APPENDIX F. PID temperature control algorithm

When you keep the ratio between heating power and work mass the same you can use the same settings for most process conditions. Beware that a heat/cool process can behave differently for heating and/or cooling.

PID control can be considered as the intelligence or "brain" of the controller. The same is when you are driving a car on the highway and wish to control the speed at 100 km/hr. Because of varying road conditions (different tarmac, driving up- or downhill) it is necessary to increase and decrease throttle to hold the desired speed. You will also notice that your foot automatically does this.
Try to concentrate more on what you are actually doing, and you will notice that it is actually quite
impressive on how the brain executes this apparently easy task. Your brain is now acting as some kind of very high level adaptive PID controller.

F.1 A bit of history

In the past century, when automation became more and more important in industry, mathematicians and engineers tried to find a way to integrate this knowledge into an automated system.

PID controllers date back to the 1890s governor design. PID controllers were subsequently developed in automatic ship steering. One of the earliest examples of a PID-type controller was developed by Elmer Sperry in 1911, while the first published theoretical analysis of a PID controller was by Russian American engineer Nicolas Minorsky in 1922. Minorsky was designing automatic steering systems for the US Navy, and based his analysis on observations of a helmsman.
Observing that the helmsman controlled the ship not only based on the current error, but also on past error and current rate of change. This was then made mathematical by Minorsky. His goal was stability, not general control, which significantly simplified the problem. While proportional control provides stability against small disturbances, it was insufficient for dealing with a steady disturbance, notably a stiff gale (due to droop), which required adding the integral term. Finally, the derivative term was added to improve control.

Trials were carried out on the USS New Mexico, with the controller controlling the angular velocity (not angle) of the rudder. PI control yielded sustained yaw (angular error) of ±2°, while adding D yielded yaw of ±1/6°, better than most helmsmen could achieve.

The Navy ultimately did not adopt the system, due to resistance by personnel. Similar work was carried out and published by several others in the 1930s.

In the early history of automatic process control the PID controller was implemented as a mechanical device. These mechanical controllers used a lever, spring and a mass and were often energized by compressed air. These pneumatic controllers were once the industry standard. Later on the first electronic analog PID controllers were made with tube-amplifiers.

After the invention of the solid-state transistor things were progressing into analog solid state PID controllers and later, with digital IC technology, the first truly digital controllers were invented that make very advanced adaptable PID controllers possible under quickly varying process conditions.

F.2 From theory to practice

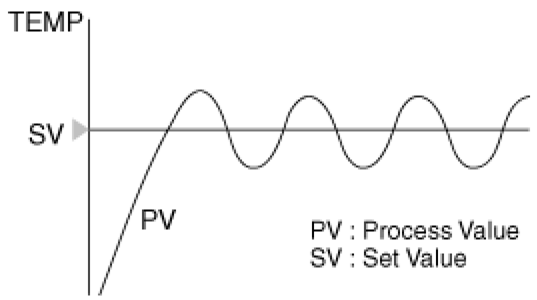
In order to fully understand a PID controller, an extensive and thorough mathematical knowledge of Laplace and Z-domain transformations is necessary. That goes too far for this manual.

We therefore take a more practical approach and try to describe a PID controller in common language that almost everybody can understand.

To make things easier we will start out with a very simple on/off controller, think of the old thermostat that was used in the past to control the temperature in a room (the modern types used nowadays are almost all PID controlled).

F.2.1 ON/OFF control action

ON/OFF temperature control is the simplest and least expensive form of control available. The output signal from a controller is either FULLY ON or FULLY OFF depending on the direction of the deviation between the set value and process value.
The figure below shows the characteristics of an ON/OFF control action:



The ON/OFF control action takes place if any deviation from the set value occurs.

This action responds quickly, but is sensitive to input noise which causes chattering (ON/OFF switching at short intervals). Therefore, in actual use, the ON/OFF temperature control action has some hysteresis which is named dead band or control sensitivity. This prevents the quick chattering around the set value when the process value is very near.

F.2.2. Proportional action

P stands for proportional action and is a multiplication factor of the deviation between actual set value and process value. Proportional action control is also referred to as P or gain in some control systems.

With the proportional action, the controlled process no longer switches as a direct result of the deviation between set value and process value but controls this *proportional* to this deviation.

The proportional action control is active within a user-definable zone around the set value, called the proportional band (Pb). When the process value (PV) enters the proportional band, the output becomes gradually smaller and the process value stabilizes somewhere within the proportional band.

In the Heat Manager you set the proportional band in *degrees*, but beware that there are other brands of systems that set this value in a percentage of the measurement span.

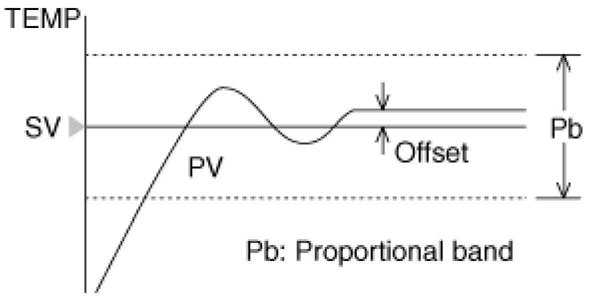
Proper adjustment of the proportional band will result in smooth control. However, it is very seldom, that the actual process value stabilizes exactly on the set value, and it usually becomes stable with some deviation from the set value, called offset.

