



PF2100 PID CONTROLLER

APPLICATION NOTE

v1.2

Revised Jan 26, 2016

by Ryan Northcott



Table of Contents

1	Overview	1
1.1	What is a PID Controller?.....	1
1.2	How Does the PF2100 Use the PID Controller?.....	1
1.3	Why Was the Controller Changed?	1
2	Using the PID Controller	2
2.1	Menu Settings	2
2.2	Recommended Tuning Procedure	3
2.2.1	Tune the Proportional Band Setting	3
2.2.2	Tune the Integral Setting.....	3
2.2.3	Tune the Derivative Setting	3
3	Theory of Operation	4
3.1	Continuous Time Model	4
3.2	Discrete Time Implementation	4
3.3	Control Exceptions	5
3.4	Detailed Explanation	6
3.4.1	Current Error	6
3.4.2	Accumulated Past Error	6
3.4.3	Integral Setting and Term.....	6
3.4.4	Predicted Future Error.....	6
3.4.5	Derivative Setting and Term.....	7
3.4.6	Total Error	7
3.4.7	Gain and Proportional Band Setting.....	7
3.4.8	Output Value	7
3.4.9	Low Fire Settings.....	8
4	Document Revision History	9



1 Overview

This Application Note discusses the PID Controller implemented in PF2100 Door Card Firmware E1.8.116 and later.

1.1 What is a PID Controller?

A PID Controller is a type of feedback control algorithm commonly used in industrial control applications. It is often considered one of the best controllers for dealing with a generic process. The purpose of the controller is to maintain a process variable as close as possible to a user defined setpoint. This is accomplished by continuously measuring the difference between the current value of the process variable and the desired setpoint value and then adjusting some manipulated variable to compensate for the difference.

The PID Controller gets its name from the initials of the three tuning coefficients that are built into the control algorithm: the Proportional coefficient, the Integral coefficient, and the Derivative coefficient. These coefficients set the Controller's response to three types of errors: the current error (Proportional), the sum of all past errors (Integral), and the predicted future error (Derivative).

1.2 How Does the PF2100 Use the PID Controller?

In the case of the PF2100, the PID Controller is used to maintain the Process Temperature as closely as possible to the Process Setpoint. The Controller accomplishes this by adjusting the 4-20mA output signal which in turn drives a proportional valve. The more the valve is open, the more gas passes through, and the more heat is applied to the process. Feedback is provided to the control algorithm by subtracting the thermocouple measurement (the Process Temperature) from the desired temperature (the Process Setpoint).

1.3 Why Was the Controller Changed?

In comparison to the 4-20mA Proportional Valve Control algorithm in previous versions of PF2100 firmware, the PID controller gives you greater flexibility by allowing you to fine tune the control algorithm to have a faster response, minimize settling time, and/or minimize over/undershoot. In short, the PID controller is much better!



2 Using the PID Controller

This section of the Application Note gives a brief overview of the various settings associated with the PID Controller and then provides a recommended procedure for fine tuning the controller.

2.1 Menu Settings

There are eight menu settings that are used to setup the PID Controller. The first five of these are the same as in previous firmware versions. The three PID settings are new in FW E1.8.116 and replace the “4-20 Gain Setting” which used to be in menu 5 in FW E1.8.115 and earlier. These settings are described as follows:

