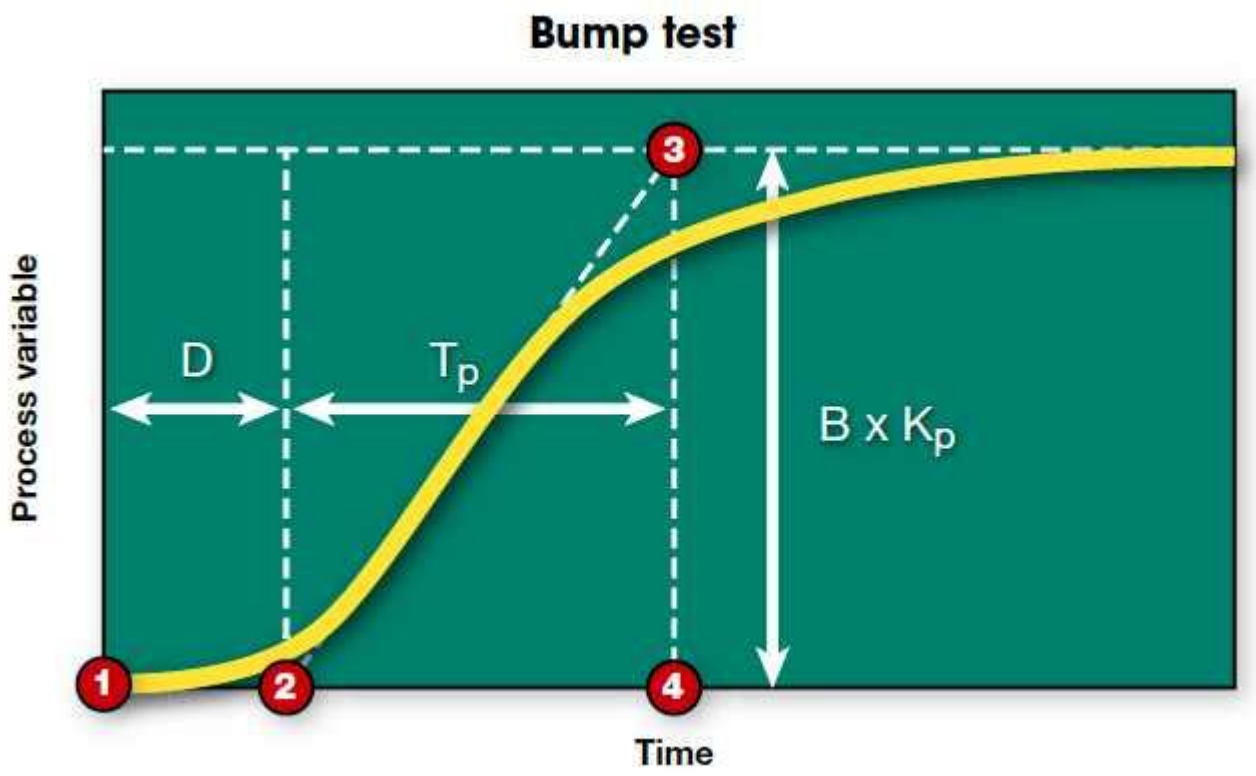
Fundamentals of lambda tuning

Understanding a particularly conservative PID controller design technique.

Vance VanDoren, PhD, PE
04/16/2013

Lambda tuning is a form of internal model control (IMC) that endows a proportional-integral (PI) controller with the ability to generate smooth, non-oscillatory control efforts when responding to changes in the setpoint. Its name derives from the Greek letter lambda (λ), which designates a user-specified performance parameter that dictates how long the controller is allowed to spend on the task of moving the process variable from point A to point B.

Like its more famous cousin, Ziegler-Nichols tuning, lambda tuning involves a set of formulas or tuning rules that dictate the values of the PI parameters required to achieve the desired controller performance. The first step in applying them is to determine how much and how fast the process responds to the controller's efforts (see the bump test graphic).



