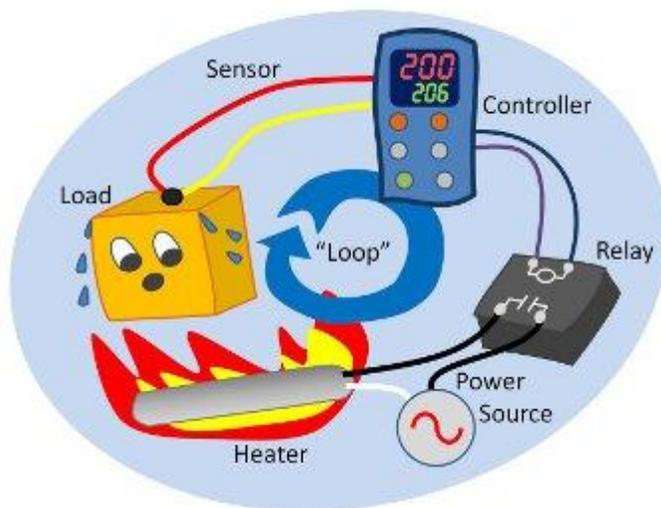


# Tuning thermal PID loops

When dealing with critical temperature applications, PID controllers are a common regulatory approach, but tuning these often requires a different strategy than other types of loops. Auto-tuning features can help if you understand them.

Jason Beyer and Sean Wilkinson, Watlow  
01/17/2012



If you are responsible for working with PID temperature controllers, you have probably already discovered that such loops can be challenging and that the needs of a given controller and application can vary widely. This discussion is intended to help explain how these controllers work and to offer some basic guidance on dealing with a PID temperature controller. We've done our best to avoid unnecessary jargon while providing basic terminology and definitions that will be helpful when referencing controller manuals and other sources. Bear in mind, though, that controllers vary and applications vary, and we would be remiss if we lead you to believe that our experiences cover all cases or that our advice never goes wrong. Please be careful to consider the unique circumstances of your application as you implement any changes.

## **Why tune controllers?**

For optimal results, a PID controller needs to know how much to adjust the heat to achieve a desired temperature change, and how long the temperature takes to react to a change in heater power. Tuning teaches the controller the characteristics of a particular system. The controller captures what it learns in its PID settings. The exact names of the PID settings depend on the controller manufacturer, but they typically are: proportional band or gain, integral or reset, and derivative or rate. Consult the manual to find the PID settings in your controller. The controller cannot know the best values for these parameters until it is tuned because every system is different.

When poorly tuned, the temperature can oscillate around the setpoint, be slow to respond to changes, or overshoot the setpoint excessively at start-up or the when the setpoint changes. This impacts productivity by making operators wait, reduces yield, and increases premature failures when products are processed at the wrong temperature.

## **How do you tune a PID controller?**

The simplest way to tune a PID controller is to use its auto-tune feature. Nearly all electronic temperature controllers now have one, but they don't all work the same way. To find out how to best use your controller's auto-tune, read its manual or call its manufacturer. Some controllers tune while the load heats up from ambient. Some tune around setpoint. Either way, auto tuning adjusts the PID settings automatically, so you don't have to. But before you engage that feature, consider these options and implications:

- The temperature may overshoot setpoint while tuning. Controllers that tune near setpoint force the temperature to go up and down. To limit the temperature, set a lower setpoint and observe the tuning behavior. You can tune again at a higher setpoint after confirming that tuning won't cause the temperature to go too high.
- There is probably a time limit on the auto tune function, so very slow processes may not tune. Check the PID settings prior to and after tuning. If they do not change, the auto tuning process failed for one reason or another. That's a good time to get help from the controller's manufacturer.
- Defining what is considered good tuning depends on the process. Some controllers have options that customize auto tuning results for your