A high P value gives the control system a large band of modulation, the major disadvantage is that the reaction to the deviation can be quite slow.

A too low P value, however gives an oscillating system where the process value continuously over- and undershoots the set value. We call that “hunting”, the controller hunts to reach the set value,

Visualize this by thinking of the gas pedal in a car (with automatic transmission): a high P value is a gas pedal where you can push it very far in so you can control in an accurate way, but need to push it far in and out to reach the speed that you want.
A low P value however is a gas pedal that has a very small travel, but controls the speed of the motor from idle to maximum rev with only a small push on the pedal. It is clear that this situation makes it also difficult to reach the speed you want: you over- and undershoot the speed all the time: more like a classic on/off control.

The “trick” is to reach the correct value where the oscillation is very low, but with a quick enough responsiveness to deviations. Large slow systems that react slowly to deviations generally need a higher P action than small quick systems.



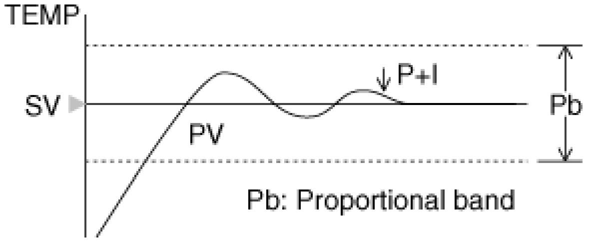
Note: Setting the P action to 0 will result in on/off control action. This will result in relatively large over- and undershoots.

F.2.3 Integral action

I stands for integral action. It takes a summation of the deviations between set value and process value from the past into account. This action is necessary for removing the small offset error that remains
between set value and process value when only a P action would be used.

The integral control action is also referred to as reset. The degree of integral action is expressed as an integral time in seconds. The purpose of the integral action is to automatically compensate for any steady state offset that is inherent with a controller that only works with a proportional action.

The integral action moves or resets the proportional band up or down depending on the offset. The integral time of the controller is adjustable and determines how fast the proportional band is moved.



A too high setting of the integral action will result in a very slowly reacting process with big over- and undershoots.

A too low setting of the integral action will result in a control system that can never reach the set value exactly. There will remain a small offset error and the desired set value is not reached.

The trick here is also to reach the setting that more or less corresponds to the actual inertia of the system under control.

Now we go back to the speed control of our car with the gas pedal. When you push in the gas pedal fully it takes some time to rev up the engine and speed up. Also when you let go of the pedal it takes some time to let the engine rev down. This is mainly because of mechanical inertia inside the engine. The weight of the car itself is also very important: a big heavy truck takes much more time to speed up then a small compact car.

In a laboratory, on rollers, we determine the speed of the car as a function of the gas pedal position: this can now be considered as the P-action.

This situation does not correspond to a real road situation where you have wind and different kinds of tarmac.

To compensate for this the gas pedal needs to be pushed in a bit more (or less) depending on the situation. This last action can be considered as the integral action: it fine tunes the actual gas pedal setting, but in a very slow way, otherwise we overshoot the desired speed.

F.2.4 Differential action

The D stands for differential (or sometimes called: derivative) action. This takes the rate of change per time unit of a deviation between set value and process value into account.

Differential action temperature control is also referred to as rate. The degree of differential action is expressed by the differential time in seconds.

The controller measures the rate of the temperature increase per time unit and moves the proportional band to minimize overshoot. The output change is directly proportional to the rate of change in the process value (PV) per time unit.

Theoretically speaking a heating process is a pure integrator and can be controlled by using only a P and I action (most slow processes are). Practice however shows that a small amount of D action can improve accuracy and especially responsiveness to changing external circumstances just a bit further.

A D-action is often used in positioning systems (servo applications) where quick movements are performed. A D-action is also important in a tunnel furnace where the product load differs from time to time (sometimes no product feed), here the D-action is used to prevent over- or undershoot under various process load conditions that change more quickly than the integral time of the process itself.

A too high D action will result in a very unstable process with large over- and undershoots.

A too low D action will result in a system that does not respond quick enough (too late) to varying process conditions. This will also result in an unstable system with over- and undershoots.

F.2.5 Cycle time

Another important setting in a PID controller is its cycle time. This time is the time span used by the controller to calculate how long a relay should be on or off.

Here is an example: we have a mechanical relay that is driving a heating element and the cycle time is set to 15 seconds.
When the controller output is 100% the relay will be continuously activated. When the process value goes a bit over the set value the controller starts to react to this by *modulating* the controller output back to a lower value, e.g. 50%.
When the output is 50%, and the cycle time is 15 seconds, the relay will be on for 7.5 seconds, then switch off for 7.5 seconds, switch back on for 7.5 seconds, etc. etc. etc.

When the controller output is only 10% the relay will be switched on for 0.1 x 15 = 1.5 seconds and then remain off for 15 – 1.5 = 13.5 seconds.