Bump test

This 9-step test, also known as an open-loop reaction curve test or step test, gives a PI controller everything it needs to know about the behavior of a non-oscillatory process in order to control it:

1. Turn off the controller by switching it to manual mode.
2. Wait until the process variable settles out to a steady-state value.
3. Manually "bump" or "step" the process by forcing the control effort abruptly upwards by B%—whatever it takes to make the process variable move appreciably but not excessively.
4. Record the process variable's reaction or step response on a trend chart as above, starting at the time when the bump was applied (step 1) and ending when the process variable settles out again.
5. Draw an ascending line tangent to the steepest part of the process variable's trend line.
6. Draw horizontal lines through the process variable's initial and final values.
7. Mark where the two horizontal lines intersect the ascending line at points 2 and 3.
8. Record the deadtime D from point 1 to point 2 and the process time constant T_p from point 2 to point 4.
9. Record the change in the process variable from point 3 to point 4 then divide that by B to get the process gain K_p.

Once the process behavior has been characterized in terms of the process parameters—the process gain Kp, the process time constant Tp, and the deadtime D—tuning the controller is simple. Just plug those values and the user's choice of λ into the formulas shown in the "lambda tuning rules" sidebar to get the required values for the PI parameters Kc and Ti.

Consider a process with an open-loop gain Kp, a time constant Tp, and a deadtime D being driven by the control effort or *control output* CO(t) from a PI controller given by

$$CO(t) = K_c\left[e(t) + \frac{1}{T_i}\int e(t)dt\right]$$

where

$$e(t) = SP(t) - PV(t)$$

is the error at time t between the process variable PV(t) and the setpoint SP(t). The rules for lambda tuning call for

$$K_c = \frac{T_p}{K_p(\lambda + D)}$$

and

$$T_i = T_p$$

in order to obtain a closed-loop system with a non-oscillatory setpoint response that will settle out in approximately 4λ seconds.

Note that these tuning rules require the user to specify only one performance parameter: λ. This not only simplifies the calculation of Kc and Ti, but it also allows the user to select the controller's desired performance in terms of a physically meaningful quantity—the time allowed to complete a setpoint change—as opposed to the less intuitive concepts of proportional band and reset time.

Closed-loop performance

A PI controller thus tuned will, theoretically, complete a setpoint change in about 4λ seconds when operating in closed-loop mode, and it will do so without overshoot. That is, it will drive the process variable towards the setpoint gradually enough to guarantee that the error between them will continue to diminish steadily.

This overdamping feature can be especially useful in applications where the process variable must be maintained near some limiting value that the process variable must not cross. The controller will never accidentally violate such a constraint because it will never drive the process variable past the setpoint. Nor will a lambda-tuned controller ever cause unstable oscillations in the process variable because it will never need to reverse course after a setpoint change. The process variable will always proceed steadily upward or steadily downward until the new setpoint is reached.

Overdamping also helps ensure consistency, which is why lambda tuning has become particularly popular for paper-making operations where fluctuations in certain process variables can cause visible irregularities in the finished product. The absence of overshoot also prevents the interacting loops in a paper-making machine from shaking the equipment to death by causing the

actuators to oscillate all at once. And individual actuators (especially valves) will be subject to less wear and tear since they will never be required to reverse course unless the setpoint does.

Coordinating multiple loops

Furthermore, since lambda tuning allows a PI controller to achieve its objective over a user-specified interval, it can be used to synchronize all of the controllers in a multi-loop operation so that the process variables will all move at roughly the same rate. This too contributes to uniformity in the paper-making process. It also helps maintain a constant ratio of ingredients in a blending operation.

Conversely, when certain interacting loops are more important than others, the most critical ones can be assigned smaller λ values to make sure that they remain out of spec for the shortest possible interval following a setpoint change. Loops that contribute less to the overall profitability of the operation and loops that have slower or less powerful actuators can be allowed to take their time with larger λ values.

