tuning Rules, So Little Time

Gregory K McMillan

For these and many other reasons, disturbance rejection was the goal, and minimizing the movement of the PID output was not a concern. Since then, I have learned that turning on external reset feedback and setting setpoint up and down rate limits on the manipulated valve or flow loop can provide the directional move suppression needed to slow down the change in the PID movement without sacrificing much peak error and IAE performance.

The relay method of auto tuning developed by Astrom was extensively used at Monsanto and Solutia. The identification of the maximum sustained ramp rate in the right direction was also used to quickly estimate the open-loop integrating process gain. A dead time block with the dead time equal to the total loop dead time was the key to a good signal-to-noise ratio in the computation of rate of change of the process variable.

To help alleviate the burden of negative statements and misunderstandings, I have done a series of tests to see how various common tuning rules would perform for unmeasured step load disturbances to lag-dominant, self-regulating processes, integrating processes, balanced self-regulating processes and dead-time-dominant processes. The results for the runaway process I chose are similar to those for integrating processes and are not included. The performance metrics are peak error and IAE. Peak error is inversely proportional to PID gain, and the IAE is the integrated error that is proportional to the ratio of reset time to PID gain for a process that is not oscillating. Thus, tuning rules that minimize the reset time and maximize the PID gain with a smooth non-oscillatory response are judged to be best here.

Test Results

Internal model control, Skogestad's internal model control plus (SIMC+), lambda and the short cut method (SCM) tuning rules were tested. The SCM rules were developed by me as Senior Fellow in Solutia Inc and modified slightly in recent years to include limits to prevent a rate time greater than $\frac{1}{4}$ the reset time (important for the ISA Standard Form) and a PID gain too small for dead-time-dominant processes. The lambda tuning rules are as practiced by consultants with the addition by me of limits similar to what I use for my SCM rules and with the addition of a rate time low limit of $\frac{1}{2}$ the dead time for near-integrating, true-integrating and runaway processes. I am sure consultants who use IMC and SIMC+ tuning have also made modifications based on field experience. Thus, the performance of these rules with modifications may be better than shown. For example, I would expect users of SIMC+ would also switch to integrating process rules for lag-dominant processes (near-integrating processes). The tuning rules used in these tests and other common tuning rules are extensively documented at the end of this paper. The ISA Standard Form was used with a PI on error and D on PV structure.

To help make the tests fairer, the gamma for the IMC and SIMC methods were set to the same value relative to the loop dead time as lambda, even though the literature tends to show gamma as a factor of time constant. Values of 1, $\frac{3}{4}$ and $\frac{1}{2}$ the dead time were chosen for gamma and lambda in the attempt to minimize the peak error and IAE. Note

So Many Tuning Rules, So Little Time

Gregory K McMillan

that in all cases we were also looking for a smooth continuous return to setpoint with a negligible crossing of the setpoint. This is not seen in the tables, but in the trend plots. For example a loop with a minimum IAE may falter in its return to setpoint because of a PID gain setting too high relative to the reset time, causing proportional action to be so much greater than integral action as to cause a premature reversal of the PID output.

If you don't want to wade through the tables and figures, the bottom line is that lambda tuning does about the same as my SCM tuning. IMC and SIMC+ tuning have problems for particular processes when you push to minimize peak error and IAE. The problems show up as high errors or, in the case of IMC, tuning a faltering (hesitation) in the return to setpoint or excessive oscillations for a gamma equal to or less than the dead time.

The dead time in the table consists of a pure process dead time plus an equivalent dead time from valve pre-stroke dead time and second-order velocity limited response, PID execution rate, transmitter update rate and sensor lag. The tuning rules all assumed a secondary time constant of 1 second was properly identified.

Studies that use a pure dead time will not give the exactly the same results. Any estimate of peak error or IAE should be considered to have an accuracy of 10% at best. Judgment of methods should not be based on small differences in the metrics.

The overall results show that SIMC+ tuning has excessive peak error and IAE, for near-integrating processes and, to a lesser extent, true-integrating processes. IMC tuning has an excessive IAE and faltering (hesitation) in the response for near-integrating and excessive oscillation for a true-integrating process. Lambda tuning and SCM tuning yielded the best peak error and IAE with a smooth response.

IMC tuning had the best IAE and peak error for a balanced self-regulating process, but developed a faltering in the response for a gamma reduced to $\frac{3}{4}$ the dead time and excessive oscillation for gamma set equal to $\frac{1}{2}$ the dead time. In contrast, SIMC+ tuning had the best IAE with no oscillation for this minimum gamma. Lambda tuning gave about as good a peak error and IAE as SCM tuning when lambda was set equal to $\frac{1}{2}$ the dead time.

For dead-time-dominant processes, IMC tuning had a faltering response for gamma equal to the dead time and became oscillatory for small gamma. Lambda, SCM and SIMC+ tuning gave about the same IAE with a relatively smooth response. The peak error for all of the tuning methods was the open-loop error of 20% as expected.

Despite the vested interest in SCM tuning, I advocate lambda instead of SCM tuning because of the simplification to a single tuning parameter to deal with other objectives, such as maximizing the absorption of variability for surge tank level, preventing violation of the cascade rule, minimizing resonance, dealing with a process lead, maximizing consistency in blending and minimizing interaction. For an excellent synopsis of this flexibility and capability see the online white paper at Control Global by Mark Coughran titled "[Lambda Tuning—the Universal Method for PID Controllers in Process Control.](#)"

So Many Tuning Rules, So Little Time

Gregory K McMillan

Table 1 – Lag Dominant Self-Regulating Process Results for 20% Load Upset

| Loop # & PID Tuning Method | Integrated Absolute Error (%-min) | Peak Error (%) | Primary Time Constant (sec) | Total Loop Dead Time (sec) | Gamma or Lambda (sec) | PID Gain | PID Reset Time (sec) | PID Rate Time (sec) |
|----------------------------|-----------------------------------|----------------|-----------------------------|----------------------------|-----------------------|----------|----------------------|---------------------|
| Loop 1 Lambda (Purple) | 1.5 | 2.3 | 100 | 10 | 10 | 7.5 | 30 | 5 |
| Loop 2 IMC (Black) | 4.7 | 2.5 | 100 | 10 | 10 | 7.0 | 105 | 4.8 |
| Loop 3 SCM (Red) | 1.3 | 2.3 | 100 | 10 | ----- | 8.0 | 23 | 4.5 |
| Loop 4 SIMC+ (Brown) | 5.1 | 3.2 | 100 | 10 | 10 | 5.2 | 80 | 1 |
| Loop 1 Lambda (Purple) | 1.3 | 2.2 | 100 | 10 | 7.5 | 8.2 | 25 | 5 |
| Loop 2 IMC (Black) | 4.0 FR! | 2.2 | 100 | 10 | 7.5 | 8.4 | 105 | 4.8 |
| Loop 3 SCM (Red) | 1.3 | 2.3 | 100 | 10 | ----- | 8.0 | 23 | 4.5 |
| Loop 4 SIMC+ (Brown) | 3.9 | 3.0 | 100 | 10 | 7.5 | 5.9 | 70 | 1 |
| Loop 1 Lambda (Purple) | 1.1 | 2.2 | 100 | 10 | 5.0 | 8.9 | 20 | 5 |
| Loop 2 IMC (Black) | 3.2 FR! | 2.1 | 100 | 10 | 5.0 | 10.5 | 105 | 4.8 |
| Loop 3 SCM (Red) | 1.3 | 2.3 | 100 | 10 | ----- | 8.0 | 23 | 4.5 |
| Loop 4 SIMC+ (Brown) | 2.9 | 2.8 | 100 | 10 | 5.0 | 6.9 | 60 | 1 |

FR! - Faltering Response

So Many Tuning Rules, So Little Time

Gregory K McMillan

Table 2 – True-Integrating Process Results for 20% Load Upset

| Loop # & PID Tuning Method | Integrated Absolute Error (%-min) | Peak Error (%) | Open-loop Integrating Process Gain (1/sec) | Total Loop Dead Time (sec) | Gamma or Lambda (sec) | PID Gain | PID Reset Time (sec) | PID Rate Time (sec) |
|----------------------------|-----------------------------------|----------------|--|----------------------------|-----------------------|----------|----------------------|---------------------|
| Loop 1 Lambda (Purple) | 1.7 | 2.5 | 0.01 | 10 | 10 | 7.5 | 30 | 5 |
| Loop 2 IMC (Black) | 2.4 OR! | 2.5 | 0.01 | 10 | 10 | 13.3 | 30 | 1 |
| Loop 3 SCM (Red) | 1.5 | 2.5 | 0.01 | 10 | ----- | 8.0 | 27 | 5.3 |
| Loop 4 SIMC+ (Brown) | 5.3 | 3.8 | 0.01 | 10 | 10 | 5.0 | 80 | 1 |
| Loop 1 Lambda (Purple) | 1.5 | 2.5 | 0.01 | 10 | 7.5 | 8.2 | 25 | 5 |
| Loop 2 IMC (Black) | 6.0 UR! | 2.4 | 0.01 | 10 | 7.5 | 16 | 25 | 1 |
| Loop 3 SCM (Red) | 1.5 | 2.5 | 0.01 | 10 | ----- | 8.0 | 27 | 5.3 |
| Loop 4 SIMC+ (Brown) | 4.1 | 3.5 | 0.01 | 10 | 7.5 | 5.7 | 70 | 1 |
| Loop 1 Lambda (Purple) | 1.3 | 2.4 | 0.01 | 10 | 5.0 | 8.9 | 20 | 5 |
| Loop 2 IMC (Black) | 7.4 UR! | 3.4 | 0.01 | 10 | 5.0 | 20 | 20 | 1 |
| Loop 3 SCM (Red) | 1.5 | 2.4 | 0.01 | 10 | ----- | 8.0 | 27 | 5.3 |
| Loop 4 SIMC+ (Brown) | 3.0 | 3.2 | 0.01 | 10 | 5.0 | 6.7 | 60 | 1 |

OR! - Oscillatory Response UR! - Unstable Response

So Many Tuning Rules, So Little Time

Gregory K McMillan

Table 3 – Balanced Self-Regulating Process Results for 20% Load Upset

| Loop # & PID Tuning Method | Integrated Absolute Error (%-min) | Peak Error (%) | Primary Time Constant (sec) | Total Loop Dead Time (sec) | Gamma or Lambda (sec) | PID Gain | PID Reset Time (sec) | PID Rate Time (sec) |
|----------------------------|-----------------------------------|----------------|-----------------------------|----------------------------|-----------------------|----------|----------------------|---------------------|
| Loop 1 Lambda (Purple) | 14.6 | 13.8 | 20 | 20 | 10 | 0.50 | 20 | 5 |
| Loop 2 IMC (Black) | 10.2 | 12.9 | 20 | 20 | 10 | 1.00 | 30 | 6.7 |
| Loop 3 SCM (Red) | 11.2 | 13.3 | 20 | 20 | ----- | 0.80 | 26.4 | 4.2 |
| Loop 4 SIMC+ (Brown) | 13.6 | 14.0 | 20 | 20 | 10 | 0.67 | 26.7 | 1 |
| Loop 1 Lambda (Purple) | 13.1 | 13.6 | 20 | 20 | 7.5 | 0.57 | 20 | 5 |
| Loop 2 IMC (Black) | 8.5 FR! | 12.8 | 20 | 20 | 7.5 | 1.20 | 30 | 6.7 |
| Loop 3 SCM (Red) | 11.2 | 13.3 | 20 | 20 | ----- | 0.80 | 26.4 | 4.2 |
| Loop 4 SIMC+ (Brown) | 11.9 | 13.9 | 20 | 20 | 7.5 | 0.76 | 26.7 | 1 |
| Loop 1 Lambda (Purple) | 11.9 | 13.4 | 20 | 20 | 5.0 | 0.67 | 20 | 5 |
| Loop 2 IMC (Black) | 7.4 OR! | 12.6 | 20 | 20 | 5.0 | 1.50 | 30 | 6.7 |
| Loop 3 SCM (Red) | 11.2 | 13.3 | 20 | 20 | ----- | 0.80 | 26.4 | 4.2 |
| Loop 4 SIMC+ (Brown) | 10.5 | 13.7 | 20 | v | 5.0 | 0.89 | 26.7 | 1 |