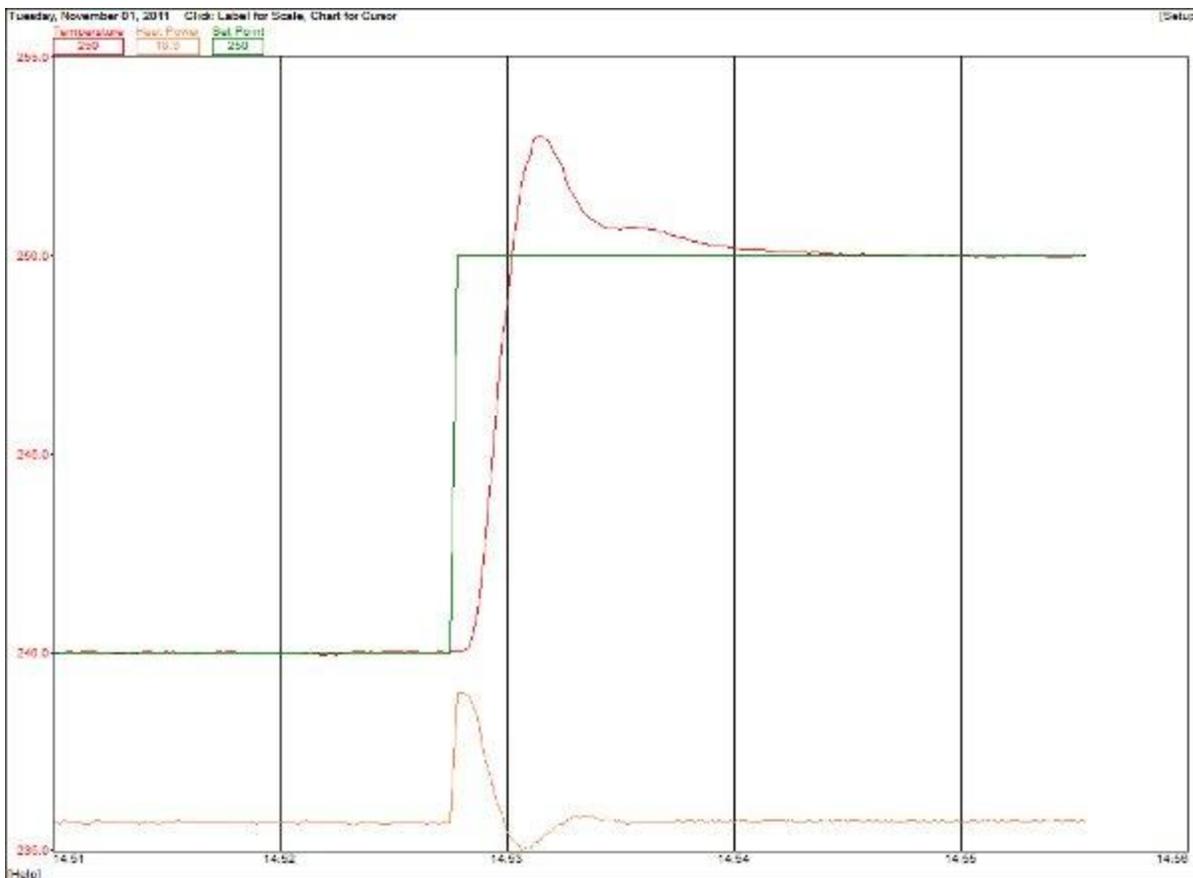
process. For example, certain [Watlow](#) controllers allow you to select whether the temperature should get to setpoint in the minimum amount of time with a bit of overshoot, or approach setpoint more cautiously to minimize or eliminate overshoot.