Setting and Description	Range	Default Value
4-20 Output Mode (Menu 5) – This must be set to “Valve Control” to enable the PID Controller and to unhide the other settings listed below.	“Valve Control”, “Temperature”	“Valve Control”
Low Fire Enable (Menu 5) – This must be set to “On at Proc Setpnt” when using a Temperature Control Valve. This will enable the Low Fire Setpoint and associated Low to High Fire Delay Setting.	“Disabled”, “On at Proc Setpnt”, “On at Low Fire Setpnt”	Disabled
Low to High Fire Delay (Menu 5) – This is used to set the draft establishment time. This is the amount of time that must elapse after the pilot is ignited before the main burner is turned on.	30 to 600secs	30secs
4-20 Low Fire Setting (Menu 5) – When the system enters the Low Fire State, it will linearly ramp the 4-20 Output signal from zero to this value over a 15s period. Once High Fire is on, the PID controller will not set the 4-20 Output less than this setting. This value is encoded as a percentage of fully open.	0 to 70% in 1% steps	40%
Low Fire Setpoint (Menu 1) – Used to adjust the temperature at which the Low Fire Valve will turn off.	2 to 1349C	85C
PID: PB Setting (Menu 1) – This is the “Proportional Band” setting which is expressed in percent and is inversely proportional to the controller’s gain.	0.0 to 999.9% in 0.1% steps	10.0%
PID: I Setting (Menu 1) – This is the “Integral” setting which is expressed in units of Minutes per Reset. It is the amount of time required for the integral term to yield the same output that the proportional term does immediately.	0.0 to 999.9 min/rst in 0.1min steps	4.0min/rst
PID: D Setting (Menu 1) – This is the “Derivative” setting which is expressed in units of Minutes. It provides a more severe output when the error moves in the wrong direction and a dampening effect when the error moves in the correct direction. By default, the derivative term is disabled (set to zero).	0.0 to 999.9min in 0.1min steps	0.0min



2.2 Recommended Tuning Procedure

The PF2100's default PID settings should result in good temperature control in most common heater applications. If the temperature control is unstable or if faster control response is needed, the following procedure can be followed to adjust the PID settings. This procedure is not a comprehensive method for adjusting PID Controllers but provides some general guidelines. This procedure is iterative and therefore may require a significant amount of time to follow.

2.2.1 Tune the Proportional Band Setting

1. Start with the Proportional Band Setting set to a large value such as 500% and with the Integral and Derivative settings set to their default values (4.0min/rst and 0.0min respectively).
2. Make a small step change (such as 5%) to the Process Setpoint and monitor the resulting temperature change.
3. Repeat decreasing the Proportional Band Setting in 50% steps until a Process Setpoint change results in overshoot of the Process Setpoint or until oscillation results.
4. Increase the Proportional Band Setting until overshoot and oscillation no longer occur.

2.2.2 Tune the Integral Setting

1. Set the Integral setting to 10min/rst.
2. Make a small change to the Process Setpoint and monitor the temperature change.
3. Repeat decreasing the Integral in ½ minute steps until a Process Setpoint change results in overshoot of the Process Setpoint or until oscillation results.
4. Increase the Integral Setting until stable.

2.2.3 Tune the Derivative Setting

If there is noise on the thermocouple inputs, using the Derivative Setting can dampen the effects of the Proportional and Integral Settings. For this reason, the Derivative Setting can usually be left set at zero unless a very fast response is needed. If it is required, follow this tuning procedure:

1. Set the Derivative Setting to 10 minutes.
2. Decrease the Derivative Setting until a small Process Setpoint change results in a fast response with some oscillation.
3. Increase the Derivative Setting until the oscillation in response to a Process Setpoint change is dampened.



3 Theory of Operation

This section is intended for those who need to understand the implementation details of the PID controller in the PF2100 more intimately. This information may be useful to those who want to tune the controller for optimal performance in a given application. An understanding of advanced math concepts including calculus will aid the reader in understanding this section.

