PIDCON Function PIDCON T60

CONTENTS

1. DESCRIPTION

The PIDCON Special Function is intended for the control of closed loop systems in process control applications. It incorporates a number of advanced features previously not found in the existing PIDABS and PIDINC Special Functions.

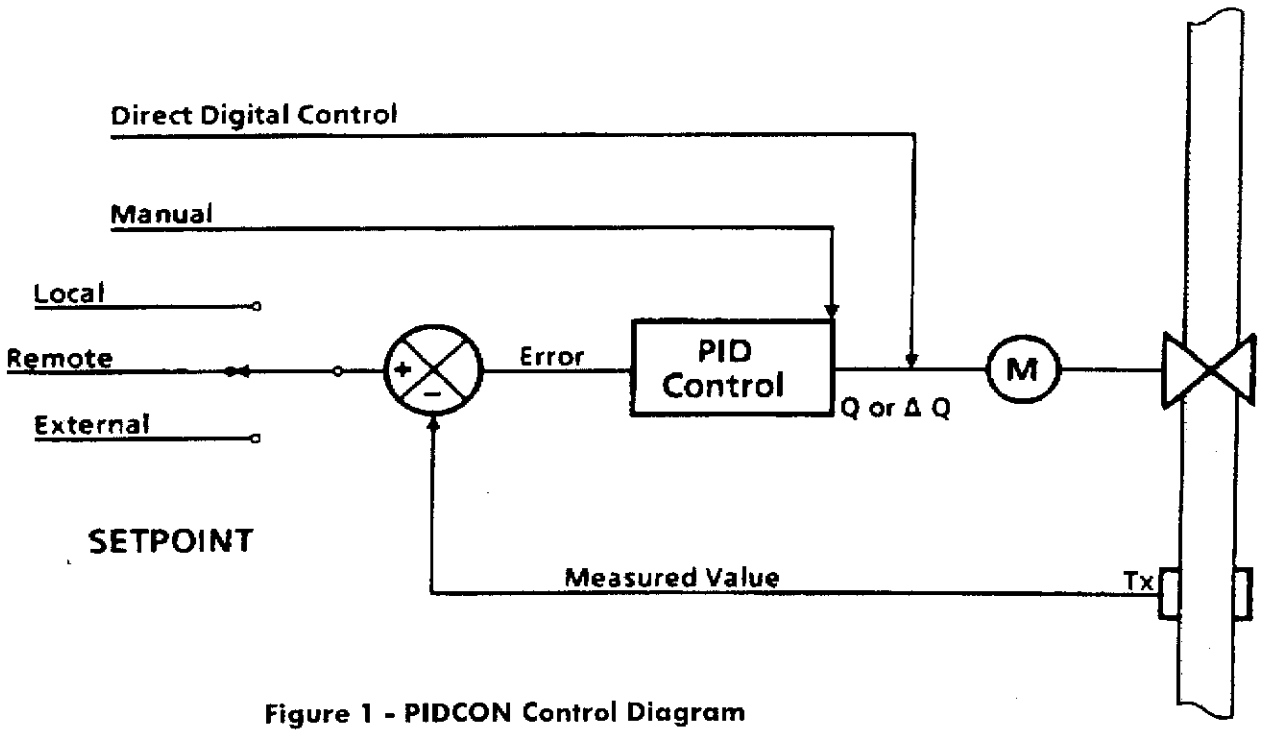


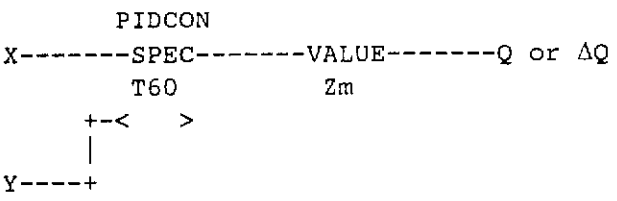
Figure 1 - PIDCON Control Diagram

2. FEATURES

- Independently programmable proportional, integral and derivative gains.
- Integral and derivative gain settings and rate limit settings are independent of execution interval
- Mode selection for greater flexibility.
- MANUAL and Direct Digital Control (DDC) modes enable the output to be raised or lowered by operator or logic intervention.
- REMOTE and LOCAL modes enable the setpoint to be taken from the rung or direct from manually alterable table locations.
- INTERNAL and EXTERNAL setpoint allow easy integration of control philosophies such as MIN/MAX.
- Choice between BASIC and INCREMENTAL equations, and between ABSOLUTE and INCREMENTAL output.
- Bumpless AUTO/MANUAL changeover
- Internal scaling of Remote Setpoint via user programmable RATIO and BIAS.
- DIRECT/REVERSE acting options.
- Derivative term can be made to operate on either the error or on the measured value only.
- Deadband facility enables small errors to be treated as zero.

