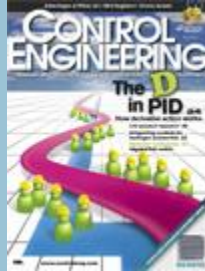


## The D in PID Control

### PID Refresher

PID (Proportional, Integral, Derivative) control is the classic method of controlling a single variable in a process with feedback only. The controller measures a process variable (which is an analog signal also known as the PV) and compares this to a setpoint (SP) value (the value set by the operator typically). The PID controller looks at the error between the setpoint and process variable and creates an output (another analog signal) that attempts to bring the error to zero. The output can be a valve to control tank jacket temperature where the PV is the jacket temperature or temperature of a fluid in the tank. A PID controller allows the engineer to adjust the gains for each of the 3 tuning parameters (P, I, and D) till the optimal level of control is achieved.



For a more complete description of PID control see the recent article in "Control Engineering" magazine at [http://www.controleng.com/article/446647-Understanding\\_Derivative\\_in\\_PID\\_Control.php](http://www.controleng.com/article/446647-Understanding_Derivative_in_PID_Control.php)

### The Need for "D"

There has been a fairly lengthy discussion going on in LinkedIn's Automation Engineers discussion group, regarding the need for D (derivative) action in a PID controller and how to adjust it correctly at [http://www.linkedin.com/groupAnswers?viewQuestionAndAnswers&discussionID=12721884&gid=75483&commentID=11482037&trk=view\\_disc](http://www.linkedin.com/groupAnswers?viewQuestionAndAnswers&discussionID=12721884&gid=75483&commentID=11482037&trk=view_disc)

Most users quickly grasp how to adjust the P & I gains. However, properly adjusting the D gain can be tricky.

D for derivative. This term produces an output that is proportional to the rate of change of the error signal. It is commonly believed that this gives the output an extra boost when upsets to the process (a rapid change in the PV) occur. This is true when the error (PV - SP) is changing rapidly. However, as the rate of change in the error signal slows it soon goes negative and so ends up dampening and slowing the response after the initial kick in output.

This is somewhat difficult for many engineers to grasp. It is common to believe that the derivative will increase the output for a short time. It is not always recognized that the derivative will also dampen and slow the response.

The Control Engineering article published on 2/1/2010 also reviews this in some detail.

In summary the article suggests modest amounts of D added to slow loops can reduce overshoot in critically tuned loops and help to stabilize them where PV signal noise is relatively low.

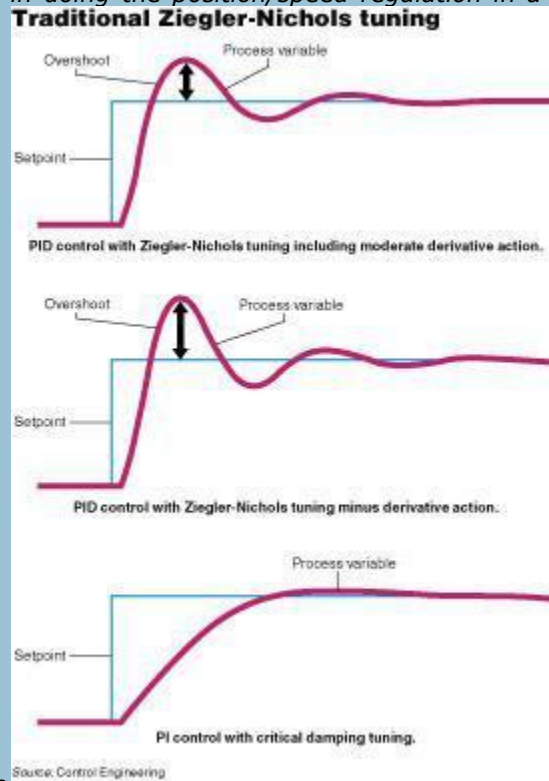
I have never found adding any D to be particularly beneficial for typical process control applications. The only exception to this is when tuning a speed/position loop on a tightly coupled high gain servo drive system. In this case adding D kept the system from overshooting on accelerations and deceleration. (Note: Drive loop tuning is not a typical process application).

In the LinkedIn discussion on the "Automation Engineers" group I advised,

*"I have yet to find a temperature application where D has been needed. Most systems have such a large lag time that D can make things go unstable quite quickly. My suggestion is to forget about D"*

unless you know you can monitor and correct this when needed. Also for newbies to PID control, just forget the D in temperature, pressure, and most flow applications.

Where I have used D has been in doing the position/speed regulation in a tightly coupled and high



performance (minimal lag) drive

system. There D had a

huge impact on getting the drive to track a moving position reference, especially on rapid accelerations and decelerations of the drive. In fact the drive ran on PD only. (The gains were so high the need for I was negligible and actually caused a similar kind of instability that adding D causes on slow systems but instead of responding to quickly, It responded to slowly not being able to catch up before the process would get out of control).

That being said, I see from the other comments that others have been able to add in some D to help even temperature loops get up to temperature faster. Still it seems this would only be worth any extra effort if it actually affected the cycle time of a machine or process. If the temperature needs to be maintained at some level it likely wouldn't matter."

I found it particularly interesting that the Controls Engineering article notes in a side bar that the faster response can be achieved without any overshoot by adjusting the P and I terms only when it says:

"Ironically, overshoot can be eliminated entirely by leaving the derivative action disabled, decreasing the proportional gain and increasing the integral time. The bottom trend chart shows the process's setpoint response under the influence of a PI-only controller tuned according to the critical damping rules that call for a 69% smaller proportional gain and an integral time lengthened by a factor equal to the process gain. No derivative action is required."

Compare the 3 trends in the figure: The 2<sup>nd</sup> graph is PI critical tuning, the 1<sup>st</sup> has derivative added to reduce overshoot, but note the 3<sup>rd</sup> which has almost no overshoot at all by reducing P and increasing I per the previous paragraph.

### **Conclusion**

With a lot of digital systems there are other options for getting quicker control over a process than adding D. This includes just opening the output fully on startup and bring it down once setpoint or another process parameter is reached, or swap in hot PI tuning parameters on startup or upsets and put in slower and smoother tuning parameters when things are near setpoint to keep output fluctuations low and slow and the process steady.

For most process control PID controller applications, forget the D. It will save time and additional headaches way more often than not. Use D sparingly, and then only when needed.

Source: <http://hennulat.wordpress.com/2010/02/15/the-d-in-pid-control/>