FR! - Faltering Response OR! – Oscillatory Response

So Many Tuning Rules, So Little Time

Gregory K McMillan

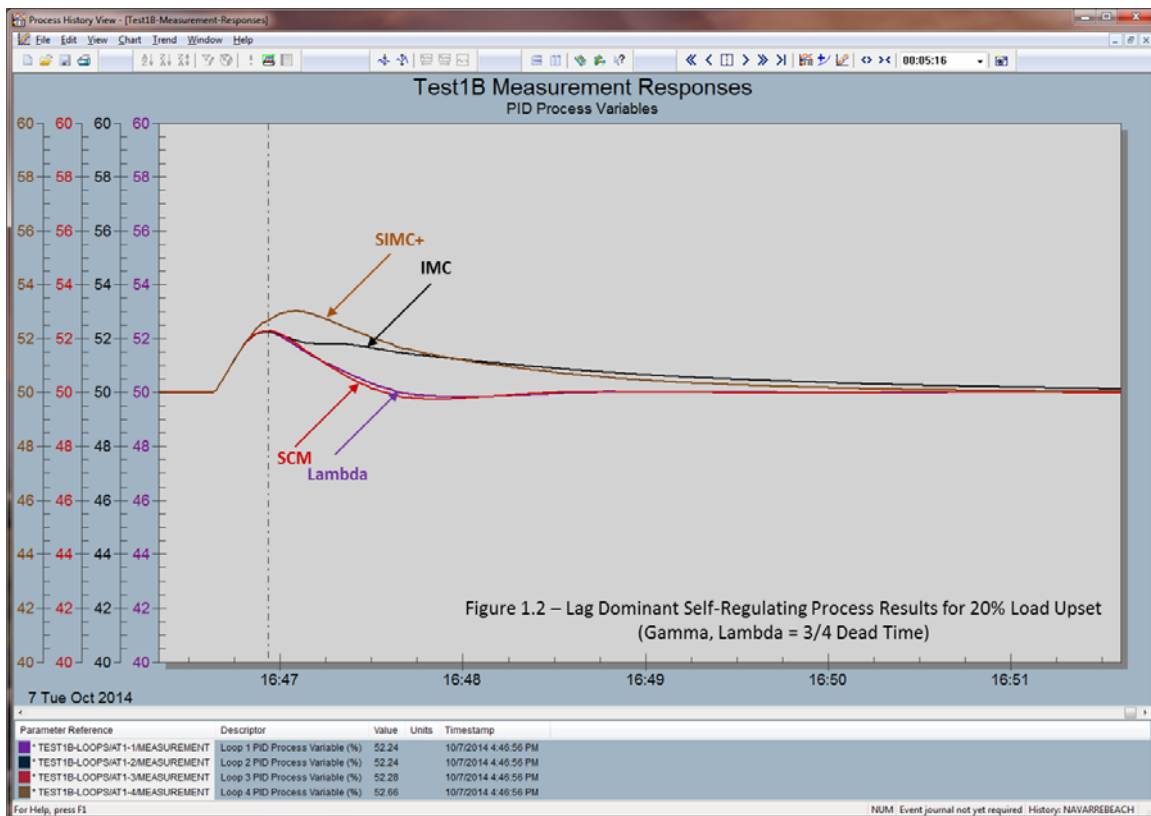
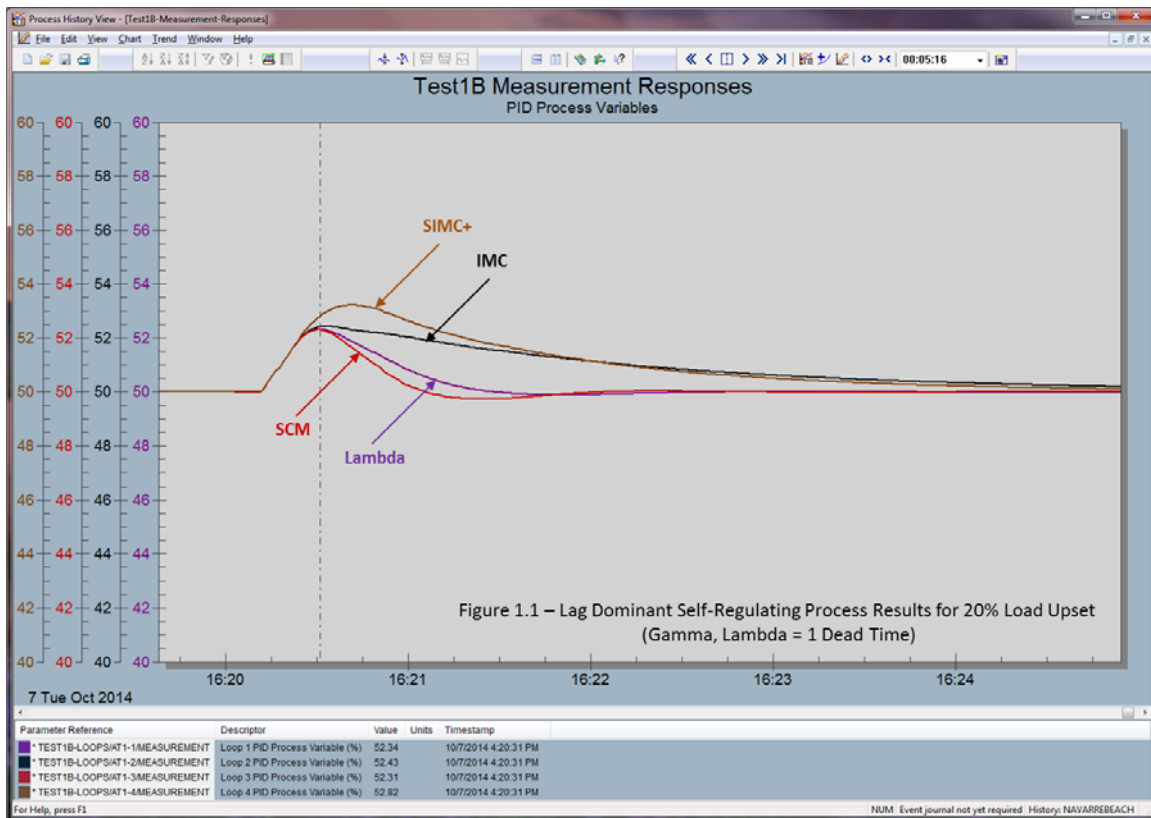
Table 4 – Dead-Time-Dominant, Self-Regulating Process Results for 20% Load Upset

| Loop # & PID Tuning Method | Integrated Absolute Error (%-min) | Peak Error (%) | Primary Time Constant (sec) | Total Loop Dead Time (sec) | Gamma or Lambda (sec) | PID Gain | PID Reset Time (sec) | PID Rate Time (sec) |
|----------------------------|-----------------------------------|----------------|-----------------------------|----------------------------|-----------------------|----------|----------------------|---------------------|
| Loop 1 Lambda (Purple) | 14.2 | 20 | 2 | 20 | 10 | 0.20 | 8 | 1 |
| Loop 2 IMC (Black) | 10.4 FR! | 20 | 2 | 20 | 10 | 0.40 | 12 | 1.7 |
| Loop 3 SCM (Red) | 13 | 20 | 2 | 20 | ----- | 0.20 | 7.3 | 0.4 |
| Loop 4 SIMC+ (Brown) | 14.1 | 20 | 2 | 20 | 10 | 0.22 | 8.7 | 1 |
| Loop 1 Lambda (Purple) | 12.4 | 20 | 2 | 20 | 7.5 | 0.23 | 8 | 1 |
| Loop 2 IMC (Black) | 9.6 OR! | 20 | 2 | 20 | 7.5 | 0.48 | 12 | 1.7 |
| Loop 3 SCM (Red) | 12.9 | 20 | 2 | 20 | ----- | 0.20 | 7.3 | 0.4 |
| Loop 4 SIMC+ (Brown) | 12.3 | 20 | 2 | 20 | 7.5 | 0.25 | 8.7 | 1 |
| Loop 1 Lambda (Purple) | 10.9 | 20 | 2 | 20 | 5.0 | 0.27 | 8 | 1 |
| Loop 2 IMC (Black) | 11.3 OR! | 20 | 2 | 20 | 5.0 | 0.60 | 12 | 1.7 |
| Loop 3 SCM (Red) | 13 | 20 | 2 | 20 | ----- | 0.20 | 7.3 | 0.4 |
| Loop 4 SIMC+ (Brown) | 10.7 | 20 | 2 | 20 | 5.0 | 0.29 | 8.7 | 1 |

