



TABLE OF CONTENTS

Appendix F: PID Control

Function of PID Control	F-2
What Does PID Control Accomplish?	F-2
PID Control Analogy.	F-2
Common Applications for PID Control	F-3
Definition of PID Loop "Directions"	F-3
Forward-Acting PID Loop (Heating Loop) (Negative-Feedback Loop)	F-3
Reverse-Acting PID Loop (Cooling Loop) (Positive-Feedback Loop)	F-3
PID Control Overview	F-4
Concept of GS20(X) PID Control & Tuning	F-5
Proportional Gain (P)	F-5
Integral Time (I)	F-5
Derivative Value (D)	F-6
Proportional Integral Control (PI)	F-6
Proportional Derivative Control (PD).	F-6
Proportional Integral Derivative Control (PID)	F-6
Tuning Example for PID Control	F-7
DURApulse GS20/GS20-GS2mode Parameters Involved in PID Control	F-9

FUNCTION OF PID CONTROL

GS20(X) series AC drives can be used to control an automated process by the Proportional-Integral-Derivative (PID) control method.



NOTE: A PID Configuration Spreadsheet tool is available for download from the drive item page support resources section.

WHAT DOES PID CONTROL ACCOMPLISH?

The primary benefit of PID control is that it achieves and maintains the desired steady-state condition of a process better and more smoothly than does ON-OFF control.

The GS20(X) drive PID algorithm constantly assesses the amount and rate of change of the quantity being controlled (Process Variable) and its deviation (Process Error) from the desired steady-state value (Setpoint). The GS20(X) drive then variably adjusts its frequency output as much or as little as needed to keep the Process Variable as close as possible to the Setpoint.

Simple ON-OFF control systems, on the other hand, continually bounce back and forth above and below the Setpoint value, but cannot maintain the Process Variable at the Setpoint value.

PID CONTROL ANALOGY

PID controllers are all around us. Many times we don't realize that we are the PID controller in a control loop. For example, the driver of a car is the PID controller for the car's speed.

PID Control System Variables:

- *Desired Speed* \approx *Setpoint*
- *Actual Speed* \approx *Process Variable*
- *Gas Pedal* \approx *Control Variable*
- *Speedometer* \approx *Feedback*

Proportional Control: The farther away you are from your Desired Speed, the more you press the gas pedal. If you did this starting from a stand-still, you would floor it and probably shoot far past the Desired Speed. Once the speed "settled in," you would never hold exactly at your Desired Speed because the difference between Desired and Actual Speed would get very small and you only have so much control over the pedal and your foot; not enough to hold the perfect speed consistently. So, Proportional Control adjusts the output based on the *difference* between the Setpoint and Process Variable much more accurately in a fine-tuned way.

Integral Control: If your Desired Speed is 70mph and your car consistently goes 69mph, you will realize that you need to press the gas pedal a little more (to overcome wind resistance, a hill, etc.). The longer you are under the Desired Speed, the more gas you give the car. That is fundamentally what Integral Control does; adjust the output based on *how long* the system is away from the setpoint.

Derivative Control: In the situation above, assume that you start going up a hill. The car's Actual Speed gets farther away from the Desired Speed, so the Proportional Control makes you press the gas pedal more. The longer the speed stays below setpoint, Integral Control makes you press the gas even more. Now assume that your car tops the hill and starts going downhill. Your speed suddenly gets faster (the error between Desired Speed and Actual Speed), so Proportional causes you to slightly let off the gas. But Integral still keeps adding to the pedal (since you still haven't reached Desired Speed). Your internal Derivative Control sees that you are rapidly approaching the Desired Speed, so you begin to let off the gas quickly. That is Derivative Control; it adds or subtracts to the Control Variable based on *how quickly* the system is approaching (or leaving) the setpoint.