The problem is that we introduce some kind of *lag time* into the system. Of course it would be a good idea to set this cycle time as low as possible, because then the controller can react more quickly to varying process conditions. The major disadvantage is then, that the *mechanical* relay has to switch very quickly on and off. This will damage the contacts of the mechanical relay too quickly and wear them out too fast.
As a *compromise* the cycle time has been set to 15 seconds for this purpose.

When you make use of thyristor modules or solid state relays the cycle time can be set at the lowest possible value (0.1s) since these modules don't have mechanical contacts. The controller output can directly drive the modules proportionally which will result in a quick response to varying process conditions and a more stable and accurate process control.

F.3 Work step-by-step and-by-one

When you adjust a PID controller by hand, a lot of experience and, as the Germans say: "fingerspitzen gefuehl" is necessary. We consider this manual adjustment procedure more like an art than a science.

It is very important that you have a graphic overview of the process values so you can see if your
modifications give improvements (or not, then you need to move in the other direction).

It is always best to first start by setting the I and D factors to 0 and only use the P-action. When you roughly "feel" how the process reacts to a step like change of the set value, you can try the addition of the I action to get the process value exactly on the set value.

Later on you add the D action to improve responsiveness.

After that, you start changing the P, I or D values one by one. Try doubling them, or halving them, and see how the process reacts to it. This procedure must be re-done until you are satisfied with the new settings.

When you see a process value that is slowly oscillating with a small amplitude and long period towards the desired set value you are on the right track with these settings.

We know from practice that the D action should be roughly 25% of the I action. Certainly not more.

F.4 Auto tuning

It is also possible to make use of the auto tuning feature in the Heat Manager. This is only possible when the controller is in a soak/dwell segment and the set value does not change over time.

After enabling this feature the controller will perform a few test-cycles where it automatically tries to determine the PID settings based on so-called step-wise changes of the set value.

After changing the controller output to maximum (or minimum) monitors the physical behavior of the process value and determines the settings based on oscillation amplitude and period.

After performing many tests with various control systems, Ziegler and Nichols, derived formulas on how to convert the results of these tests into PID settings as a starting-point before actual fine tuning can take place.

It sometimes takes a while before the system actually determines the new values during this auto tuning cycle. Please be patient and wait until you see the newly determined values appear. This can take some minutes and depends on how quickly the process reacts to the controller.

If the controller was not able to determine the new values it will leave the PID settings on the values before the start of the auto tuning procedure.

Watch out: these calculated PID values are only a rough indication of the actual perfect PID settings and –can- result in an unstable process, so keep in mind that manual fine tuning is often necessary.

Important: only perform auto tuning when the process reached a stable state (controller output and process value more or less stable). The set value must also be stable, so this is only possible in a soak segment and not in a ramp up/down segment.

Warning: Never execute an auto tuning procedure when the process conditions are not stable. The software algorithm *assumes* that all external factors with regard to the process are not changing while performing this operation.

Please remember, that a manually tuned controller almost always outperforms a controller that was set with auto tuning. Auto tuning is more or less a rough indication if you need some “direction” on where to start.

F.5 Write down the newly determined PID values!

It is very important, that you write down the newly determined PID values so you don’t lose these painstakingly difficult to determine values.

A software upgrade, re-init procedure or controller module re-init (necessary after a replacement) will restore the PID values back to their default values as programmed in the sotware.

F.6 Common pitfalls

A system that is oscillating around its set value, or reacting too slow to process variations, is almost always caused by a PID controller that needs fine-tuning.

Beware when it takes many minutes of time for a process value to change after step-wise change of the set value. This is caused by a so-called dead-time in the process and is very difficult to overcome with PID settings in 1 PID loop alone.

The most common solutions for these kind of process issues are:
Measure more directly the actual product process value (bring sensor nearer to product).
Make use of a so-called cascaded control system where we make use of a master-slave PID loop where the slave loop controls the heating to the process (with its separate sensor near the heater as process value input]. The set value for the slave comes from output of the master PID controller that measures the actual desired process value of the product.

When you change a P-action, you also influence the I and D action responses in some kind of way.
The same goes when you change the I or D action, everything is connected together more or less. Don't start to move down in spiraling circles and change only 1 parameter at a time and verify very well if this improves the situation or not.

A good idea is to always write down carefully what you are doing, so you begin to see a "pattern" in your workflow and take a few steps back, if desired.

Don't set the D action too high, since this very easily results in an unstable process control. Use this only for fine tuning after you set the P and I factors correctly.

When you change an action you need to wait for the process to respond to the new situation. This can take many minutes and depends on the inertia of the process. Watch the recorder output of the process value to determine if it stabilizes or not.

Generally speaking it is important to keep the ratio between heating power versus mass of the system the same, after determining the correct PID settings for your system. You can then use the same settings for different situations.

F.7 More information

Consult the internet on more information about this, there is an excellent Wikipedia page with much more information on this subject.
See: http://en.wikipedia.org/wiki/PID_controller

Parts of this description comes from the website of RKC instruments,
See: http://www.rkcinst.co.jp/english/control_01.htm