3. SPECIFICATION

Control language:-



Basic Equation

(1) Absolute Output

$$Q_n = \frac{P}{100} E_n + \frac{D}{100t}\left(X_n - X_{n-1}\right) + \frac{It}{100} E_n + Q_I$$

(2) Incremental Output

$$\Delta Q_n = \frac{P}{100} + \frac{D}{100t} (X_n - X_{n-1}) + \frac{It}{100} E_n + Q_1 - Q_{n-1}$$

i.e.

$$\Delta Q_n = Q_n - Q_{n-1}$$

Incremental Equation

(1) Absolute Output

$$Q_n = Q_{n-1} + \frac{P}{100} (E_n - E_{n-1} + \frac{It}{100} E_n + \frac{D}{100t} X_n - 2X_{n-1} + X_{n-2})$$

i.e.

$$Q_n = Q_{n-1} + \Delta Q_n$$

(2) Incremental Output

$$\Delta Q_n = \frac{P}{100} (E_n - E_{n-1}) + \frac{It}{100} E_n + \frac{D}{100t} (X_n - 2X_{n-1} + X_{n-2})$$

-where:

Q_n	=	Output on scan 'n'
Q_{n-1}	=	Output on scan 'n - 1'
Q_I	=	The integral component of the output on scan 'n - 1'
E_n	=	Error on scan 'n'
X_n	=	Either error or measured value on scan 'n'
X_{n-1}	=	Either error or measured value on scan 'n - 1'
X_{n-2}	=	Either error or measured value on scan 'n - 2'
P	=	Proportional gain
D	=	Differential gain
I	=	Integral gain
t	=	Auto time dependent - Time since last execution(s) User time dependent - Time(s)

Table 1 Program Rung Data

Symbol	Location	Use	Range
X	Rung input (upper branch)	Remote setpoint	0 to 32767 No range checking to prevent negative number entry
Y	Rung input (lower branch)	Measured value	0 to 32767 No range checking to prevent negative number entry
Q	Rung output	Control output	-32768 to +32767 Subject to limits in value data
Zm	Value location	Start of parameter list - 30 locations	Any write-enable table except Q, useable by Special Functions

Table 2 Value Data

Symbol	Location	Use	Range
F	Zm	Fault Code	See Table 4
* MODEWORD	Zm+1	Bit selection of mode(s)	See Application Notes
MV	Zm+2	Measured value Y rung input to PIDCON	0 to 32767
WSP	Zm+3	The working setpoint is the setpoint after internal scaling	0 to 32767
** Q_n	Zm+4	Output on scan 'n'	-32768 to +32767 for incremental output. Subject to output limits for absolute.
* RATIO	Zm+5	Internal scaling factor for remote setpoint	0 to 32377 1000
* BIAS	Zm+6	Internal offset for remote setpoint	-32768 to +32767
* P	Zm+7	Proportional gain	0 to 32767 ¶ No range checking to prevent negative number entry
* I	Zm+8	Integral gain	0 to 32767 ¶ No range checking to prevent negative number entry
* D	Zm+9	Derivative gain	0 to 32767 No range checking to prevent negative number entry

Table 2 Value Data (Continued)

Symbol	Location	Use	Range
* DBAND	Zm+10	Deadband value. If error lies within ± this value, it is set to zero.	-32768 to +32767
RSP	Zm+11	Remote setpoint input to PIDCON	0 to 32767
* LOCAL	Zm+12	Local setpoint	0 to 32767
* DLIM	Zm+13	Deviation limit	-32768 to +32767
* MANUAL	Zm+14	Raise/Lower rate Fast = 10*Zm+14	-3200 to +3200
* LOLIM	Zm+15	Absolute output. Lower limit	-32768 to +32767
* HILIM	Zm+16	Absolute output. Upper limit	0 to 32767
* L1	Zm+17	Rate limit away from zero. Only applies when in AUTO mode, absolute output.	0 to 32767
* L2	Zm+18	Rate limit toward zero. Only applies when in AUTO mode, absolute output.	0 to 32767
E_n	Zm+19	Error on current scan (scan 'n')	-32768 to +32767
E_{n-1}	Zm+20	Error on last scan (scan 'n-1')	-32768 to +32767
E_{n-2}	Zm+21	Error on scan before last (scan 'n-2')	-32768 to +32767
ISP	Zm+22	Internal setpoint	0 to 32767
* T	Zm+23	Time since PIDCON last executed	0 to 65535 ms
Q_{n-1}	Zm+24	Output on scan 'n-1'	-32768 to +32767 for incremental output. Subject to output limits for absolute.
MV_{n-1}	Zm+25	MV on scan 'n-1'	0 to 32767
MV_{n-2}	Zm+26	MV on scan 'n-2'	0 to 32767
R_I	Zm+27	Integration reminder	0 to 32767
Q_I	Zm+28	Integral component of output last scan	0 to 32767 Subject to integral limit
T_L	Zm+29	Count at last call	-32768 to +32767