FR! - Faltering Response OR! – Oscillatory Response

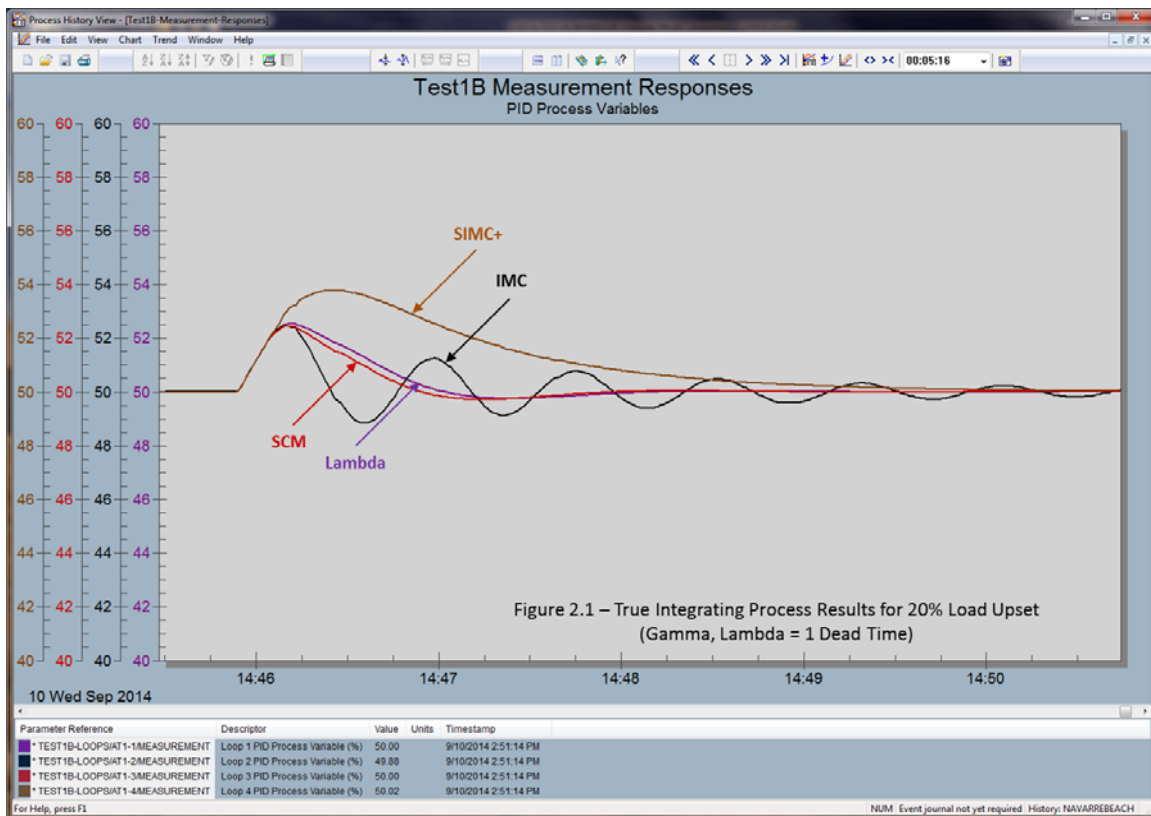
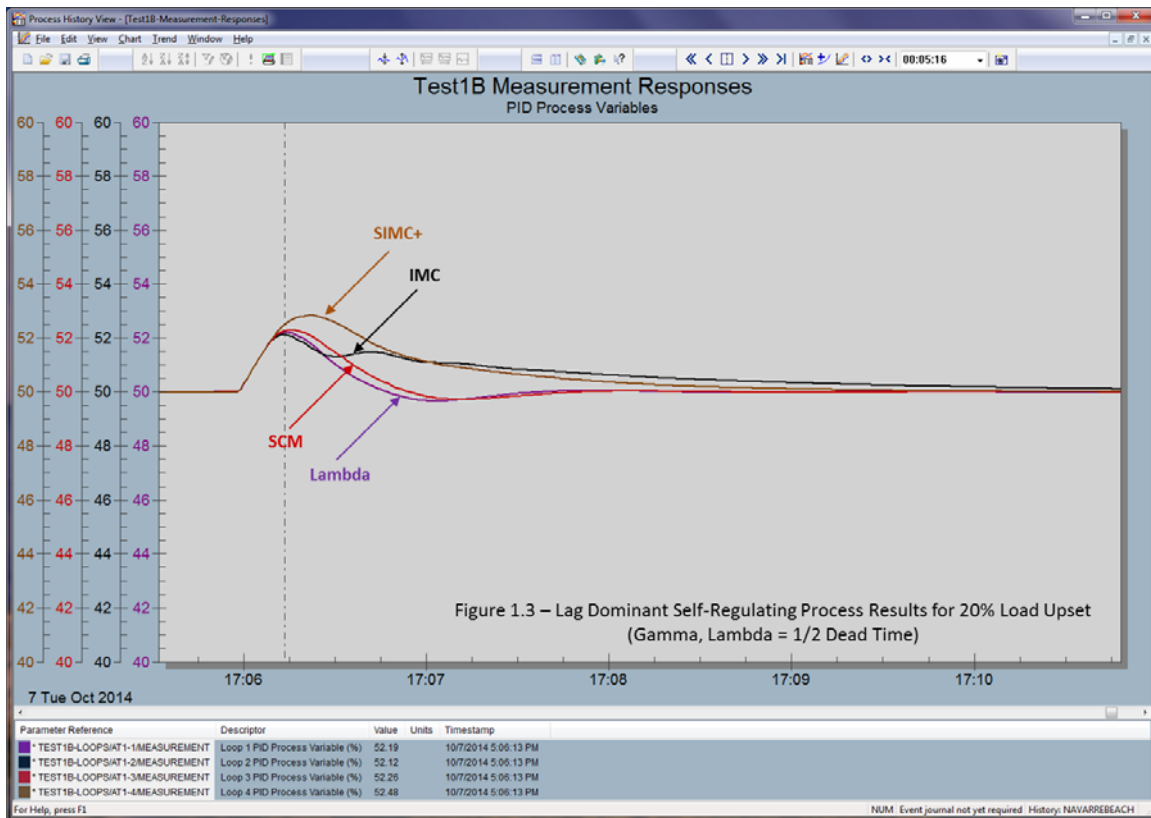
So Many Tuning Rules, So Little Time

Gregory K McMillan



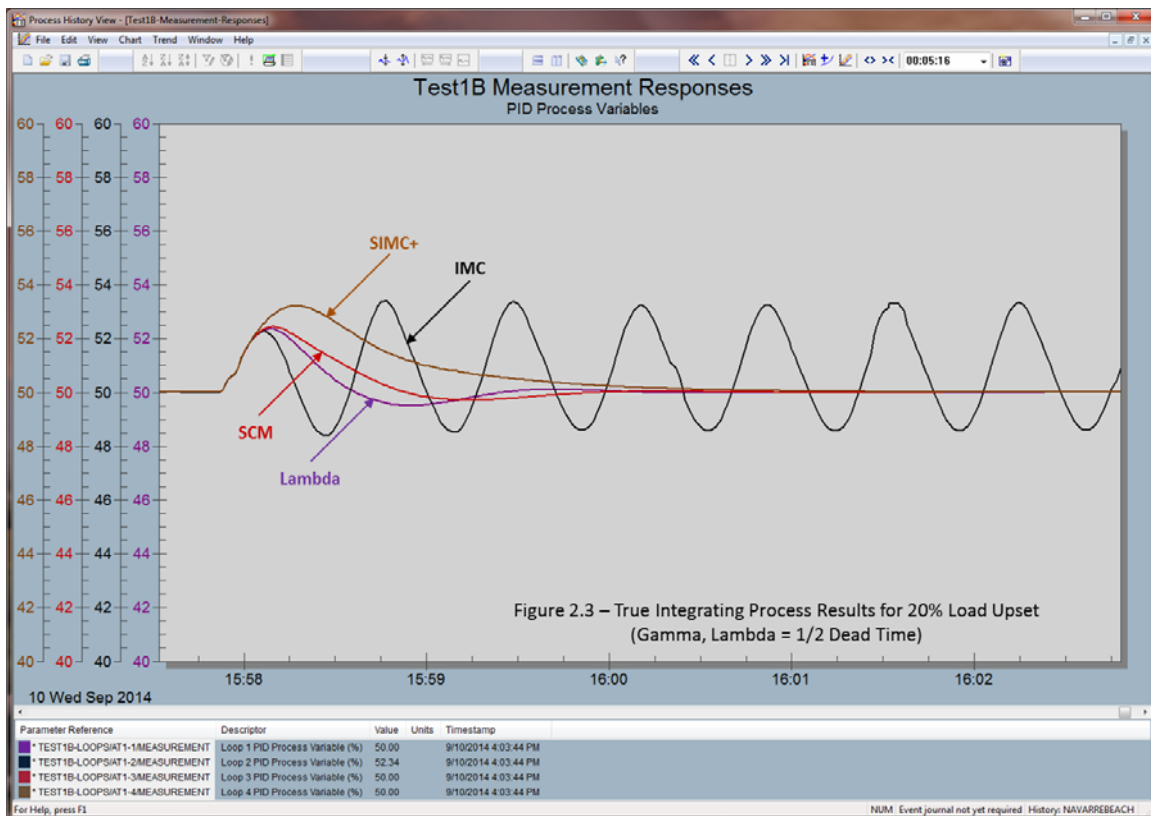
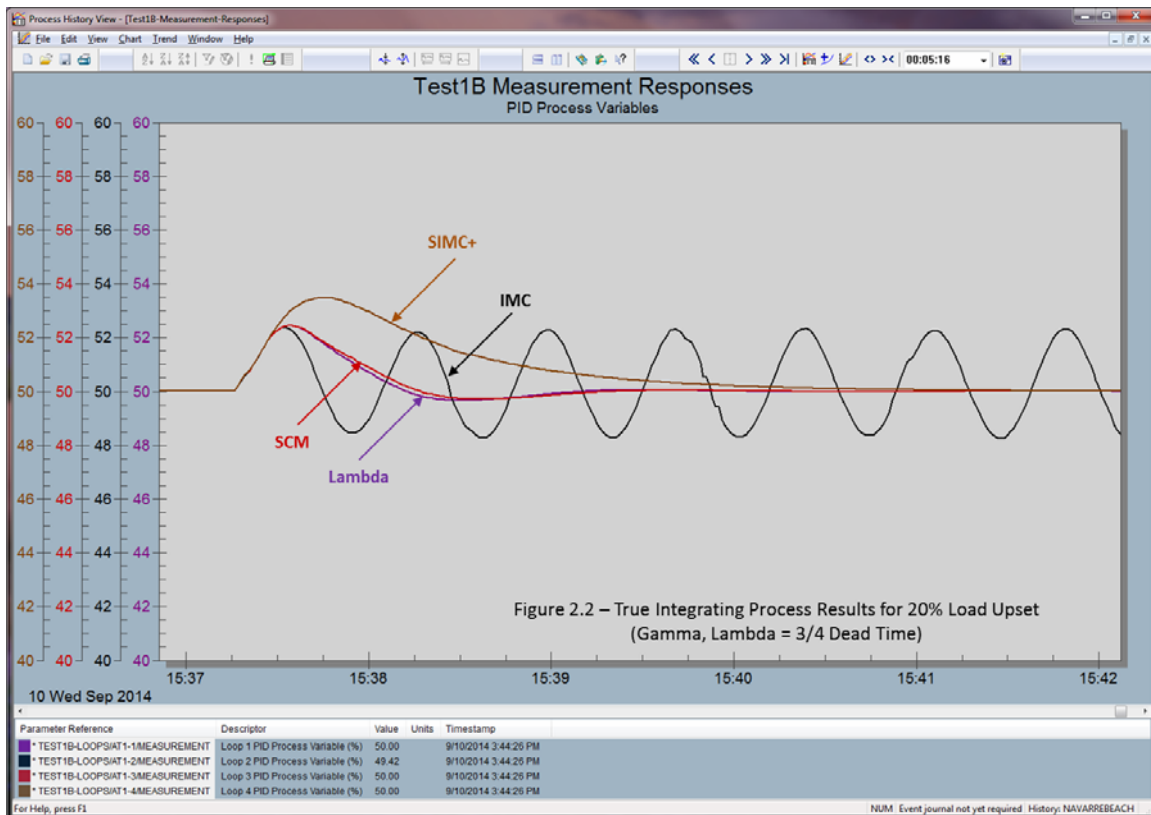
So Many Tuning Rules, So Little Time

