Basic
Application Guide
To
PID2 Instruction

May 18, 1998
Revision 2.0
Schneider Automation

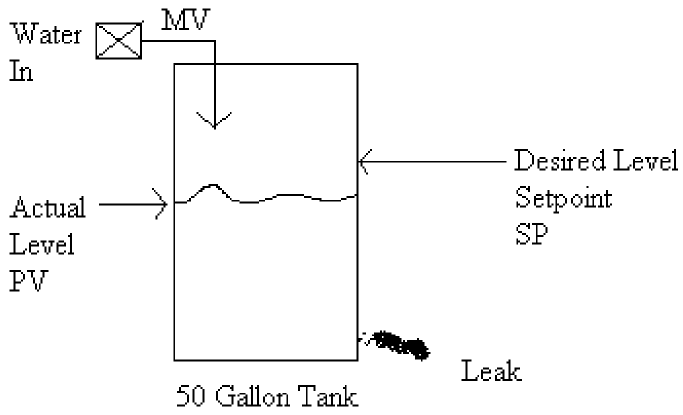
Introduction

What follows is a brief description of how to setup the PID2 instruction block on Schneider Automation controllers. This assumes that you have read some of the existing documentation on the PID2 block and need further assistance.
This paper looks at terminology used and then gives an example case. A rough troubleshooting guide is given at the end of this application guide. This does not discuss Integral preload, or go into the in-depth theory of PID. That will be covered in the advanced discussion on PID2 and the PCFL blocks.

Terminology

Lets take the example of controlling the level of water in a tank. For this test case example you have a 50 gallon drum that has a leak in it and you are using a water hose with a control valve on the end of it to fill it. The water slowly leaks out of the tank and the control valve will variably open to maintain a certain level in the tank.

The following Figure shows the tank:



The following terminology is used.

SP = Setpoint this is the Desired Level in the tank
PV = Process Variable, this is the Actual level of water in the tank.
MV = Manipulated Variable, this is the control valve that we are variably opening or closing.

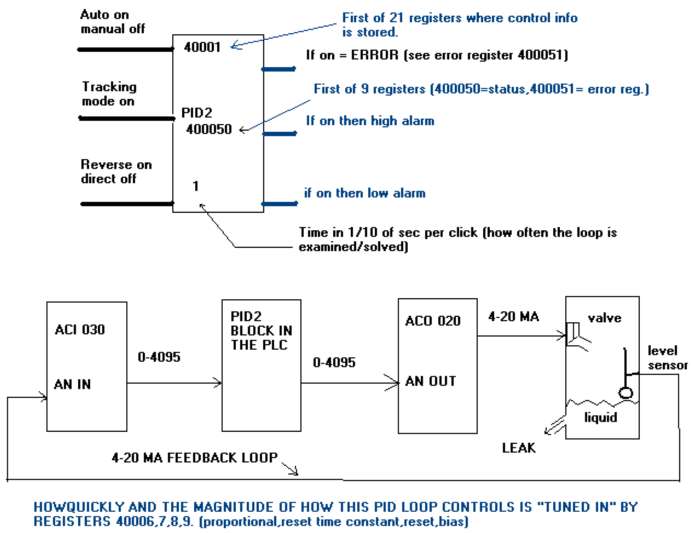
The goal is to have the MV which is the control valve open to let the right amount of water in that is the same flowrate as the amount of water that is leaving the tank via the leak.

The MV which is the output from the PID2 block is the result from the equation
MV = Proportional Contribution + Integral Contribution + Derivative Contribution + bias

PID2 Instruction Block

Overview

What is shown below is how the PID2 block is programmed in the ladder logic and how the PID2 block is used to control our actual process.



The above PID2 block shows the Top, Middle, and Bottom input and output from the block. In our example we only have the top input on.
The block diagram below this illustrates how the input from the level sensor goes into the ACI030 Quantum analog input module. The ACI030 generates a number from 0 to 4095 which goes into the PID2 block. The PID2 block solves and outputs a 0 to 4095 value that is output through the ACO020 module which controls how open or closed the water valve that is feeding the tank.

PID2 Specifics

The PID2 instruction block has built in DX zoom screens that provide easier setup. After putting the PID2 block on the screen and entering in the 4x registers, highlight the particular PID2 block you want to work on and:

Modsoft: Hit the Alt Z button for zoom
Concept: Hit the Control and D button

Example Register Configuration

Top Node	400001
Middle Node	400050
Bottom Node	1

What follows is a listing of all of the registers included in the PID2 block. These are not in the order that the zoom screens show but are described in the order that makes sense to the user.