* Data set by user
** May be set by user while in DCC mode
¶ Corresponds to gains of 0 to 327.67 in steps of 0.01

4. APPLICATION NOTES

4.1. Mode Selection

To select modes of operation for PIDCON, it is necessary to set the states of individual bits in the MODEWORD (Zm+1). Details of these modes are given in Section 4.1.1 and in Table 3.

Table 3 MODEWORD

Bit	OFF	ON
0	Manual	Auto
1	Manual or Auto as set by bit 0	DDC
2	Setpoint Local	Setpoint Remote
3	Reverse Acting	Direct Acting
4	MV Derivative	Error Derivative
5	Basic Equation	Incremental Equation
6	Incremental Output	Absolute output
7	-	Raise
8	-	Lower
9	Slow	Fast
10	-	Derivative Scale
11	Not evaluate	Evaluate
12	User Time Dependent	Auto Time Dependent
13	Internal Setpoint	External Setpoint
14	Not used	
15	No fault	Bit set in fault word

Table 4 Fault Conditions and Flags, Zm

Bit Set	Value	Condition	Result
0	@1	Rate limit L1 exceeded	$Q_n = Q_{n-1} + \frac{L_{1I}}{100}$
1	@2	Rate limit L2 exceeded	$Q_n = Q_{n-1} + \frac{L_{2I}}{100}$
2	@4	Integral component of O/P limited at HILIM	Q_n = QP + QD + HILIM
3	@8	Integral component of O/P limited at LOLIM	Q_n = QP + QD - LOLIM
4	@10	O/P limited by LOLIM	Q_n = LOLIM
5	@20	O/P limited by HILIM	Q_n = HILIM
6	@40	Deviation limit	Indication only: Q_n calculated as normal
7	@80	WSP outside limits - too large RATIO or BIAS	Indication only: WSP limited at 32767
8	@100	Return from MANUAL	Indication only: Q_n calculated as normal
9	@200	General parameter error	$Q_n + Q_{n-1}$
10	@400	Incremental O/P overflow	Indication only
11	@800	Measured Value <0	Indication only: MV limited at 0

4.1.1. Operation Modes

4.1.1.1. MANUAL Mode

MANUAL mode is selected by default if neither bit 0 nor bit 1 have been set ON in the MODEWORD, Zm+1.

In MANUAL mode, the plant output can be raised or lowered:-

- To raise the output, set the RAISE bit, Zm+1 bit 7, ON in the MODEWORD
- To lower the output, set the LOWER bit, Zm+1 bit 8, ON in the MODEWORD

Using negative value will reverse the direction of RAISE and LOWER functions.

The output goes up or down each time the PIDCON is evaluated, by the value set in Zm+14. If the FAST bit, Zm+1 bit 9, is set ON in the MODEWORD, the output is raised or lowered ten times as fast, that is, by 10*(Zm+14) each time PIDCON is evaluated.

If neither the RAISE nor LOWER bits are ON, or if both the RAISE and LOWER bits are ON simultaneously, the output freezes at the value it had on the previous execution.

The integral, output and rate limits do not operate in MANUAL mode. However, the output is limited to the 100% output range, that is, 0 to 32767 in ABSOLUTE output and +100% (+32767 or -32768) in INCREMENTAL output mode.

The error history (Zm+19, Zm+20, Zm+21), measured value history (Zm+2, Zm+25, Zm+26)and output history, Zm+24, are updated. The integral component of output last scan, Zm+28, tracks the output, Zm+4, and the integration remainder is set to zero. These steps enable bumpless transfer to occur on reversion to AUTO mode regardless of the form of equation being evaluated. The return-from-manual bit in the FAULT WORD, Zm+8 is set on for indication only.