Gregory K McMillan



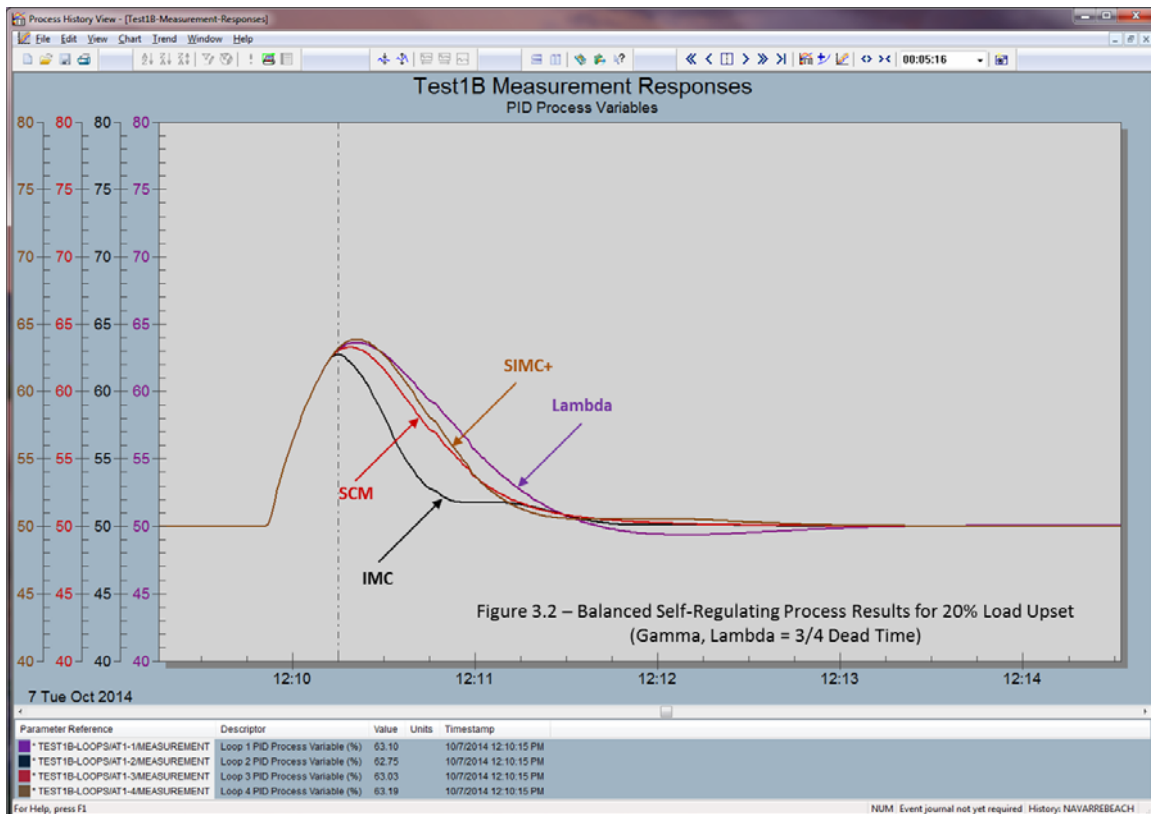
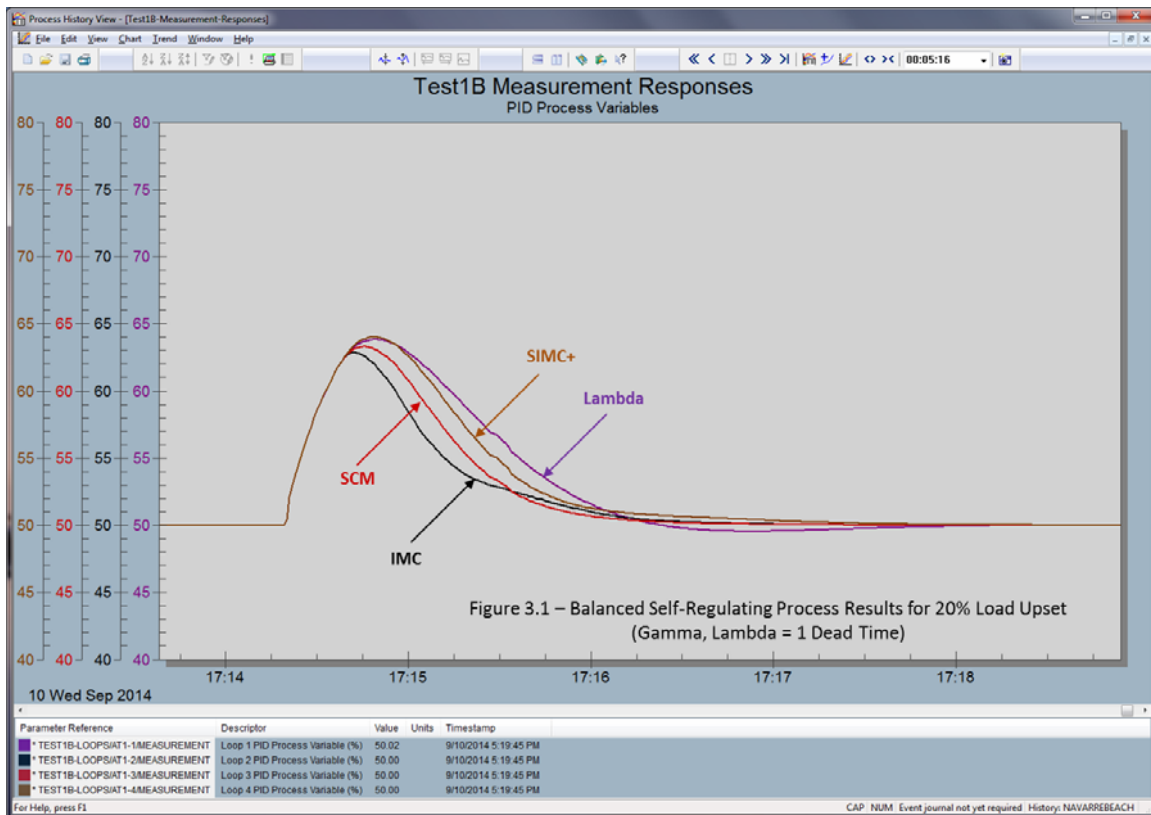
So Many Tuning Rules, So Little Time

Gregory K McMillan



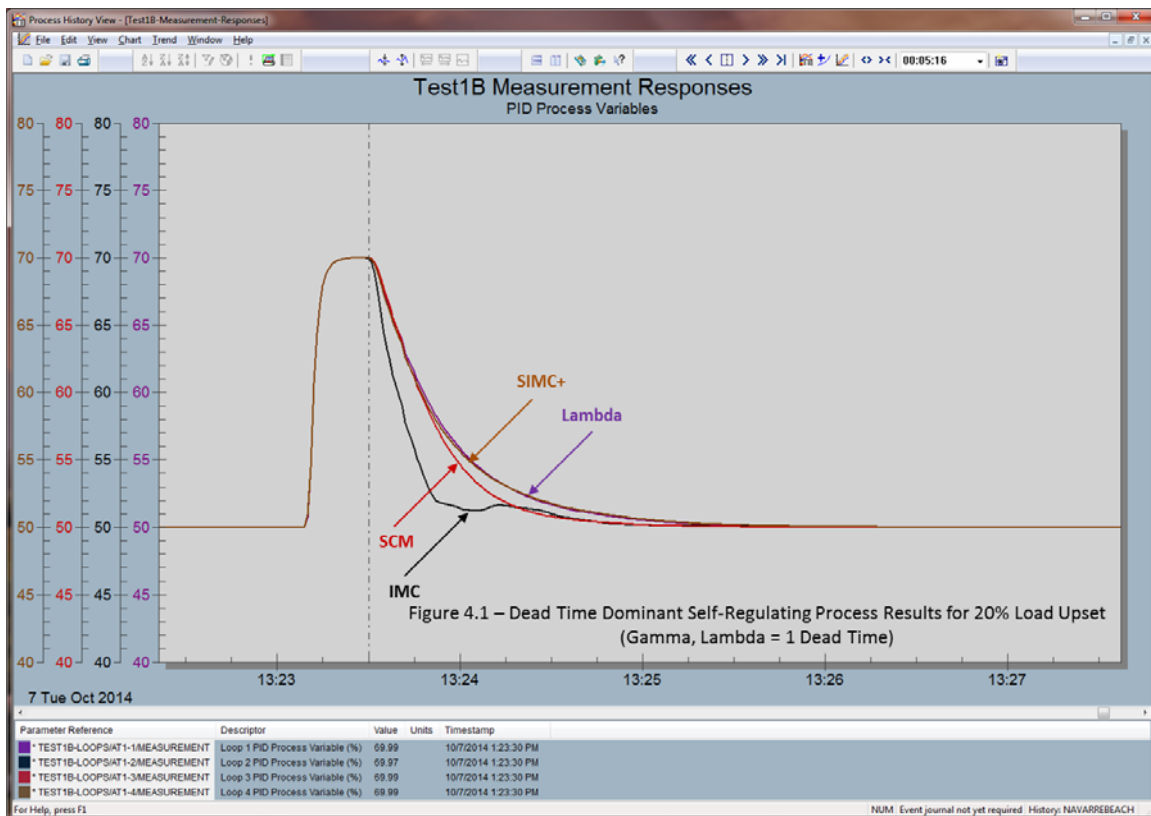
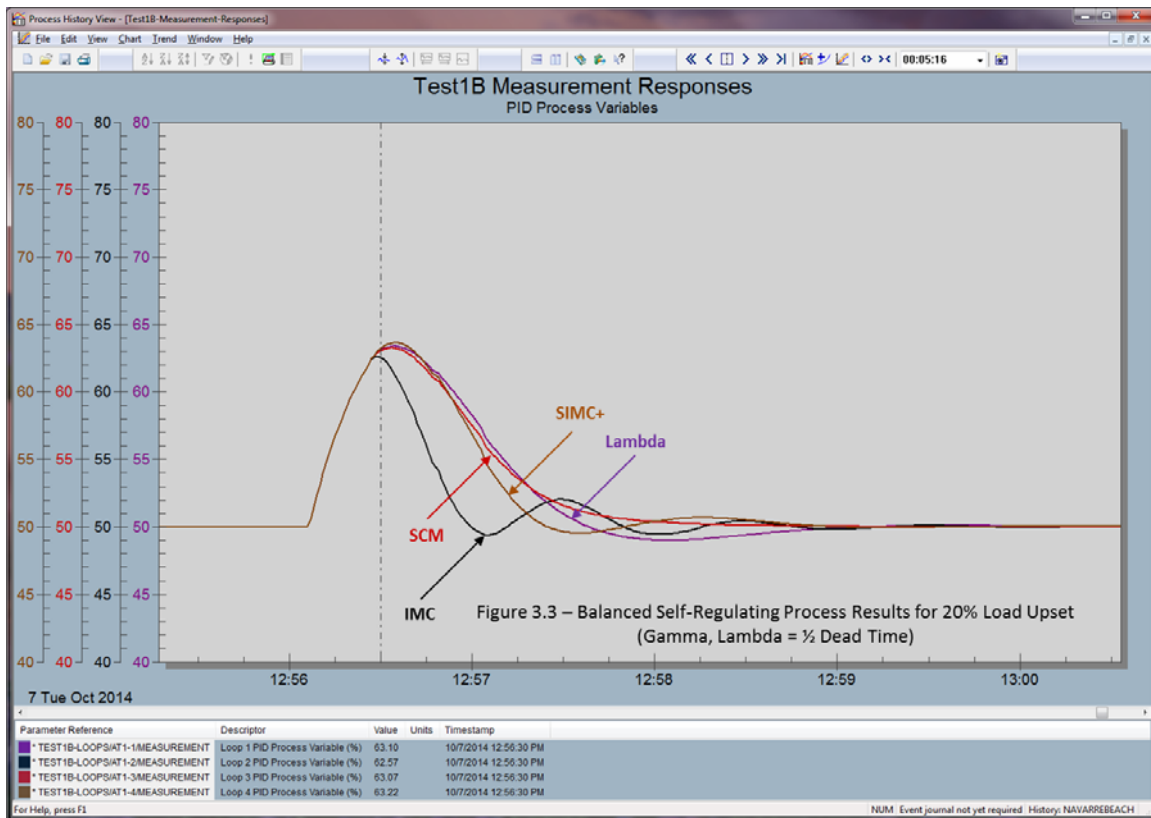
So Many Tuning Rules, So Little Time