Item 1. Raw Analog Measurement
Register 400014

This is the raw analog signal that is the actual level of water in the tank, PV. The actual level of water in the tank is actually a variable signal like a 4-20mA signal. When the 4-20mA signal comes into an input card on the PLC it is converted by the PLC to COUNTS. So in this example a level of 0 inches would be 0 counts and a level of 50 gallons would be 4095 counts.

THIS MUST RANGE FROM 0 to 4095. DO NOT SCALE THIS BEFORE PUTTING THIS INTO THIS REGISTER.

Item 2. High Engineering Range / Low Engineering Range
Register 400012 / Register 400013

The High and Low engineering range are the numbers you want to scale your raw input to. In our test case example we have a count value of 0 that would correspond to 0 gallons of water in the tank and a count of 4095 would correspond to a High engineering range of 50. Thus when a value of 2047 goes into the Raw Input register a Scaled PV will show 25.
High Engineering Range = 50.
Low Engineering Range = 0.

Item 3. Scaled PV.
Register 400001

The Scaled PV is calculated for you by the PID2 block. It takes the Raw analog signal in and converts it to a value between 0-50, which is the engineering ranges.

Item 4. Set Point (in engineering units)
Register 400002
This is the setpoint that you want to control your level of water to. Lets take a test case to say that you want to control the level to 40 gallons of water in the tank. Then register 400002 is equal to 40. This must be between the high and low engineering ranges.

Item 5. MV (Loop Output)
Register 400003

This is the output from the PID2 block that will be fed to the control valve through an analog output card. This is between 0 to 4095. If your output card that you are feeding this number into requires different ranges you must do the scaling outside of the PID2 block. In our case a value of 0 would completely close the valve and a value of 4095 would completely open the valve.

Item 6. High Alarm Limit
Register 400004

If the actual water level goes above this value you would turn the output middle node on. If you set this to 50 and the water level goes above 50 the middle node output would go on and if you put a coil on this output you could sound an alarm.

Item 7. Low Alarm Limit
Register 400004

If the actual water level goes below this value you would turn the bottom node output on. If you set this to 5 and the water level goes below 5 the bottom node output would go on and if you put a coil on this output you could sound an alarm to say that the water level in the tank is way too low.

Item 8. Proportional, Reset Time Constant, Reset and Bias
Register 400006, 7 , 8, and 9.

These registers control how quickly and the magnitude of how this PID2 block controls. Please refer to the TUNING section for detailed descriptions.

Item 9. High Integral Windup Limit
Low Integral Windup Limit
Register 400010, Register 400011

When the level of the tank is not equal to the setpoint level the PID2 block constantly tries to adjust the MV output value so that it can make the tank level equal to it. (If you are using the integral contribution.) For example if the tank level is less than the Setpoint the PID calculation constantly over time keeps increasing the integral contribution. There must be a limit where the Integral Contribution stops accumulating. The High Integral Windup Limit is the limit that the MV can get to that when it gets to the Limit will stop accumulating the Integral Portion.
The High Integral Windup Limit in our case would be 4095.
The Low Integral Windup would be the minimum value that the MV can get to and then stop the Integral contribution. The Low Integral Windup Limit would be 0.

Item 10. Pointer to the Loop Counter Register
Max Number of Loops in Scan.
Register 400015, Register 400016

In our case we are going to set both of these registers to zero.
The reason to use these is if you have more than one PID2 block (actually quite a few) and are very concerned to keep the scan time of your program down to a minimum.

Lets take an example of 20 PID2 blocks in one program and you only want 2 of these PID2 blocks to solve on the same scan.
Register 4000016 would be equal to 2.
Now we need a free additional 4X register (Outside of the PID2 block) Lets take 400100.
Register 400015 would equal 100. This tells the PID2 block it can use register 400100.
You must also add logic for this to work. At the very first Network and first Segment set 400100 equal to zero. As the logic proceeds every PID2 block that solves on this scan will increase the number in 400100 until it gets to 2. When it gets to 2 it will solve no more PID2 blocks in this scan. That is why you clear the register on the first Network.
If you do not clear this register the PID2 blocks will stop solving. Since we have 20 PID2 blocks we would need to clear each of the 20 PID2 blocks Loop Counter Registers on each scan.

Item 11. Ptr to the Reset Feedback Input Register
Register 400017

This is very important register. This MUST be the MV register. In this block the MV register is 400003.
Thus register 4000017 is equal to 3. Each PID2 block must point to its own MV.
The PID2 block uses the output from itself in it own equation. Thus it needs to know which register MV is in.