COMMON APPLICATIONS FOR PID CONTROL

- 1) *Flow control: A flow sensor is used to feed back the flow rate in a pipe, and the GS20(X) drive PID adjusts its output frequency to the pump that forces the liquid or gas through that pipe.*
- 2) *Level control: A level sensor is used to feed back the liquid level in a reservoir or tank, and the GS20(X) drive PID adjusts its output frequency to the pump that fills or empties that tank.*
- 3) *Pressure control: A pressure sensor is used to feed back the pressure in a tank, and the GS20(X) drive PID adjusts its output frequency to the pump that pressurizes or vacuums that tank.*
- 4) *Speed control: A speed sensor is used to feed back the shaft speed of a motor or machine driven by that motor, and the GS20(X) drive PID adjusts its output frequency to that motor.*
- 5) *Temperature control: A thermocouple or thermistor is used to feed back the temperature of an area or device, and the GS20(X) drive PID adjusts its output frequency to the fan that affects that temperature.*

DEFINITION OF PID LOOP "DIRECTIONS"



Please note that the following nomenclature describes how the GS20(X) PID system operates, which may differ from the operation of some other PID systems.

FORWARD-ACTING PID LOOP (HEATING LOOP) (NEGATIVE-FEEDBACK LOOP)

The terms "Forward-Acting," "Direct-Acting," "Heating," and "Negative-Feedback" are used to describe a PID loop that can be used to control processes such as pressure, heating, and flow (among others).

- Greater Output Frequency (Hz) drives the Process Variable (PV) upward toward the Setpoint (SP)
- GS20(X) drive frequency output increases if the Process Error is negative ($SP > PV$)

REVERSE-ACTING PID LOOP (COOLING LOOP) (POSITIVE-FEEDBACK LOOP)

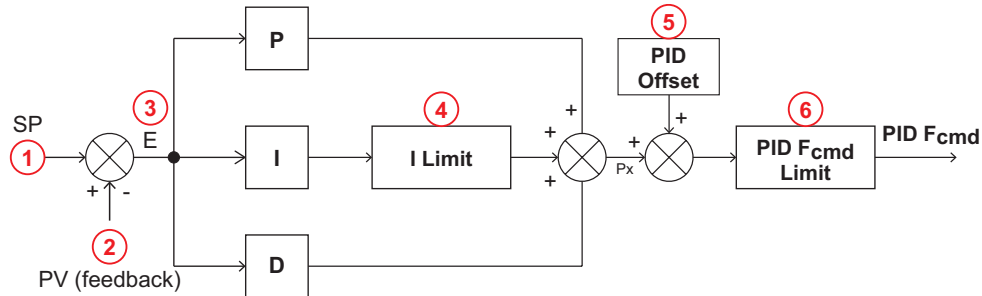
The terms "Reverse-Acting," "Cooling," and "Positive-Feedback" are used to describe a PID loop that can be used to control applications such as cooling.

- Greater Output Frequency (Hz) drives the Process Variable (PV) downward toward the Setpoint (SP)
- GS20(X) drive frequency output increases if the Process Error is positive ($SP < PV$)

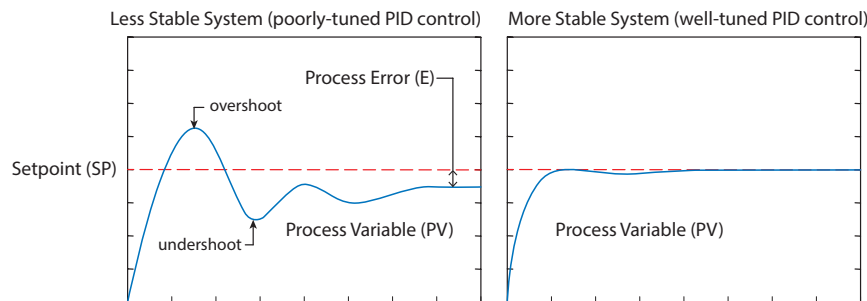
PID CONTROL OVERVIEW

PID control is a closed output and feedback loop for the purpose of automatically controlling a portion of a process to a specific condition by utilizing a target setpoint and the process’s actual condition as feedback to the controller. You determine the setpoint and let the system reach that setpoint using the process’s conditional feedback and the PID control system.