Gregory K McMillan



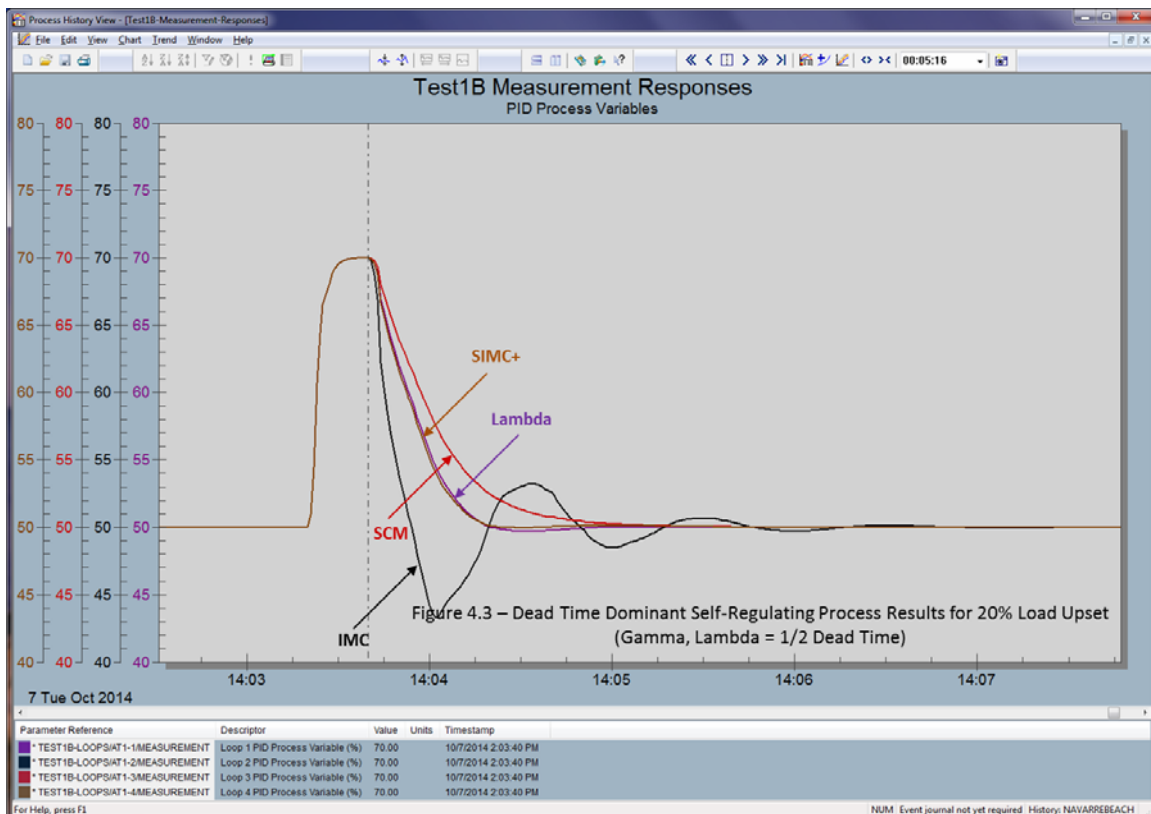
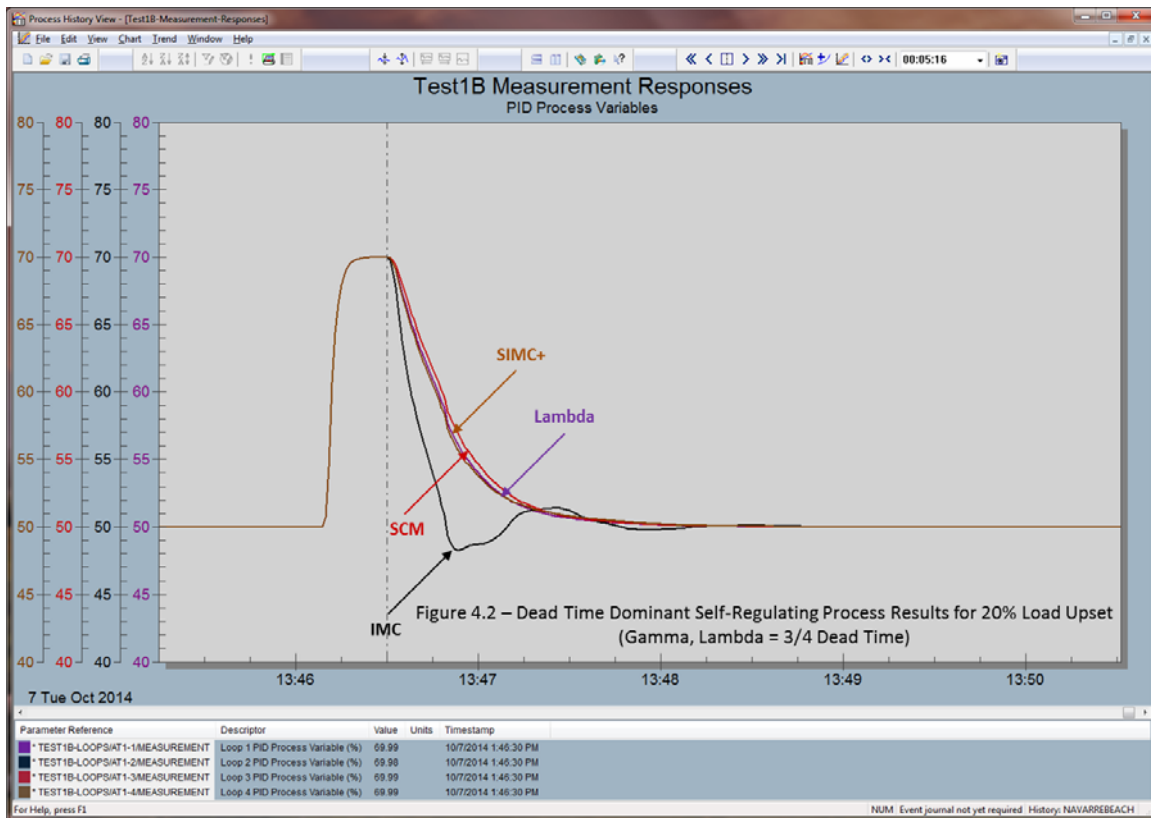
So Many Tuning Rules, So Little Time

Gregory K McMillan



So Many Tuning Rules, So Little Time

Gregory K McMillan



So Many Tuning Rules, So Little Time

Gregory K McMillan

Common Tuning Rules (*Tuning and Control Loop Performance 4th Ed* excerpt)

The incredible range and impact of the size of the primary time constant to the dead time ratio and the location of the disturbance have led to considerable disagreement. A switch to integrating process tuning rules when the primary time constant is much larger than the dead time, and the realization most disturbances are a change in a process input (load upset) result in a convergence of rules when the objective is minimization of integrated error. Often the difference in factors settings quibbled about are less than the estimation error and variability in the dynamic terms (gains, dead time and time constants). The claims by authors of the more than 100 tuning methods are often more a result of gamesmanship rather than reality. Some conclusions are based on tests where the dynamics are changed to make the control less stable. The most common test is to show how the Ziegler-Nichols method causes excessive oscillation for a 25% increase in process gain. The solution could have been to reduce the PID gain by 25%. The other test is to show excessive overshoot for a setpoint change. The solution could have been to simply add a setpoint lead-lag with the lag time equal to the reset time and the lead time equal to 20% of the lag time to eliminate the overshoot and still get to setpoint quickly.

The Ziegler-Nichols method and most other tuning methods before the 1980s were developed to minimize the peak and integrated error for an unmeasured step disturbance on the process input (load upset). While this aggressive action is important in certain applications to prevent the activation of relief systems or initiation of a runaway condition, the robustness is generally insufficient and will not deal with practical problems and other objectives. The more recent tuning methods, such as lambda tuning, focus on adding robustness, minimizing the effect of nonlinearities, interactions and resonance and meeting other process objectives such as maximizing the absorption of variability for surge tank level, the coordination of loops for ratio control, and the consistency of a setpoint response of lower loops for cascade control and model predictive control. The *InTech* January/February 2012 article “PID Tuning Rules” and the online Appendix C show how the actual and minimum errors for load disturbances depend upon PID tuning and process dynamics, respectively.

Most of the control literature focuses on improving the response to a setpoint or a disturbance on the process output for moderate self-regulating processes. Gamesmanship tends to rule and algorithms and tuning methods are touted based on improvements without a realization of the diversity of conditions and demands in actual industrial installations.

Lambda tuning rules are presented because of their flexibility in dealing with different processes, nonlinearities, objectives and extenuating circumstances. Modifications are presented to the normal lambda tuning rules to enable lambda tuning to give a similar load disturbance rejection capability as the short cut method when this is the criterion without interfering with the flexibility of lambda tuning.

So Many Tuning Rules, So Little Time

Gregory K McMillan

A short cut method (SCM) developed by the author is used to provide a benchmark of aggressive tuning settings that will minimize the peak and integrated error from load disturbances (process input disturbances) without causing an oscillatory response if the dynamics are perfectly known and constant. The short cut method first estimates the ultimate period that has many diagnostic uses beyond the computation of tuning settings.

Other tuning methods can be used as long as the advocate uses good software to compute the settings and the window of allowable controller gains for integrating and runaway processes are not violated. More important than the tuning rules is the actual tuning of the loop with good software, including any change in settings by an experienced practitioner to address unknowns, nonlinearities, varying objectives and extenuating circumstances. Consultants often tout their tuning rules when in actuality the process of intelligently tuning a loop was more important than the rules used. PID tuning totally set manually based on intuition is in most cases messed up.

Consultants have such pride and time invested in their tuning rules, the improvement in a loop is often more attributed to the rules than the software that gets setting based on identified dynamics and the intelligence used to modify the settings to deal with the conditions and meet the demands of industrial applications.

The use of lambda rather than lambda factors offers many important advantages. Consultants in the use of lambda tuning for industrial processes do not enter a lambda factor, but an actual lambda that is a closed-loop time constant for self-regulating processes and an arrest time for integrating processes. Lambda typically ranges from one dead time to 3 dead times to meet application requirements. Limits are imposed on the reset and rate times to help maximize load rejection. Finally processes with a time-constant-to-dead-time ratio greater than four are treated as near-integrating processes, and lambda tuning rules for integrating processes are used.

All of the equations assume the maximum open-loop self-regulating process gain or maximum open-loop integrating process gain, minimum primary time constant, and maximum dead time and secondary time constant were evaluated at different operating conditions, and the worst-case condition is used that results in the smallest gain and rate time and largest reset time. At low production rates, the dead time and open-loop gain is often the largest for composition and temperature control of well-mixed liquid volumes. If these dynamic terms vary, adaptive tuning or scheduling of tuning settings is needed.

You can convert an open-loop, self-regulating process (steady state) gain (K_o) for a process with a primary time constant (τ_o) much larger than the dead time (θ_o) to an open-loop integrating process gain (K_i) by the use of Equation 1.5a developed for the near-integrator approximation to enable you to use tuning rules for integrating processes to improve the disturbance rejection. The equation can also save a huge amount of time in the identification of the process dynamics, since the largest ramp rate, $Max(\Delta\% PV / \Delta t)$, in the right direction within four dead times per Equation 1.5b is used to compute the

So Many Tuning Rules, So Little Time

Gregory K McMillan

integrating process gain instead of waiting for the process to essentially reach a steady state (e.g. 98% response time) to identify the primary time constant and open-loop steady state gain. For a time constant that is 20 times the dead time (not uncommon for well-mixed liquid reactors), the identification time is 20 times faster.

$$K_i = \frac{K_o}{\tau_o} \quad (1.5a)$$

$$K_i = \frac{\text{Max}(\Delta\% PV / \Delta t)}{\Delta\% CO} \quad (1.5b)$$

For integrating processes, the product of the controller gain (K_c) and integral time (reset time) setting (T_i) must be greater than twice the inverse of the open-loop integrating process gain (K_i) to prevent the start of slow rolling oscillations.

$$K_c * T_i > \frac{2}{K_i} \quad (1.5c)$$

Since most PID on integrating processes have a controller gain much less than the maximum allowed, the equation is reformulated to show the minimum integral time.

$$T_i > \frac{2}{K_c * K_i} \quad (1.5d)$$

The oscillations will decay slowly unless the following inequality is enforced:

$$T_i > \frac{1}{4 * K_c * K_i} \quad (1.5e)$$

Recent test results show that the above rules apply to near-integrating and runaway processes as well. As you go from near-integrating to true-integrating to runaway processes, the consequences of violating these rules get more severe in that the oscillations are larger and slower to decay.