3.1 Continuous Time Model

The PID controller implemented by the PF2100 follows the “standard form” and incorporates a reciprocal gain coefficient. The equations on which it is based are as follows:

$$Output(t) = \frac{100}{PB} \left(Error(t) + \frac{1}{I} \int_0^t Error(\tau) d\tau + D \frac{d}{dt} Error(t) \right)$$

$$Error(t) = \frac{Setpoint(t) - Input(t)}{Setpoint(t)}$$

Where:

- Output(t)* = The 4 – 20mA Output percentage at time *t*
- Setpoint(t)* = The Process Setpoint at time *t*
- Input(t)* = The measured Process Temperature at time *t*
- Error(t)* = The Percentage Error at time *t*
- PB* = The Proportional Band setting
- I* = The Integral setting
- D* = The Derivative setting
- t* = The time

3.2 Discrete Time Implementation

Since the temperature and error measurements are taken by the PF2100 at fixed 2 second intervals, the integral term must be implemented as a summation of discrete samples. Likewise, the derivative term must be implemented as a difference between samples per unit time. To prevent Derivative Kick (undesired sudden response to sudden change in error) the derivative is further averaged over 3 consecutive samples. This also helps to filter out noise. Factors of 60 have been added to convert between minutes and seconds as required. The discrete time equation is therefore:

$$Output(n) = \frac{100}{PB} \left(Error(n) + \frac{2}{60I} \sum_{N=0}^n Error(N) + \frac{60D}{3} \sum_{N=n-2}^n \frac{Error(N) - Error(N-1)}{2} \right)$$

$$Error(n) = \frac{Setpoint(n) - Input(n)}{Setpoint(n)}$$

Where:



- Output(n)* = The 4 – 20mA Output percentage at interval n
- Setpoint(n)* = The Process Setpoint at interval n
- Input(n)* = The measured Process Temperature at interval n
- Error(n)* = The Percentage Error at interval n
- PB* = The Proportional Band setting
- I* = The Integral setting
- D* = The Derivative setting
- n* = The interval number

3.3 Control Exceptions

There are some situations where the 4-20mA Output is not controlled by the PID Controller Algorithm. These situations are as follows:

During the Low to High Fire Delay

During the first 15 seconds of the Low to High Fire Delay, the system will linearly ramp the 4-20 Output from zero up to the 4-20 Low Fire Setting using the following equation:

$$Output(t) = \begin{cases} \text{If } t \leq 15, & \frac{t * LF}{15} \\ \text{If } t > 15, & LF \end{cases}$$

Where:

- Output(t)* = The 4 – 20mA Output percentage at time t
- LF* = The 4 – 20mA Low Fire Setting
- t* = The elapsed time in seconds since the beginning of the Low to High Fire Delay

Above the Low Fire Setpoint

If the temperature rises above the Low Fire Setpoint, the 4-20 Output will be set to zero as illustrated by the following equation:

$$Output(t) = 0$$

Where:

- Output(t)* = The 4 – 20mA Output percentage at time t
- t* = The elapsed time in seconds since the Process Temp rose above the Proces Setpoint

If the PID Controller Output is Less Than the 4-20 Low Fire Setting

If the PID Controller Output drops below the 4-20 Low Fire Setting, the 4-20 Output will be set to the 4-20 Low Fire Setting as illustrated by the following equation:

$$Output(t) = LF$$

Where:



$Output(t)$ = The 4 – 20mA Output percentage at time t
 LF = The 4 – 20mA Low Fire Setting
 t = The elapsed time in seconds since the PID Controller Began Running

3.4 Detailed Explanation

This section explains the basic theory of operation by walking through the definition of various terms related to the PID controller. These terms are not listed in alphabetical order. Rather, they are in an order that attempts to explain the operation of the controller by building on the previously defined terms.

3.4.1 Current Error

The Current Error is simply the difference between the Process Setpoint and the measured Process Temperature expressed as a fraction of the setpoint.

$$Current\ Error = \frac{Process\ Setpoint - Process\ Temperature}{Process\ Setpoint}$$