- *P* = Proportional control (also known as “Gain”)
- *I* = Integral control (also known as “Reset”)
- *D* = Derivative control (also known as “Rate”)
- Process Variable (PV) = the quantity being measured and controlled
- Setpoint (SP) (also known as Target Value) = the desired value of the Process Variable
- Error (E) = the difference between the Setpoint and the Process Variable



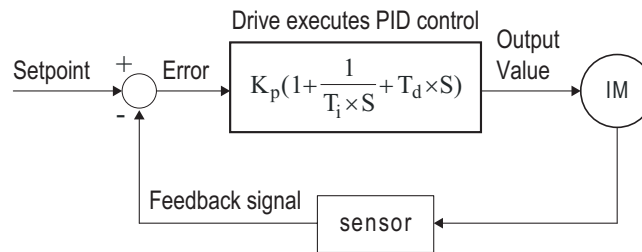
- 1) Setpoint: -100% to +100% (PID Setpoint Gain + PID Setpoint Offset)
- 2) Feedback: -100% to +100% (Feedback Gain)
- 3) Error: -100% to +100% (in percent change)
- 4) I Limit: 0~150% (Upper Limit for Integral Time P08.04)
- 5) PID Offset: P08.16 determines how the PID Offset will be controlled; by P08.17, or by an Analog Input (P03.00, P03.01)
- 6) PID F_{cmd} Limit: See P01.10, P01.11



Since a PID controller relies only on the measured Process Variable, instead of knowledge of the underlying process, it is applicable to a broad variety of system processes. By tuning the three parameters of the model, a PID controller can deal with specific process requirements. The response of the controller can be described in terms of its responsiveness to an error, the degree to which the system overshoots a setpoint, and the degree of any system oscillation. The use of the PID algorithm does not guarantee optimal control of the system or even its stability.

Some applications may require using only one or two terms to provide the appropriate system control. This is achieved by setting the other parameters to zero. A PID controller is called a PI, PD, P, or I controller in the absence of the other respective control actions. PI controllers are fairly common, since Derivative action is sensitive to measurement noise, whereas the absence of an Integral term may prevent the system from reaching its target value.

CONCEPT OF GS20(X) PID CONTROL & TUNING



K_p : Proportional Gain (P) T_i : Integral Time (I) T_d : Derivative Value (D) S: Operator

When **GS20(X) drive PID is enabled by P08.00 [PID Action/Mode]**, P08.65 “reflects” the PID Setpoint Source determined by what is set in P00.20 (Remote) or P00.30 (Local), and what Mode the Drive is in, i.e. Remote or Local Mode. PID control operates with the feedback signal as reflected by P08.65 either 0~10V voltage or 4~20mA current.

PROPORTIONAL GAIN (P)

The first parameter of GS20(X) PID control is *Proportional Gain (P08.01)*.

The GS20(X) drive’s frequency output is proportional to the Process Error (when the GS20(X) is configured for PID control). If only the Proportional Gain control component is used, the controller will not be able to get the Process Variable to exactly match the Setpoint at steady-state.

For a given process, if the Proportional Gain value is set too low, the control action will be too sluggish. If the Proportional Gain value is set too high, the control action will be unstable. To find the correct setting for Proportional Gain, set the Integral Time (I) and Derivative Value (D) to zero (0). Begin tuning the process with a low Proportional Gain value, and increase the Proportional value until the system becomes unstable. When instability is reached, reduce the Proportional value slightly until the system becomes stable (smaller values reduce system gain).

INTEGRAL TIME (I)

The second parameter of GS20(X) PID control is *Integral Time (P08.02)*.

The GS20(X) drive’s frequency output compensation due to the integral component is proportional to the integral of the Process Error. To eliminate the steady-state Process Error, an “integral component” needs to be added to the controller.

The Integral Time (I) decides the relation between integral component and Process Error. The integral component will be increased even if the error is small. It gradually increases the controller output to eliminate the error until it is 0.

Begin tuning with a higher number for Integral Time (100.0 is max; 1.0 is default), and slowly move to a smaller number until you reach the setpoint with minimized overshoot/undershoot. Tuning is normally done utilizing the GSoft2 software scope function (or an oscilloscope) to monitor the Process Variable as you incrementally change the Integral Time value until the Setpoint is satisfactorily maintained.

- **Overshoot:** The Process Variable moves further past the Setpoint than desired.
- **Undershoot:** The Process Variable does not reach the desired Setpoint.

Refer to [“Tuning Example for PID Control” on page F-7](#) of this appendix for more PID tuning information.

DERIVATIVE VALUE (D)

The third parameter of GS20(X) PID control is *Derivative Value (P08.03)*.