To get the best results when tuning, make sure conditions are like those at which the system will normally function. Here are our tips for a successful auto-tune implementation:

1. Set the setpoint before starting the auto-tune process.
2. Make sure the system's temperature is stable before starting.
3. Tune the system at the time and location at which it will be used. Tuning in a lab on a summer day in California may not yield the results necessary to control well on a winter night in Minnesota.
4. Tune with the same heater voltage to be used in operation. If the heaters use 240 Vac when tuning but 208 Vac when installed at the user's site, the controller will likely have to be tuned again since the change in power will change the way the heaters perform.
5. Tune with actual product or a reasonable simulation in place. An oven full of metal parts tunes differently than an empty one.
6. Tune fully assembled and installed systems. A machine with its cover panels off can perform differently when covered.
7. Consider all the sources of heat. A batch of powered circuit cards in a test oven can significantly change the way it tunes.
8. Consider all the heat sinks. Imagine installing the first machine in a line where several will share the exhaust duct. If the vents that will connect to the other machines are closed when you tune the first machine, the cooling effects of the exhaust may be much greater than after the other machines are installed and operating.
9. Consider the range of temperatures at which you want the system to perform well, and tune at the highest, lowest, and midpoint; or at each operating temperatures if there are not too many. Make a record of the PID settings resulting from each trial; controllers typically overwrite the previous settings each time you tune. If the trials all control well, use the

widest proportional band (lowest gain), the least active integral (lowest repeats per minute or highest minutes per repeat) and the least active derivative (typically the smallest number).

10. When multiple temperatures are controlled and the heat from one can affect another, for controllers that tune at setpoint, tune the loops one at a time with the other loops stable at setpoint. For a controller that tunes while heating from ambient, it may be best to tune the loops simultaneously.
11. If product and heat flows from one temperature control zone to another in a conveyor oven, for example, tune the loops in that order.