There is not enough space or time to cover all of the tuning rules effectively used in industry. Here we focus on seven major sets of tuning rules. The section concludes with a description of how to compute the arrest time to maximize the absorption of variability for level control in surge tanks and other volumes when the manipulated flow is the feed to a downstream unit operation. Since the nomenclature is extensive and necessary to a full understanding, we start with the nomenclature definition.

So Many Tuning Rules, So Little Time

Gregory K McMillan

Nomenclature for Tuning Calculations

A = cross-sectional area of surge tank (m^2)

AR_{-180} = amplitude ratio at -180 degrees phase shift (dimensionless)

a = gain factor in traditional open-loop methods (0.4 to 1.0)

b = reset time factor in traditional open-loop methods (0.5 to 4.0)

c = rate time factor in traditional open-loop methods (0 to 1.0)

K_c = controller gain (dimensionless)

K_i = open-loop integrating process gain (1/sec)

K_m = measurement gain (%/m for level)

K_p = process gain (m/kg for level)

K_v = valve or variable-speed drive gain (kg/sec/%)

K_o = open-loop self-regulating process (steady state) gain (%/%) (dimensionless)

K_p' = open-loop runaway process gain (%/%) (dimensionless)

K_q = controller gain that causes quarter amplitude oscillations (dimensionless)

K_u = controller ultimate gain (dimensionless)

L = Ziegler-Nichols lag graphically estimated as intersection with original PV of a tangent to the inflection point of the PV open-loop response (sec)

N_m = measurement noise amplitude (%)

R = Ziegler-Nichols ramp rate sensitivity graphically estimated as slope of tangent to inflection point of the PV open-loop response (% per sec per %)

S_m = measurement threshold sensitivity limit (%)

S_v = valve stick-slip or resolution limit (%)

t_{fsr} = full-scale residence time (sec)

T_i = controller integral time (reset time) (sec)

T_d = controller derivative time (rate time) (sec)

T_q = quarter amplitude period (sec)

T_u = ultimate period (sec)

% CO = operating point controller output (%)

% CO_{Limit} = controller output limit (%)

$\Delta\%CO_{max}$ = maximum available change in controller output (%)

% PV_{Limit} = process variable limit (%)

$\Delta\%PV_{max}$ = maximum allowable change in process variable (%)

% SP = operating point setpoint (%)

ΔF_{max} = maximum change in valve or variable-speed drive flow (e.g. flow span) (m/sec)

So Many Tuning Rules, So Little Time

Gregory K McMillan

ΔL_{\max} = maximum change in level (e.g. level span) (m)

γ_f = gamma factor for IMC closed-loop time constant or arrest time (dimensionless)

γ = gamma for IMC closed-loop time constant or arrest time (sec)

λ_f = lambda factor for closed-loop time constant or arrest time (dimensionless)

λ = lambda for closed-loop time constant or arrest time (sec)

θ_o = total loop dead time (sec)

τ_o = primary time constant (open-loop time constant - largest time constant in loop)
(sec)

τ_s = secondary time constant (second largest time constant in loop) (sec)

τ_p' = positive feedback process time constant (largest time constant in loop) (sec)

ω_n = natural frequency (critical frequency) (radians/sec)

ρ = liquid density in surge tank (kg/m³)

Lambda Tuning for Self-Regulating Processes

The lambda tuning method for self-regulating processes is advocated by the author for use when the open-loop time constant is less than 4 times the total loop dead time. For self-regulating processes that are not near-integrating ($\tau_o < 4 * \theta_o$), we have the following series of equations.

The reset time is set equal to the open-loop time constant. The reset time steadily decreases as the ratio of the time constant to dead time decreases from its maximum of 4 dead times if this tuning method is used when the open-loop time constant is less than 4 times the total loop dead time. For systems with an open-loop time constant much less than the dead time (dead time dominant), the PID action becomes essentially integral only due to a steady decrease in the controller gain and the reset time. This type of control helps deal with the abrupt almost step changes and noise in a dead time dominant system. Thus, the PID controller takes over the role of providing a gradual and smoothing response that is missing in the process. The best scenario is to have a large process time constant to fulfill this role, but for plug flow and sheet line processes, this is not possible.

$$T_i = \tau_o \quad (1.6a)$$

A low limit is added to traditional equations to prevent a reset time smaller than ¼ the dead time for loops where the time constant is very small to provide some gain action.

$$T_i = \text{Max} [0.25 * \theta_o, \tau_o] \quad (1.6b)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$K_c = \frac{T_i}{K_o * (\lambda_f * \tau_o + \theta_o)} \quad (1.6c)$$

In industrial applications the lambda factor multiplication of the open-loop time constant is replaced with lambda, the closed-loop time constant ($\lambda = \lambda_f * \tau_o$), which is the time after the dead time to reach 63% of a setpoint change. The use of lambda rather than lambda factor provides the recognition that lambda is set as a multiple of the dead time to minimize variability in the PID input or output, and to provide the gain margin needed to deal with extenuating circumstances (e.g. interaction, inverse response and resonance).

$$K_c = \frac{T_i}{K_o * (\lambda + \theta_o)} \quad (1.6d)$$

For maximum unmeasured disturbance rejection, a lambda equal to the dead time is used:

$$K_c = 0.5 * \frac{\tau_o}{K_o * \theta_o} \quad (1.6e)$$

A lambda of 3 to 5 dead times minimizes the consequences of nonlinearities, inverse response and resonance.

Normally a rate setting is not included as part of the lambda tuning for self-regulating processes. If the primary time constant is greater than ½ the dead time, rate action may be beneficial. If a secondary time constant (next largest time constant) can be identified, the rate time is set equal to the secondary time constant. The rate time should be larger than ½ the dead time, but not be greater than ¼ the reset time for an ISA Standard Form).

If the primary time constant is greater than ½ the dead time ($\tau_o > 0.5 * \theta_o$):

$$T_d = \text{Min} [0.25 * T_i, \text{Max} (0.5 * \theta_o, \tau_s)] \quad (1.6f)$$

Lambda Tuning for Integrating Processes

The lambda tuning method for integrating processes is advocated by the author for use on self-regulating processes when the open-loop time constant is greater than 4 times the total loop dead time (near-integrating processes) besides for use on true-integrating and runaway processes to provide a faster return to setpoint for load upsets.

For PID control, the reset time is twice the arrest time plus the dead time.

$$T_i = 2 * \lambda + \theta_o \quad (1.7a)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

For PI control, a low limit is added so that the reset time is not less than 4 dead times.

$$T_i = \text{Max} [4 * \theta_o, 2 * \lambda + \theta_o] \quad (1.7b)$$

$$K_c = \frac{T_i}{K_i * \left(\frac{\lambda_f}{K_i} + \theta_o \right)^2} \quad (1.7c)$$

In industrial applications the lambda factor multiplication of the inverse of the integrating process gain is replaced with lambda, the closed-loop arrest time ($\lambda = \lambda_f / K_i$), which is the time to stop an excursion for an unmeasured disturbance (time to peak error). The use of lambda rather than lambda factor provides the recognition that lambda is set as a multiple of the dead time to minimize variability in either the PID input or output and to provide the gain margin needed to deal with extenuating circumstances (e.g., interaction, inverse response and resonance) or is set for surge tank level control to provide the maximum absorption of variability.

The numerator uses the unlimited reset time.

$$K_c = \frac{2 * \lambda + \theta_o}{K_i * (\lambda + \theta_o)^2} \quad (1.7d)$$

For maximum unmeasured disturbance rejection, a lambda equal to the dead time is used:

$$K_c = 0.75 * \frac{1}{K_i * \theta_o} \quad (1.7e)$$

A lambda of 3 to 5 dead times minimizes the consequences of nonlinearities, inverse response and resonance.

The rate time should be greater than 1/2 the dead time, but less than 1/4 the reset time.

$$T_d = \text{Min} [0.25 * T_i, \text{Max} (0.5 * \theta_o, \tau_s)] \quad (1.7f)$$

Internal Model Control Tuning for Self-Regulating Processes

The internal model control (IMC) method was developed to provide a controller that is the inverse of the process dynamics. Similar to lambda tuning, IMC tuning uses pole-zero cancellation theory. The IMC tuning rules vary with author, dead time approximation and vintage. The IMC rules presented here are documented in the book *Advanced Control Unleashed* (Blevins, 2003). Many authors have offered improvements for different

So Many Tuning Rules, So Little Time

Gregory K McMillan

dynamics. Most notably are the ones by Skogestad documented in “Simple Analytic Rules for Model Reduction and PID Controller Tuning” (Skogestad, 2003).

For self-regulating processes:

$$K_c = \frac{T_i}{K_o * (\gamma_f * \tau_o + 0.5 * \theta_o)} \quad (1.8a)$$

In industrial applications the gamma factor multiplication of the open-loop time constant is replaced with gamma, the closed-loop time constant ($\gamma = \gamma_f * \tau_o$) for a setpoint change.

$$K_c = \frac{T_i}{K_o * (\gamma + 0.5 * \theta_o)} \quad (1.8b)$$

The reset time is set equal to the open-loop time constant plus ½ the total loop dead time. By including the dead time, IMC tuning prevents the PID from using incredibly small reset times and small gains when the time constant is much less than the dead time ($\tau_o \ll \theta_o$). Thus, the IMC tuning behaves more like the modified Ziegler-Nichols tuning for dead time dominance. However, the reset time is even larger than optimum for systems where the time constant is much greater than the dead time ($\tau_o \gg \theta_o$) resulting in an even slower recovery and, consequently, much larger integrated error from unmeasured disturbances. The user can take the same approach recommended for lambda tuning, which is to use a near-integrator tuning when the time constant becomes four times larger than the dead time ($\tau_o > 4 * \theta_o$).

$$T_i = \tau_o + 0.5 * \theta_o \quad (1.8c)$$

For maximum unmeasured disturbance rejection, a gamma equal to the dead time is used:

$$K_c = 0.7 * \frac{\tau_o}{K_o * \theta_o} \quad (1.8d)$$

$$T_i = \text{Max} (2 * \Delta t_x, \tau_o + 0.5 * \theta_o) \quad (1.8e)$$

The IMC rate term is quite different in being proportional to the product of the open loop time constant and total loop dead time. The rate time should not be greater than ¼ the reset time for an ISA Standard Form.

$$T_d = \text{Min} (0.25 * T_i, \frac{\tau_o * \theta_o}{2 * \tau_o + \theta_o}) \quad (1.8f)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

Internal Model Control Tuning for Integrating Processes

The IMC tuning method is similar to the lambda tuning method for integrating processes, except for the use of ½ of the dead time instead of the whole dead time in the denominator to calculate the controller gain.

$$K_c = \frac{T_i}{K_i * (\frac{\gamma_f}{K_i} + 0.5 * \theta_o)^2} \quad (1.9a)$$

In industrial applications the gamma factor multiplication of the inverse of the integrating process gain is replaced with gamma, the closed-loop arrest time ($\gamma = \gamma_f / K_i$) for an unmeasured disturbance.

$$K_c = \frac{T_i}{K_i * (\gamma + 0.5 * \theta_o)^2} \quad (1.9b)$$

The reset time is twice the arrest time plus the dead time.

$$T_i = 2 * \gamma + \theta_o \quad (1.9c)$$

For maximum unmeasured disturbance rejection, a gamma equal to the dead time is used:

$$K_c = 1.3 * \frac{1}{K_i * \theta_o} \quad (1.9d)$$

$$T_i = 3 * \theta_o \quad (1.9e)$$

If a secondary time constant (next largest time constant) can be identified, derivative action can be used with the rate time set equal to the secondary time constant. The rate time should not be greater than ¼ the reset time for an ISA Standard Form.