The GS20(X) drive's frequency output compensation due to the derivative component is proportional to the derivative of the Process Error. Derivative Value (D) control is performed based on the quickness of changes in the Process Error.

When the Proportional Gain (P) and Integral Time (I) control components are set to eliminate the Process Error so that the system runs at steady state, outside forces may suddenly cause oscillation or instability within the system. Without a Derivative Value component, the control output may be too sluggish to quickly respond to these sudden changes. The derivative component can suppress these effects by acting before the error occurs.

Begin tuning with a high Derivative Value and reduce the value to the point of system instability. Then increase the Derivative Value until the control output regains stability. Stability can be tested by moving between two wide-spread setpoint values.



Since Derivative Control is performed based on sudden changes in Process Error, it is a very sensitive control. Therefore, it may also react to extraneous signals and noise, and can easily lead to unstable system control. Derivative control is not normally required for the control of processes such as flow, pressure and temperature.

Refer to [“Tuning Example for PID Control” on page F-7](#) of this appendix for more PID tuning information.

PROPORTIONAL INTEGRAL CONTROL (PI)

When processes are controlled by Proportional Gain only, Process Error cannot be eliminated entirely. Proportional + Integral control (PI) can be used to eliminate Process Error incurred by the targeted value changes and the constant external disturbances. However, if the I action is excessively powerful, it will delay the responding correction, and will allow unstable system operation.

PROPORTIONAL DERIVATIVE CONTROL (PD)

In deciding when to use Proportional-Derivative Control, we need to understand how the system would react as a Proportional-Integral-Derivative system. When a Process Error due to a disturbance in the process occurs in a controlled system, the system sees a greater load than the derivative has provided energy to control. If that Process Error is small, the system PV can oscillate if the Proportional Gain and the Integral Time are being applied to the system too often within a small length of time. To prevent this type of system reaction, the use of Proportional and Derivative (PD) alone may be warranted. The use of Proportional Gain *and* the feed-forward action of the Derivative Value can result in a faster-acting operation to stabilize the system.

PROPORTIONAL INTEGRAL DERIVATIVE CONTROL (PID)

When choosing to use Proportional-Integral-Derivative (sometimes called PID) control, the Integral Time is utilized to provide better control of the Process Error while the Derivative Value is used to restrain PV oscillation.

TUNING EXAMPLE FOR PID CONTROL

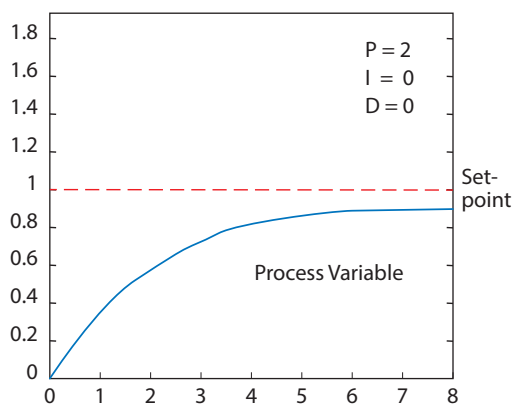
The PID settings should be adjusted, or “tuned,” with the controlled process in actual operation while monitoring the actual Process Variable. The tuning can be done using the GSoft2 software scope function, or with an oscilloscope.

We recommend starting by first adjusting the Proportion Gain only, with the Integral Time and Derivative Value set to zero. The following hypothetical example illustrates PID tuning with settings as shown:

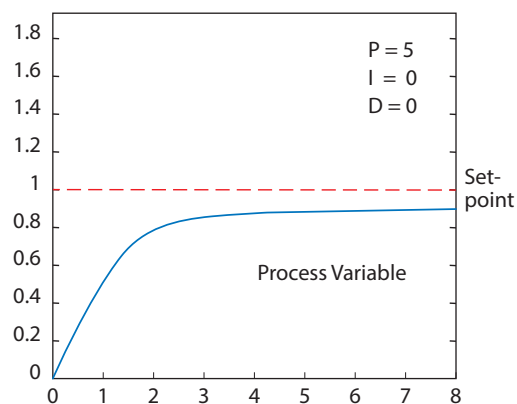
- P = Proportional Gain = GS20(X) drive parameter P08.01,
- I = Integral Time = GS20(X) drive parameter P08.02,
- D = Derivative Value = GS20(X) drive parameter P08.03.