Using highly disparate lambda values for two interacting loops can also help decouple them. The faster loop will see little or no effect from the slower one since the latter will appear to be more-or-less stationary during the interval that the former is completing its latest setpoint change. Conversely, the faster loop will have finished its work by the time the slower one gets underway. The decoupling won't be complete, but the interactions between the loops will be mitigated at least somewhat, thereby reducing the apparent loads that each would otherwise cause the other.

A less obvious advantage of lambda tuning is its robustness. Because a lambda-tuned controller is so conservative, it can withstand considerable discrepancies between the estimated and actual values of the process parameters, whether those discrepancies are due to a poorly executed bump test or a change in the process that occurs after the tuning is complete. The resulting distortions in the calculated PI parameters may well make the controller more or less conservative than if the tuning had been accomplished with perfect knowledge of the process's behavior, but the closed-loop system is likely to remain overdamped either way.

Disadvantages

On the other hand, lambda tuning has its limits, especially when speed is of the essence. It tends to make a slow process even slower, causing the process variable to remain out of spec for a long time. Specifically, λ is generally assigned a value between Tp and 3Tp, making the closed-loop response to a setpoint change up to three times longer than the corresponding open-loop step response. An even larger value of λ is required if the deadtime D is significant. In such cases, λ > D is the practical lower limit since the controller can't be expected to react any faster than the deadtime allows.

But arguably the most challenging drawback to a lambda-tuned controller is its limited ability to deal with an external load on the process. It can still bring the process variable back to the setpoint if a random load should ever upset the process variable, but it will make no effort to do so particularly quickly or efficiently. Even measuring the disturbances won't help because the lambda tuning rules make no provisions for the behavior of the load, only the process.

The best the user can do is to set λ as small as possible in order to increase the controller's speed overall, but doing so will tend to make the controller less robust. And lambda tuning wouldn't be a particularly good choice anyway when a fast response is required since there are other tuning rules that are much more effective for time-sensitive applications.

Mathematical challenges

There are also some subtle limitations to lambda tuning buried deep in the underlying mathematics. For one, it can't be applied to a process that is itself oscillatory. If an open-loop bump test yields a step response that fluctuates before settling out, the process cannot be completely characterized by just the three parameters Kp, Tp, and D and the controller cannot be tuned with the lambda rules, though there are several related IMC techniques that will work just fine in such cases.

The mathematics also break down when the deadtime D is especially large. The calculations required to compute Kc and Ti suffer from an approximation that becomes less and less accurate as D increases. Several alternative approaches have been proposed to improve the accuracy of that approximation, but those efforts have also generated considerable confusion—multiple sets of tuning rules all called "lambda tuning." They all achieve roughly the same closed-loop performance but look nothing alike. Some apply to a PI-only controller while others require a full PID controller equipped with derivative action.

Lambda tuning rules also take on different forms for an integrating process that has no time constant. These occur in applications such as level control where a bump from the controller (opening the inlet valve) results in a process variable (liquid level) that continues to rise without leveling off. A lambda-tuned controller can force an integrating process to reach a steady-state, but it takes longer for the process variable to settle out—about 6 λ seconds—and the process variable will overshoot the setpoint along the way.

Nonetheless, lambda tuning is relatively simple, intuitive, and bullet-proof. It will no doubt remain popular in applications where a conservative controller is required.

Vance VanDoren, PhD, PE, is Control Engineering *contributing content specialist. Reach him at controleng(at)msn.com.*

Key concepts

- When applied properly, lambda tuning can move a process to a new setpoint in a specified amount of time and without overshoot.
- This approach is particularly valuable when there are critical limits that a process should not cross.
- It isn't appropriate for every application, especially those that need quick response.

See more articles on process control strategy below:

Source:

http://www.controleng.com/single-article/fundamentals-of-lambda-tuning/010e177f99bb831fc647e8b8975a073a.html