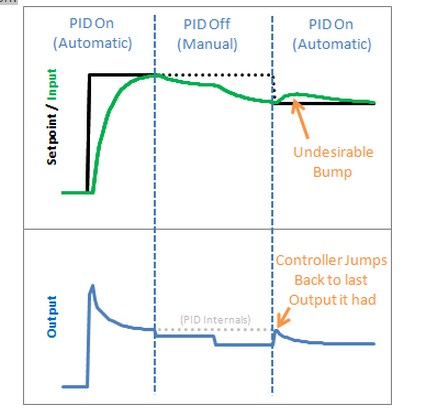
**Improving the Beginner’s PID: Initialization**

(This is Modification #6 in a [larger series](http://brettbeauregard.com/blog/2011/04/improving-the-beginner%e2%80%99s-pid-initialization/improving-the-beginners-pid-introduction) on writing a solid PID algorithm)

**The Problem**

In the last section we implemented the ability to turn the PID off and on. We turned it off, but now let’s look at what happens when we turn it back on:



Yikes! The PID jumps back to the last Output value it sent, then starts adjusting from there. This results in an Input bump that we’d rather not have.

### The Solution

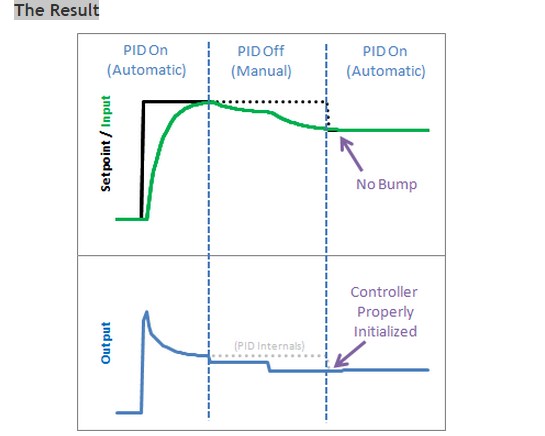
This one is pretty easy to fix. Since we now know when we’re turning on (going from Manual to Automatic,) we just have to initialize things for a smooth transition. That means massaging the 2 stored working variables (ITerm & lastInput) to keep the output from jumping.





We modified SetMode(…) to detect the transition from manual to automatic, and we added our initialization function. It sets ITerm=Output to take care of the integral term, and lastInput = Input to keep the derivative from spiking. The proportional term doesn’t rely on any information from the past, so it doesn’t need any initialization.

### The Result



We see from the above graph that proper initialization results in a bumpless transfer from manual to automatic: exactly what we were after.  
[Next >>](http://brettbeauregard.com/blog/2011/04/improving-the-beginner%e2%80%99s-pid-initialization/improving-the-beginners-pid-direction)

### Update: Why not ITerm=0?

I have been getting a lot of questions recently asking why I don’t set ITerm=0 upon intialization. As an answer, I’d ask you to consider the following scenario: The pid is in manual, and the user has set the output to 50. After a time, the process steadies out to an input of 75.2. The user makes the Setpoint 75.2 and turns on the pid. What should happen?

I contend that after switching to automatic the output value should stay at 50. since the P and D terms will be zero, the only way this will happen is if ITerm is initialized to the value of Output.

If you are in a situation where you need the output to initialize to zero, there is no need alter the code above. Just set Output=0 in your calling routine before turning the PID from Manual to Automatic.