$$T_d = \text{Min} [0.25 * T_i, \tau_s] \quad (1.9f)$$

Skogestad Internal Model Control Tuning for Self-Regulating Processes

Sigurd Skogestad recognized the problem with incredibly slow and fast integral action in IMC and lambda tuning for processes with an extremely large and small primary time constant, respectively because the reset time was set proportional to this time constant. The Skogestad internal model control (SIMC) rules prevent the reset time from becoming exceptionally large or small. The rules end up with about the same controller gain as

So Many Tuning Rules, So Little Time

Gregory K McMillan

lambda tuning for self-regulating processes. Here we will use gamma (γ) employed by IMC tuning instead of the closed-loop time constant (τ_c) used in the paper “Simple Analytic Rules for Model Reduction and PID Controller Tuning” (Skogestad, 2003). The rules presented here are for PID control of a process with dead time and a large open-loop time constant (τ_o) and a small secondary time constant (τ_s). The reset time and gain have been modified to provide a larger controller gain and a larger reset time for dead-time-dominant processes per paper “Performance and Robustness Trade-offs in PID control” (Garpinger et. al., 2014). The revised rules are termed SIMC+. If gamma is set relative to dead time instead of a multiple of the closed-loop time constant, there is no change in the tuning settings until the primary time constant becomes much larger than the total loop dead time. For near-integrating processes, the SIMC+ rules provide a larger PID gain and thus a smaller peak and integrated error for gamma equal to less than the total loop dead time. Thus, the real benefit turns out to be for processes that are the opposite of what was stated in the SIMC+ paper.

$$K_c = \frac{\tau_o + \theta_o / 3}{K_o * (\gamma + \theta_o)} \quad (1.10a)$$

$$T_i = \text{Min} [\tau_o + \theta_o / 3, 4 * (\gamma + \theta_o)] \quad (1.10b)$$

$$T_d = \text{Min} [0.25 * T_i, \tau_s] \quad (1.10c)$$

Skogestad Internal Model Control Tuning for Integrating Processes

For integrating processes, the SIMC reset time is about 270% larger than the reset time per the lambda and IMC tuning methods for maximum disturbance rejection where gamma is set equal to the dead time. The SIMC controller gain is about 30% smaller for this case. The product of the SIMC gain and reset time is about twice as large as for lambda tuning, giving more margin to prevent slow rolling oscillations (Equation 1.5c).

$$K_c = \frac{1}{K_i * (\gamma + \theta_o)} \quad (1.11a)$$

$$T_i = 4 * (\gamma + \theta_o) \quad (1.11b)$$

$$T_d = \text{Min} [0.25 * T_i, \tau_s] \quad (1.11c)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

Traditional Open-Loop Tuning

The following equations are for an ISA Standard Form PID controller. The limit on the rate time ensures the rate time is not larger than ¼ the reset time. The a, b and c factors are decreased as the ratio of time constant to dead time decreases. The minimum numbers in the nomenclature definition are for a time constant much less than the dead time (dead-time-dominant). Without the limits on the controller gain and reset time, the controller becomes exceptionally slow for dead-time-dominant loops.

$$K_c = \text{Max} \left[0.2 * \frac{1}{K_o}, a * \left(\frac{\tau_o}{K_o * \theta_o} \right) \right] \quad (1.12a)$$

$$T_i = \text{Max} [0.4 * \theta_o, b * \theta_o] \quad (1.12b)$$

$$T_d = \text{Min} [0.25 * T_i, c * \theta_o] \quad (1.12c)$$

Modified Ziegler-Nichols Reaction Curve Tuning

The Ziegler-Nichols reaction curve method, unlike the ultimate oscillation method, is conducted with the controller in manual. A step change is made in the manual controller output ($\Delta \% CO$), and a maximum ramp rate per percent change in controller output is estimated. The original paper shows this parameter R as being graphically estimated as the slope of a tangent line to the inflection point of a self-regulating process open-loop response that goes to completion. The L parameter is used to denote a lag that is estimated as the time from the controller output change to the intersection of the tangent with the original PV. The official definition of a lag is any phase lag that can be due to a dead time or a time constant. In most publications today, lag time is used interchangeably with time constant whereas Ziegler-Nichols was using lag time as a dead time. The method is modified to provide a smoother than quarter-amplitude response and to add some robustness by applying a 0.5 factor to the ZN gain as shown in Equation 1.13a.

The parameter R is really the integrating process gain that can be measured online by passing a new $\% PV$ through a dead time block whose parameter is the total loop dead time to create an old $\% PV$ that is subtracted from the new $\% PV$ to create a delta $\% PV$. The maximum of the delta $\% PV$ results divided by the dead time (θ_o), and the change in controller output is the maximum ramp rate per percent change in controller output.

$$K_c = 0.5 * \frac{1}{R * L} \quad (1.13a)$$

$$T_i = 3 * L \quad (1.13b)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

The max ramp rate per percent change in controller output is the integrating process gain.

$$R = \frac{Max\left(\Delta\%PV/\Delta t\right)}{\Delta\%CO} = K_i \quad (1.13c)$$

The L parameter is the observed loop dead time

$$L = \theta_o \quad (1.13d)$$

If we substitute the equations for the definition of R and L parameters, we end up with the lambda tuning Equations 1.7d and 1.7e for integrating processes when lambda is set equal to the dead time.

$$K_c = 0.5 * \frac{1}{K_i * \theta_o} \quad (1.13e)$$

$$T_i = 3 * \theta_o \quad (1.13f)$$

The integrating process gain for a near-integrating process is the open-loop gain divided by the open-loop time constant.

$$K_i = \frac{K_o}{\tau_o} \quad (1.13g)$$

If we substitute Equation 1.13g into Equation 1.13e, we end up with Equation 1.13d that is the lambda tuning Equation 1.6d for a self-regulating process for a 3:1 time-constant-to-dead-time ratio when lambda is set equal to the dead time. The reset time is the same as Equation 1.13f, since the time constant is equal to 3 times the dead time.

$$K_c = 0.5 * \frac{\tau_o}{K_o * \theta_o} \quad (1.13h)$$

Modified Ziegler-Nichols Ultimate Oscillation Tuning

The Ziegler-Nichols ultimate oscillation tuning method is conducted with the controller in automatic. If the loop is lined out, the controller is momentarily put in manual, and a step change is made in the controller output, and the controller is immediately returned to automatic. With the reset time at a maximum (more than 100 times greater than the dead time) and the rate time set to zero to give essentially a proportional only controller, the controller gain is increased until there are equal amplitude oscillations. The controller

So Many Tuning Rules, So Little Time

Gregory K McMillan

gain that caused these oscillations is the ultimate gain, and the period of the oscillations is the ultimate period. Generally this technique is too exciting in that the loop is on the border line of instability where the oscillations could rapidly grow in magnitude. Consequently, the manual quarter-amplitude oscillation method or the relay auto tuner is preferred to get the ultimate gain and period. Here we look at how the Ziegler-Nichols ultimate oscillation method gives about the same results as the much more practical Ziegler-Nichols reaction curve method.

Starting with fundamental relationship that ultimate gain is the inverse of the product of the open-loop gain and amplitude ratio at -180 degrees phase shift.

$$K_u = \frac{1}{K_o * AR_{-180}} \quad (1.14)$$

For self-regulating single time constant processes the amplitude ratio at -180 degrees is:

$$AR_{-180} = \frac{1}{\sqrt{1 + (\tau_o * \omega_n)^2}} \quad (1.15a)$$

$$K_u = \frac{\sqrt{1 + (\tau_o * \omega_n)^2}}{K_o} \quad (1.15b)$$

Using natural frequency relationship to ultimate period $\omega_n = \frac{2 * \pi}{T_u}$

$$K_u = \frac{\sqrt{1 + (\tau_o * 2 * \pi / T_u)^2}}{K_o} \quad (1.15c)$$

For a loop dominated by a large time constant ($\tau_o \gg \theta_o \Leftrightarrow T_u \ll \tau_o$), the ultimate gain equation simplifies to:

$$K_u = \frac{2 * \pi * \tau_o}{K_o * T_u} \quad (1.15d)$$

For an ultimate period being about 4 dead times ($T_u \cong 4 * \theta_o$):

$$K_u = \frac{2 * \pi * \tau_o}{K_o * 4 * \theta_o} \quad (1.15e)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

If we use the Ziegler-Nichols ultimate oscillation equations for a PID controller, we end up with the Ziegler-Nichols reaction curve method tuning, except the reset factor is 2 instead of 3. The rate time for the ISA Standard Form is $\frac{1}{4}$ the reset time. We can convert back and forth between a lag-dominant and near-integrator calculation for the controller gain by the use of Equation 1.13g to convert the dead time to time constant ratio to an integrating process gain. The method is modified to provide a smoother than quarter-amplitude response and to add some robustness by applying a 0.5 factor to the ZN gain as shown in Equation 1.13a.

$$K_c = 0.5 * 0.6 * K_u = 0.5 * 0.6 * \frac{2 * \pi * \tau_o}{K_o * 4 * \theta_o} = 0.5 * \frac{\tau_o}{K_o * \theta_o} \quad (1.16a)$$

$$T_i = 0.5 * T_u = 0.5 * 4 * \theta_o = 2 * \theta_o \quad (1.16b)$$

$$T_d = 0.125 * T_u = 0.125 * 4 * \theta_o = 0.5 * \theta_o \quad (1.16c)$$

Even if the Ziegler-Nichols ultimate oscillation method is not used, knowing the ultimate gain from using the relay tuner offers knowledge of how close the loop is to instability. The gain margin is the ratio of the ultimate gain to the current PID gain. This gain margin is extensively used to deal with not only changes in open-loop gain, but also dead time and time constants

Quarter-Amplitude Oscillation Tuning

The Ziegler-Nichols ultimate oscillation method has been heavily criticized for being too disruptive and potentially unsafe by requiring the user to create equal amplitude oscillations that put the loop on the verge of instability. The quarter-amplitude method prevents excessive oscillations during the closed-loop test to identify the loop dynamics and tuning settings by only pushing the loop to rapidly decaying quarter-amplitude oscillations. This method also prevents mistaking limit cycles, which have equal amplitude oscillations, for ultimate oscillations. The quarter-amplitude period and controller gain are then used to approximate the ultimate period and gain, given some of the non-ideal effects. While the method is not as accurate as ultimate oscillation method, the error is usually well within the uncertainty of tuning settings due to nonlinearities. The controller gains are cut in half via a 0.5 factor to give a smoother response.

For an ISA Standard Form proportional only (P only) controller:

$$K_c = 0.5 * 0.6 * K_u = 0.3 * K_u \quad (1.17a)$$

For an ISA Standard Form proportional-integral (PI) controller:

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$K_c = 0.5 * 0.4 * K_u = 0.2 * K_u \quad (1.17b)$$

$$T_i = 0.8 * \frac{T_u}{\text{Min}(4, 10 * (\frac{4 * \theta_o}{T_u} - 1)^2 + 1)} \quad (1.17c)$$

For an ISA Standard Form proportional-integral-derivative (PID) controller:

$$K_c = 0.5 * 0.6 * K_u = 0.3 * K_u \quad (1.17d)$$

$$T_i = 0.6 * \frac{T_u}{\text{Min}(4, 10 * (\frac{4 * \theta_o}{T_u} - 1)^2 + 1)} \quad (1.17e)$$

$$T_d = 0.8 * \text{Max} [0.0, 0.25 * (T_i - 0.5 * \theta_o)] \quad (1.17f)$$

Note that the derivative time (rate time) for an ISA Standard form must be less than the integral time (reset time), or instability will result from a reversal in controller gain. The Series form inherently prevented the effective rate time from becoming larger than the reset time.