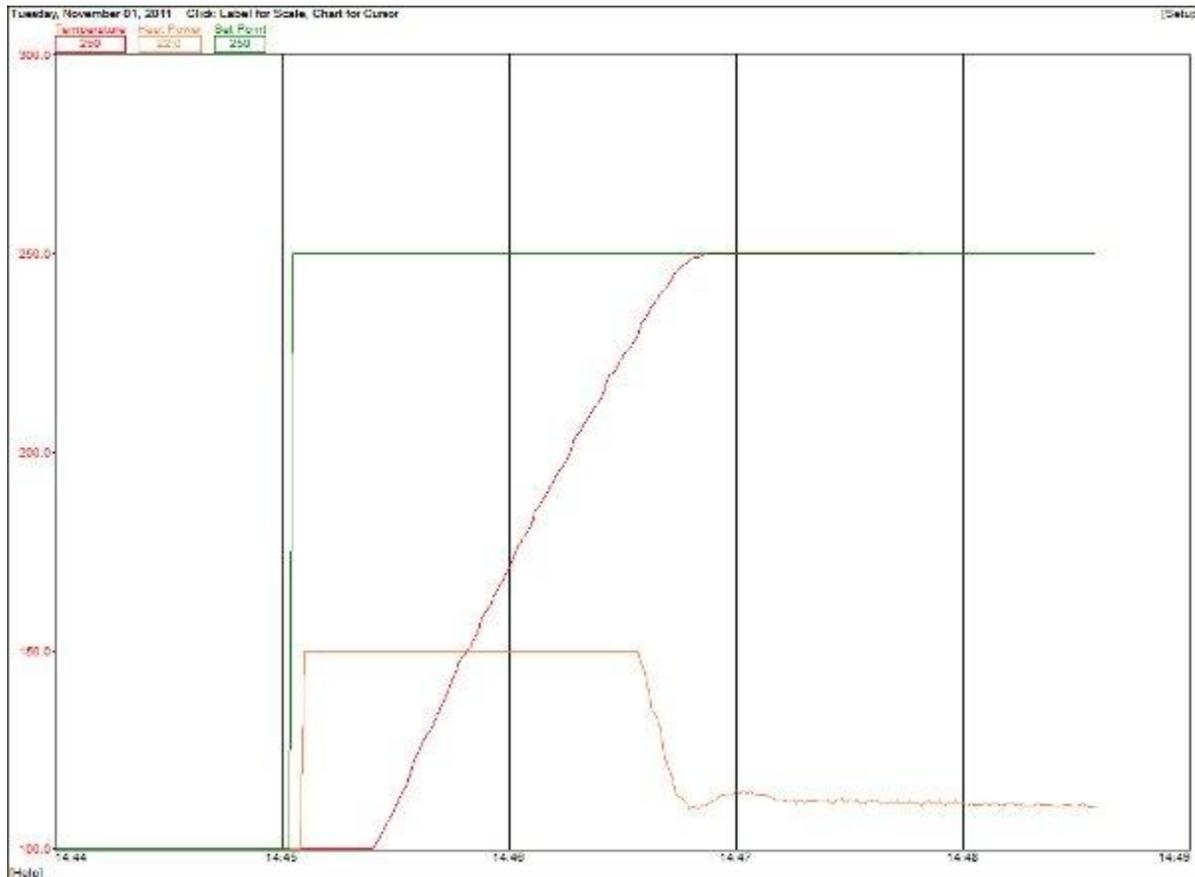


### When is it well tuned?

The system is tuned well when it heats up and settles quickly at setpoint and when the temperature settles at a new setpoint without oscillating excessively. Of course, quickly and excessively are relative terms, and as noted above, some processes tolerate a little overshoot, allowing the system to change temperature in the minimum time, and others do not. In a system that

tolerates some overshoot, we look for responses like those shown in the graphs. For us, the key indicator of a well-tuned system is not just that the temperature is stable, but that the output power is also stable—it should not hunt or oscillate more than a few percent.

Use software such as SpecView to graph the temperature, setpoint, and percent heat power. With a graph you can quantify performance by measuring time to setpoint, time to stabilize, and oscillation amplitude, if any. This allows you to measure if the tuning meets your needs.



### If auto tuning doesn't work

If the temperature does not perform to your satisfaction, consider these possibilities:

- Was auto tuning performed under ideal conditions? Review our tips above. If something was not right, correct it and try auto-tuning again.

- When the temperature is at setpoint, if the heat power is not between 10% and 90%, look for a problem such as missing covers or a failed heater. Otherwise, there may be issues with the design or installation.
- If the heat is at 100% and the temperature doesn't increase or never reaches setpoint, shut the heat off and check that the sensor is positioned and connected correctly. Otherwise, try to determine why there is not enough heating power or why there is too much cooling. Tuning is probably not the problem.
- If the temperature is oscillating, is it due to the power switching method? If the frequency of the oscillation is the same as the time proportioning cycle time, reduce the cycle time setting if your relay allows it, or replace the relay with a solid-state power controller that allows much faster switching.
- If the performance is poor because the operating conditions change too dramatically for PID control to adapt, consider using adaptive tuning if your controller offers it.

## **Switching to manual**

If you still need to make improvements, you can manually adjust the tuning. Providing detailed instructions on manually tuning PID control is beyond our scope here, but consider the following:

- If the temperature does not reach setpoint fast enough, you may be able to fix that, but you may have to tolerate a little overshoot and settling time.
- If it doesn't settle down fast enough, you may be able to fix that, but you may have to tolerate slower responses to setpoint changes.
- Adjust only one PID setting at a time.
- Make sure you know which way to adjust each parameter for the desired result.
- Double or halve the PID setting when making adjustments. With most controllers, small changes will have negligible effects.
- Change the setpoint to test the system's responsiveness.

- Wait long enough to see the results of each change before making another. How long to wait depends on how quickly the system can respond. Wait three or four cycles if it is oscillating.
- Graph the results each time you make a change and record the PID settings on the graph. This allows you to evaluate whether or not your changes are improvements.
- Graph the output power. If output power is oscillating, even if the temperature is stable, the system is probably not stable. Output power is as close to a crystal ball as you get, it tells you what the control system is trying to do before the heater makes it happen and before the system filters the results to the sensor.

We hope these suggestions will help improve the performance of your controllers. More extensive discussions of PID tuning strategy are available at [Watlow's website](#).

*Beyer is a technical support specialist for controllers and power switching devices with Watlow, where he has worked for 32 years. Wilkinson is a product manager for multi-loop controllers and software with Watlow, where he has worked for 15 years.*

Source:

<http://www.controleng.com/single-article/tuning-thermal-pid-loops/aebf3cf5326fbf2db40450e202903301.html>