Item 12. Output Clamp High.
Register 400018

Put the value that you want the MV to be max. Set Register 400018 to 4095.

Item 13. Output Clamp Low.
Register 400019

Put the value that you want the MV to be min. Set Register 400018 to 0.

Item 14. Rate Gain Limit
Register 400020

If you are using the derivative action of the PID2 block this applies. Fine tuning of this parameter will be covered in the advanced PID2 application guide.

Item 15. Pointer to the Track Input Register
Register 400021

We will not utilize this feature in our example so Register 400021 = 0. Also the middle input node on the actual PID2 block will not have any logic tied to the power rail.

Item 16. Bottom Node Input to the PID2 Block

This bottom node input controls the direct or reverse acting of the PID2 block.

Item 17. Manual and Automatic

The Top Input node to the PID2 block controls when the PID2 block is in Automatic Mode or Manual Mode. Manual Mode is that the PID2 block does not change the MV. In Manual Mode the user must change the MV for a manual change.

Item 18. Bumpless Transfer

When you switch from Automatic to Manual Mode, if you want the PID2 block to be bumpless you must do the following:
In manual mode write the value that you want the MV to be, into the MV register. If you write the value directly to the analog cards output 4x register, the PID2 block does not know what that last value is. It will assume that the last MV value that is in the register is the last value and will ramp up or down from there.

Item 19. Bottom Node of the PID2 block

You enter the time that you want the particular PID2 block to solve. If you have a quick responding process you would want to make this value as small as possible. This is in 1/10th of a second. So for example if you put a 1 in there the block will solve every 1/10th of a second or every 100 milliseconds. When the PID2 block does not solve it does not add any scan time to your program. When it solves it does add time to your scan time. This is why Item 10 can be a great benefit if you have a crucial scan time and have many PID2 blocks.
For our example we are going to put a value of 5 in the bottom node to solve every 5/10ths or ½ a second.

PID2 Tuning

After all of the appropriate other entries are made there are only three items to adjust to tune the PID2 loop. They are:

Register 400006 Proportional Band	Proportional Factor
Register 400007 Reset Time Constant	Integral Factor
Register 400008 Rate Time Constant	Derivative Factor

Step 1. Set your Reset Time and Rate Time constant to zero. Give the Proportional Band a value of 250.

Step 2. Make sure that the PID2 block is in automatic mode. (Top node input to the PID2 block is tied to the power rail.) Monitor your process. We want to put a value into the Proportional Band that will cause the level in the Tank to go over and under the setpoint evenly. In this example lets say that this is the value 125. This value causes the level in the tank to go 3 inches over and 3 inches under the setpoint.
Make sure that you keep decreasing the gain until you get an even oscillation. The key to this is that you get an even oscillation above and below your setpoint. What this does is establish how much proportional amount is required.

Step 3. Monitor the time between when the level in the tank is 3 inches over to when it is 3 inches under and then when it is 3 inches over again. The time between when it is over to over again is the Ultimate Time. Lets say that it is 3 minutes.

Step 4. Now use the following equations to put numbers in and you should have good PID control:

Proportional Band = Proportional Band Number that made even oscillation * 2
So Proportional Band = 125*2 = 250.

Reset Time = 1 / Ultimate Time (minutes) * 100
Reset Time = 1 / 3 *100 = 33.

Rate Time = Reset Time / 8
Rate Time = 33 / 8 = 4

This is the idea tuning method developed by Ziegler Nichols, called the Ultimate Gain Method. This is a rough guide to assist your PID tuning. There are many processes that this tuning method does not work well on, however this will get you close. The advanced section will discuss how some processes need more Derivative action that Integral action for appropriate tuning.

Troubleshooting

1. Problem
My block does not solve and the middle node register +1 gives an error code. In the test case example from above this would be register 400101.

1. Solution
An error code says that one of the registers that you have put the following information has been filled in incorrectly. Please refer to the Modicon Ladder Logic Block library for a listing of the error codes.

2. Problem
The block solves and there is no error but the output is going in the wrong direction to control.

2. Solution
So for example in our example case, if when the water level was below the setpoint and the water fill valve MV was closing instead of opening. Tie the Bottom input to the power rail. This will make the PID2 decrease the output when the Error increases.

3. Problem
The block solves and there is no error and it controls correctly but it freezes the output of the MV even though you are not exactly at the setpoint.

3. Solution
Make sure that you have a full 0 to 4095 range going into the raw analog signal.
Make sure that the PTR to Loop Feedback Register is set to the MV (see item 11.)
Make sure that if you have a Max Number of Loops Per Scan register set that you clear the register every scan (see item 10).