The ultimate gain and period can be approximated as follows from the gain and period of quarter amplitude oscillations for industrial loops with a dead band or a resolution limit:

$$K_u = 1.5 * K_q \quad (1.17g)$$

$$T_u = 0.7 * T_q \quad (1.17h)$$

If the dead band or resolution limit in the valve or variable-speed drive and measurement is negligible, Equation 1.17h factor approaches one (quarter amplitude oscillation period approaches ultimate period).

Short Cut Method Tuning for Self-Regulating Processes

The ultimate period can be estimated from the primary time constant and total loop dead time based on Bode plot results. The PID gain is limited to being greater than 20% of the inverse of the open loop gain to prevent too small of a gain for dead time dominant loops. The settings computed here provide the most aggressive response to a load upset. In practice, the gain is reduced to provide a smoother and more robust response.

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$T_u = 2 * \left[1 + \left(\frac{\tau_o}{\tau_o + \theta_o} \right)^{0.65} \right] * \theta_o \quad (1.18a)$$

For an ISA Standard Form proportional-integral (PI) controller:

$$K_c = \text{Max} \left(0.2 / K_o, 0.6 * \frac{\tau_o}{K_o * \theta_o} \right) \quad (1.18b)$$

$$T_i = 0.8 * \frac{T_u}{\text{Min}(4, 10 * (\frac{4 * \theta_o}{T_u} - 1)^2 + 1)} \quad (1.18c)$$

For an ISA Standard Form proportional-integral-derivative (PID) controller:

$$K_c = \text{Max} \left(0.2 / K_o, 0.8 * \frac{\tau_o}{K_o * \theta_o} \right) \quad (1.18d)$$

$$T_i = 0.6 * \frac{T_u}{\text{Min}(4, 10 * (\frac{4 * \theta_o}{T_u} - 1)^2 + 1)} \quad (1.18e)$$

$$T_d = \text{Min} [0.25 * T_i, 0.8 * \text{Max} [0.0, 0.25 * (T_i - 0.5 * \theta_o) + \tau_s]] \quad (1.18f)$$

Short Cut Method Tuning for Integrating Processes

The ultimate period can be estimated from the secondary time constant and total loop dead time based on Nyquist plot results. The PID gain is limited to being less than the valve stick-slip (S_v) divided by the difference between the measurement noise (N_m) and the measurement sensitivity limit (S_m) to prevent fluctuations in the PID output from exceeding the stick-slip causing excessive packing wear and high-frequency disturbances. The divisor is limited to a 16-bit analog/digital (A/D) convertor resolution of 0.003%. In older DCSs with a 12-bit A/D, a resolution of 0.05% should be used in the divisor.

$$T_u = 4 * \left[1 + \left(\frac{\tau_s}{\theta_o} \right)^{0.65} \right] * \theta_o \quad (1.19a)$$

For an ISA Standard Form proportional-integral (PI) controller:

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$K_c = \text{Min} \left(\frac{S_v}{\max[(N_m - S_m), 0.003]}, 0.6 * \frac{1}{K_i * \theta_o} \right) \quad (1.19b)$$

$$T_i = 0.8 * T_u \quad (1.19c)$$

For an ISA Standard Form proportional-integral-derivative (PID) controller:

$$K_c = \text{Min} \left(\frac{S_v}{\max[(N_m - S_m), 0.003]}, 0.8 * \frac{1}{K_i * \theta_o} \right) \quad (1.19d)$$

$$T_i = 0.6 * T_u \quad (1.19e)$$

$$T_d = \text{Min} \left[0.25 * T_i, 0.8 * \text{Max} \left[0.0, 0.25 * (T_i - 0.5 * \theta_o) + \tau_s \right] \right] \quad (1.19f)$$

Short Cut Method Tuning for Runaway Processes

The ultimate period can be estimated from the positive feedback time constant, secondary time constant and total loop dead time based on Nyquist plot results. The PID gain is limited to being greater than twice the inverse of the open-loop runaway process gain (K_p') to prevent the process from going unstable due to insufficient feedback action. PI tuning is not offered since the omission of derivative action is not advisable.

$$T_u = 4 * \left[1 + \left(\frac{N}{D} \right)^{0.65} \right] * \theta_o \quad (1.20a)$$

$$N = (\tau_p' + \tau_s) * (\tau_p' * \tau_s) \quad (1.20b)$$

$$D = (\tau_p' - \tau_s) * (\tau_p' - \theta_o) * \theta_o \quad (1.20c)$$

For an ISA Standard Form proportional-integral-derivative (PID) controller:

$$K_c = \text{Max} \left(2.0 / K_o, 0.8 * \frac{\tau_p'}{K_p' * \theta_o} \right) \quad (1.20d)$$

$$T_i = 0.6 * T_u \quad (1.20e)$$

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$T_d = \text{Min} \left[0.25 * T_i, 0.8 * \text{Max} \left[0.0, 0.25 * (T_i - 0.5 * \theta_o) + \tau_s \right] \right] \quad (1.20f)$$

Maximizing Absorption of Variability Tuning for Surge Tank Level

When the absorption of variability must be maximized, lambda is chosen to be as large as possible. The most common occurrence is a level control application where the transfer is minimized of flow changes coming into the volume to flow changes going out of the volume. The flow changes coming in are absorbed as much as possible by allowing the level to change within operating limits, such as low and high alarm points. The lambda integrating tuning method is used, and the arrest time is as large as possible without causing violation of a level limit. The maximum arrest time lambda depends upon the integrating process gain, the allowable change in the process variable and the available change in the manipulated flow. The following equations show how to calculate lambda for a generic application and then a surge tank level loop.

The maximum arrest time lambda (λ) is the maximum allowable % excursion ($\Delta\%PV_{\max}$) divided by the maximum possible PV ramp rate. The maximum possible ramp rate is the PV rate of change per percent output change (open-loop integrating process gain) multiplied by the maximum available percent output change ($\Delta\%CO_{\max}$).

$$\lambda = \frac{|\Delta\%PV_{\max}|}{\left| \frac{\frac{\Delta\%PV}{\Delta t}}{\Delta\%CO} \right| * |\Delta\%CO_{\max}|} \quad (1.21a)$$

Realizing that the integrating process gain is the PV ramp rate per percent output change:

$$\lambda = \frac{1}{K_i} * \frac{|\Delta\%PV_{\max}|}{|\Delta\%CO_{\max}|} \quad (1.21b)$$

An equivalent setpoint rate limit on the controller output (e.g. flow controller setpoint):

$$\left| \Delta\%CO / \Delta t \right|_{\max} = \frac{|\Delta\%CO_{\max}|}{\lambda} = K_i * \frac{|\Delta\%CO_{\max}|^2}{|\Delta\%PV_{\max}|} \quad (1.21c)$$

For a PV limit ($\%PV_{\text{Limit}}$) and corresponding CO limit ($\%CO_{\text{Limit}}$) we have:

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$\lambda = \frac{1}{K_i} * \left| \frac{\%PV_{Limit} - \%SP}{\%CO_{Limit} - \%CO} \right| \quad (1.21d)$$

The above calculation would be done for high and low operating limits and various setpoints. The smallest of the arrest times would be used in tuning.

We can obtain the more detailed requirements for surge level tank level control by computing the integrating process gain for level. The integrating process gain is the product of the valve, process and measurement gains:

$$K_i = K_v * K_p * K_m \quad (1.22a)$$

The valve gain or variable speed drive gain for a linear installed characteristic or flow loop is:

$$K_v = \frac{\Delta F_v}{\Delta \%CO} = \frac{\Delta F_{max}}{100\%} \quad (1.22b)$$

The level process gain for mass flow is (omit density term for volumetric flow):

$$K_p = \frac{1}{A * \rho} \quad (1.22c)$$

The level measurement gain is:

$$K_m = \frac{\Delta \%PV}{\Delta L} = \frac{100\%}{\Delta L_{max}} \quad (1.22d)$$

Substituting in the valve, process and measurement gains, the integrating process gain is:

$$K_i = K_v * K_p * K_m = \frac{\Delta F_{max}}{100\%} * \frac{1}{A * \rho} * \frac{100\%}{\Delta L_{max}} = \frac{\Delta F_{max}}{\Delta L_{max}} * \frac{1}{A * \rho} \quad (1.22e)$$

The consequential arrest time for a level loop is:

$$\lambda = \frac{A * \rho * \Delta L_{max}}{\Delta F_{max}} * \frac{\Delta \%PV_{max}}{\Delta \%CO_{max}} \quad (1.22f)$$

An equivalent setpoint rate limit on the controller output (e.g., flow controller setpoint):

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$\left| \Delta\% CO / \Delta t \right|_{\max} = \frac{|\Delta\% CO_{\max}|}{\lambda} = \frac{\Delta F_{\max}}{A * \rho * \Delta L_{\max}} * \frac{|\Delta\% CO_{\max}|^2}{|\Delta\% PV_{\max}|} \quad (1.22g)$$

The computation of the arrest time can be significantly simplified if the dead time is assumed to be negligible. This is a reasonable assumption for surge tank level control because the total loop dead time is much smaller than the arrest time in Equations 1.7b and 1.7c for lambda tuning of integrating processes.

$$K_c = \frac{T_i}{K_i * (\lambda)^2} \quad (1.22h)$$

$$T_i = 2 * \lambda \quad (1.22i)$$

If we substitute Equation 1.22i into 1.22h and cancel out lambda in the numerator by one of the lambdas in the denominator, we have a simpler Equation 1.22j for the PID gain:

$$K_c = \frac{2}{K_i * \lambda} \quad (1.22j)$$

$$K_c = \frac{2}{\frac{\Delta F_{\max}}{\Delta L_{\max}} * \frac{1}{A * \rho} * \frac{A * \rho * \Delta L_{\max}}{\Delta F_{\max}} * \frac{\Delta\% PV_{\max}}{\Delta\% CO_{\max}}} = 2 * \frac{\Delta\% CO_{\max}}{\Delta\% PV_{\max}} \quad (1.22k)$$

The first expression in Equation 1.22f can be simplified to Equation 1.22l if we define a full scale residence time (t_{fsr}) as the maximum volume ($A * \rho * \Delta L_{\max}$) for the maximum span of the level measurement divided by the maximum flow capacity of the valve or VSD (ΔF_{\max}). This full-scale residence time term is not to be confused with the actual residence time (operating volume divided by operating flow rate) used in process calculations in chemical engineering and throughout the rest of this book that is particularly important for computing back mixed time constants or plug flow transportation delays and the time for reaction conversion.

$$t_{fsr} = \frac{A * \rho * \Delta L_{\max}}{\Delta F_{\max}} \quad (1.22l)$$

We get Equation 1.22m using the full scale residence time (t_{fsr}) by substitution per Equation 1.22l into the first expression in Equation 1.22f for lambda:

So Many Tuning Rules, So Little Time

Gregory K McMillan

$$\lambda = t_{fsr} * \frac{\Delta\% PV_{\max}}{\Delta\% CO_{\max}} \quad (1.22m)$$

We have the simple Equation 1.22n from the result of Equation 1.22k for the PID gain:

$$K_c = 2 * \frac{\Delta\% CO_{\max}}{\Delta\% PV_{\max}} \quad (1.22n)$$

We can solve Equation 1.22n for the maximum desired change in the process variable (level) divided by the maximum desired change in manipulated flow:

$$\frac{\Delta\% PV_{\max}}{\Delta\% CO_{\max}} = \frac{2}{K_c} \quad (1.22o)$$

If we substitute Equation 1.22o into 1.22n we end up with the simple Equation 1.22p for the reset time:

$$T_i = 2 * t_{fsr} * \frac{\Delta\% PV_{\max}}{\Delta\% CO_{\max}} = 2 * t_{fsr} * \frac{2}{K_c} = t_{fsr} * \frac{4}{K_c} \quad (1.22p)$$