Proportional Gain: Adjust the P setting so that the PV response is neither too sluggish, nor too fast, and without excessive overshoot or undershoot. (Process error cannot be eliminated by P)

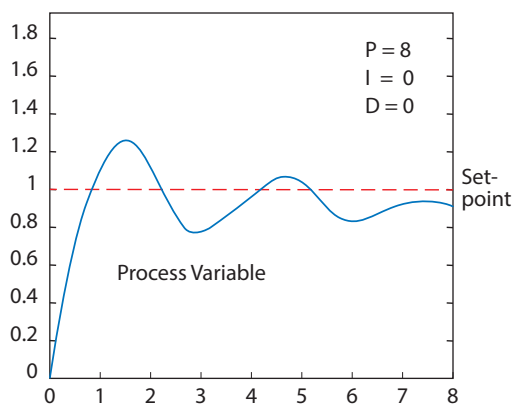
Sluggish PV response; process error



More practical PV response; process error



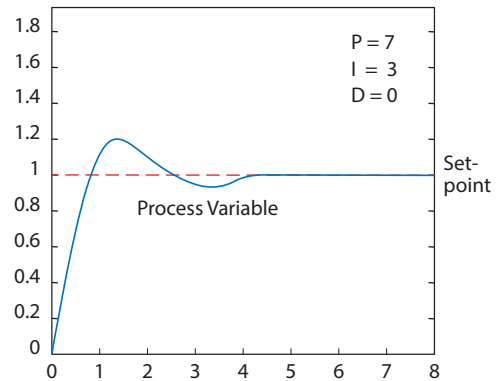
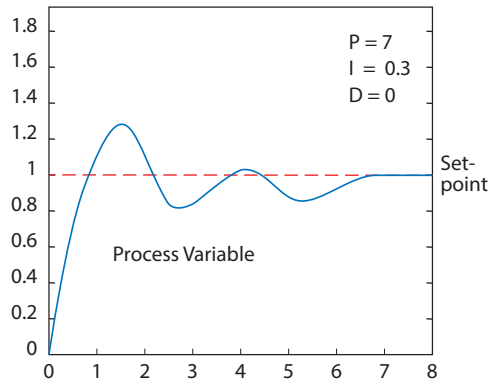
Overshoot & undershoot; process error



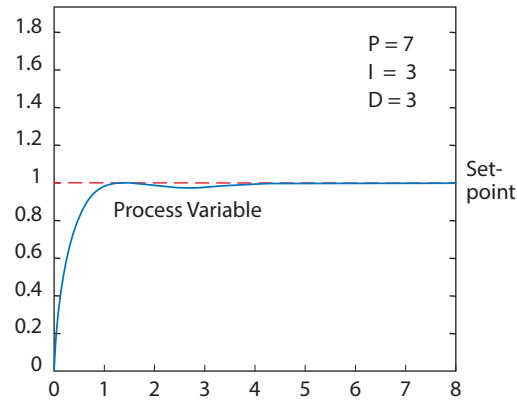
(Example continued next page)

PID Tuning Example (continued)

Integral Time: Adjust the I setting to minimize over/undershoot, and to eliminate the process error.



Derivative Value: Adjusting the D setting may not be necessary for all processes, but it can be particularly helpful in reducing over/undershoot and instability that may be caused by sudden changes in the system input variable.



DURAPULSE GS20/GS20-GS2MODE PARAMETERS INVOLVED IN PID CONTROL

The following GS20(X) AC drive parameters are often involved in setting up PID control.



NOTE: The information provided herein is applicable only to the PID function. For fully detailed parameter information and for the complete set of parameters, please refer to “Chapter 4: AC Drive Parameters.”