3.4.2 Accumulated Past Error

The Accumulated Past Error is the sum of all past errors while the controller is powered up and running the process control algorithm. This sum is reset to zero when the PF2100 is powered off. The equation for calculating Accumulated Past Error is:

$$Accumulated\ Past\ Error = Sum\ of\ All\ Previous\ "Current\ Error"\ Values$$

3.4.3 Integral Setting and Term

The Integral Setting is used to scale the Accumulated Past Error to create the Integral Term. The scaling equation is implemented as follows:

$$Integral\ Term = \frac{Accumulated\ Past\ Error}{Integral\ Setting}$$

The Integral Term is used to influence the PID Controller's Output Value to remove any continuous error offset created by the Proportional Band Setting. The Integral Setting has units of "Minutes per Reset". In other words, it represents the number of minutes it takes to adjust the Output Value by the amount of the Proportional Band Setting. The result is that the output will constantly be adjusted until there is no error. Decreasing the Integral Setting will speed up the output adjustment however too low of a setting can result in severe output changes causing temperature oscillation.

3.4.4 Predicted Future Error

The Predicted Future Error is an estimate of what the error will be in the near future. This estimate is based on observing whether the Current Error is increasing or decreasing and at what speed. The equation for calculating Predicated Future Error is:

$$Predicated\ Future\ Error = \frac{Current\ Error - Previous\ Error}{Elapsed\ Time}$$



3.4.5 Derivative Setting and Term

The Derivative Setting is used to scale the Predicted Future Error to create the Derivative Term. The scaling equation is implemented as follows:

$$\text{Derivative Term} = \text{Derivative Setting} * \text{Predicted Future Error}$$

The Derivative Setting is usually set to zero unless very fast response is required. Using the Derivative Setting allows the Proportional Band and Integral Settings to be set for very fast output response which by themselves would result in oscillation. The Derivative Term applies dampening to the output when the temperature is changing quickly.

3.4.6 Total Error

The PID Controller calculates the Total Error as being the sum of the Current Error, Integral Term, and Derivative Term as follows:

$$\text{Total Error} = \text{Current Error} + \text{Integral Term} + \text{Derivative Term}$$

3.4.7 Gain and Proportional Band Setting

The PID Controller's Gain determines the amount of response to a given amount of Total Error. In the PF2100, the Gain is represented by the Proportional Band (PB) Setting which is inversely related to the Gain of the controller. The PB Setting is expressed as a percentage and the gain setting is dimensionless (without units). To convert between Gain and the PB Setting, use the following simple equations:

$$\text{Gain} = \frac{100\%}{\text{PB}} \quad \text{PB} = \frac{100\%}{\text{Gain}}$$

If the PB Setting is set too large, then the gain will be too small leading to very slow reaction time. This may result in the Process Temperature never reaching the Process Setpoint.

If the PB Setting is set too small, then the gain will be high leading to a very fast reaction time. This may result in the Process Temperature continuously oscillating far above and far below the Process Setpoint.

3.4.8 Output Value

In the PF2100, the Output Value is the 4-20mA output signal used to control the proportional valve which indirectly controls the temperature. The PID Controller sets the Output Value in response to the Total Error. It is proportional to the Gain as illustrated by the following equation:

$$\text{Output Value} = \text{Gain} \times \text{Total Error}$$

As an example, consider the case where the PB Setting is 20% and the Total Error is 10%. In this case, the Gain will be 5 and the Output Value will be $5 \times 10\% = 50\%$.

Note that if the Total Error is zero, the Output Value will also be zero.



3.4.9 Low Fire Settings

For the PID Controller to function correctly, the Low Fire Settings must be setup correctly as described in the Menu Settings section of this document. Note that the PID Controller will not set the 4-20mA Output lower than the 4-20 Low Fire Setting when controlling. There are other cases where it may be set less as described in the Control Exceptions section of this document.



4 Document Revision History

Version Number	Date	Who	Description of Changes
v1.0	2013-08-16	Alan Johnson and Ryan Northcott	Initial Release
v1.1	2015-03-25	Matthew Kipper and Ryan Northcott	Corrected the discrete time equation in Section 3.2 to include the missing factor of 2 in the derivative term
v1.2	2016-01-26	Ryan Northcott	Footer updated for public release. Misc typo corrections.