DURA GS20(X) & GS2 PID Parameter Comparisons – Summary			
GS20(X) PID Parameter		GS20-GS2 Mode PID Parameter	
P08.00	Terminal selection of PID feedback	P7.00	Input Terminal for PID Feedback
P08.01	Proportional gain (P)	P7.20	Proportional Control
P08.02	Integral time (I)	P7.21	Integral Control
P08.03	Differential time (D)	P7.22	Derivative Control
P08.04	Upper limit of integral control	P7.23	Upper Bound for Integral Control
P08.05	PID output command limit (positive limit)	P7.25	PID Output Frequency Limit
P08.06	PID feedback value by communication protocol	n/a	
P08.07	PID delay time	P7.24	Derivative Filter Time Constant
P08.08	Feedback signal detection time	P7.26	Feedback Signal Detection Time
P08.09	Feedback signal fault treatment	P7.27	PID Feedback Loss
P08.10	Sleep frequency	n/a	
P08.11	Wake-up frequency	n/a	
P08.12	Sleep time	n/a	
P08.13	PID feedback signal error deviation level	P3.18	PID Deviation Level
P08.14	PID feedback signal error deviation detection time	P3.19	PID Deviation Time
P08.15	PID feedback signal filter time	n/a	
P08.16	PID compensation selection	n/a	
P08.17	PID compensation	n/a	
P08.18	Sleep mode function setting	n/a	
P08.19	Wake-up integral limit	n/a	
P08.20	PID mode selection	n/a	
P08.21	Enable PID to change the operation direction	n/a	
P08.22	Wake-up delay time	n/a	
P08.23	PID control flag	n/a	
P08.26	PID output command limit (reverse limit)	n/a	
P08.27	Acceleration / deceleration time for PID command	n/a	
P08.29	Frequency base corresponding to 100.00% PID	n/a	
P08.31	Proportional gain 2	n/a	
P08.32	Integral time 2	n/a	
P08.33	Differential time 2	n/a	
P08.65	PID target value source	P7.02	PID Setpoint Source
P08.66	PID target value setting	n/a	
P08.67	Master and auxiliary reverse running cutoff frequency	n/a	
P08.68	PID deviation limit	n/a	
P08.69	Integral separation level	n/a	
P08.70	Smart start-up level	n/a	
P08.71	Smart start-up frequency command	n/a	
P08.72	Smart start-up acceleration time	n/a	
P08.75	PID2 parameter switch condition	n/a	
P08.76	PID2 parameter switch deviation 1	n/a	
P08.77	PID2 parameter switch deviation 2	n/a	
P08.78	Allowed reverse running time after start-up	n/a	
n/a	Use GS20 PLC to program this function	P7.01	PV 100% Value

DURA GS20(X) & GS2 PID Parameter Comparisons (continued)			
GS20(X) PID Parameter		GS20-GS2 Mode PID Parameter	
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.10	Keypad PID Setpoint
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.11	PID Multi-setpoint 1
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.12	PID Multi-setpoint 2
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.13	PID Multi-setpoint 3
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.14	PID Multi-setpoint 4
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.15	PID Multi-setpoint 5
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.16	PID Multi-setpoint 6
<i>n/a</i>	Use GS20 PLC to program this function	◆ P7.17	PID Multi-setpoint 7
P00.04	User Display (Can be set to display PID values)	◆ P8.00	User display
P00.03	Start-up display Selection	<i>n/a</i>	
P00.25	User-defined characteristics (COEFF ATT)	<i>n/a</i>	
P00.26	User-defined maximum value (COEFF MAX)	<i>n/a</i>	
P00.27	User-defined value (COEFF SET)	<i>n/a</i>	
P02.01 - P02.07	Multi function inputs selections: 71: Disable PID function, force PID output return to 0 72: Disable PID function, retain the output value before disabled 73: Force PID integral gain return to 0, disable integral 74: Reverse PID feedback	P03.02 - P03.05	Multi function inputs selections: 03: Multi-Speed/PID SP Bit 1 04: Multi-Speed/PID SP Bit 2 05: Multi-Speed/PID SP Bit 3 17: PID Disable (N.O.) 18: PID Disable (N.C.)
P02.13, P02.16, P02.17	Multi function outputs selections: 15: PID feedback error (P08.13, P08.14)	P03.11 - P03.12	Discrete Output Config 10: PID Deviation Alarm
P03.00, P03.01	Multi function analog input selections: 4: PID target value 5: PID feedback signal 13: PID compensation value	P04.00	Frequency config to AI. 02: Freq by AI 0-10V 03: Freq by AI 4-20mA 04: Freq by AI 0-20mA 05: Freq by RS-485
P03.20	Multi function Analog output selections. 0: Output frequency 1: Frequency command 9: AI1 10: AI2 21: RS-485	P04.11	Analog Output Config. 02: PV