4.1.1.2. AUTO Mode

The AUTO mode is the normal operating mode of the PIDCON Special Function. To select AUTO mode, set bit 0 in the MODEWORD, Zm+1, ON.

When in AUTO mode, PIDCON calculates the controlled output to the plant. The equation used in this calculation, the type of output, and the action of the derivative term, etc., are controlled by a number of sub-modes. These are detailed further below.

4.1.1.3. Direct Digital Control (DDC) Mode

To select DDC mode, set bit 1 ON in the MODEWORD, Zm+1. The DDC overrides both AUTO and Manual modes. DDC mode can be selected, for instance, for overriding the PIDCON output via the PLC User Software under emergency conditions or when supervising the PIDCON operation from a supervisory computer.

In DDC mode, external logic can be used to write directly to the plant output location, Zm+4. As limits are not applied to this location in DDC mode, make sure that sensible values are written to Zm+4 under DDC mode.

The error history (Zm+19, Zm+20, Zm+21), measured value history (Zm+2, Zm+25, Zm+26), and output history, Zm+24, are updated. The integral component of output last scan, Zm+28, tracks the output, Zm+4, and the integration remainder is set to zero. These steps enable bumpless transfer to take place on reversion to AUTO or MANUAL mode, regardless of the equation being evaluated. However, careful consideration should be given to the effect of the proportional term.

4.1.2. Setpoint Control

Please refer to Figure 6 at the rear of this data sheet.

If INTERNAL setpoint is selected by setting bit 13 in the MODEWORD, Zm+1, OFF, there is a further choice between REMOTE and LOCAL setpoint. When INTERNAL setpoint is selected, the Internal Setpoint, Zm+22, is automatically copied into the Working Setpoint, Zm+3.

4.1.2.1. LOCAL Setpoint

If REMOTE setpoint is not selected, then by default, PIDCON uses the LOCAL setpoint from the Local Setpoint location, Zm+12. The Internal Setpoint, Zm+22, is set equal to the LOCAL Setpoint, without any scaling or offset. This value is then automatically copied into the Working Setpoint, Zm+3.

Note... When switching from LOCAL to REMOTE, bumpless transfer of the setpoint is not automatic. To make sure that this transition is bumpless, make PIDCON's REMOTE setpoint rung input track the LOCAL Setpoint, Zm+12. This can be achieved by inserting the following rung into the ladder program immediately after the PIDCON rung:

```
| MWRSP  LOCAL                  RSP      |
+---]/[----<AND>----------------<OUT>---+
| Zm+1.2  Zm+12       .                  |
```

The mnemonic RSP represents the table location from which PIDCON's X-rung input is taken.

4.1.2.2. REMOTE Setpoint

In AUTO, the normal operating mode is for the setpoint to come from the X-rung input. This is referred to as the REMOTE setpoint mode. To select the REMOTE setpoint, bit 2 in the MODEWORD, Zm+1, must be set ON.

The rung input value is copied into the Remote Setpoint location, Zm+11. It is then scaled by the RATIO, Zm+5, and offset by the BIAS, Zm+6, to form the Internal Setpoint, Zm+22, as detailed below, subject to the limits as detailed in Section 4.2.9.

$$(Zm+22) = \frac{(Zm+11) \times (Zm+5)}{1000} + (Zm+6)$$

The Internal Setpoint is then automatically copied into the Working Setpoint, Zm+3. While REMOTE setpoint is selected, the LOCAL Setpoint, Zm+12, tracks the Working Setpoint. This enables bumpless transfer to take place on switching from REMOTE to LOCAL mode.

4.1.2.3. EXTERNAL Setpoint

If EXTERNAL setpoint is selected by setting bit 13 ON in the MODEWORD, Zm+1, the setpoint is taken directly from the Working Setpoint location, Zm+3. It is then up to the user software to enter values in the range 0 to 32767 into this Working Setpoint location. The INTERNAL Setpoint, Zm+22, is calculated as normal but is not automatically copied into the Working Setpoint location. This facility allows control philosophies such as MIN/MAX. or Setpoint Ramping to be easily implemented.

Example 1

```
|                                                                  L/LIM   |
+---<AND>-------------------------------------------------------<OUT>---+
|    G5                                                            W1      |
|                                                                  H/LIM   |
+---<AND>-------------------------------------------------------<OUT>---+
|    G6                                                            W2      |
|    ISP      LIMIT                                                        |
+---<AND>-----SPEC------VALUE-------------------------------<OUT>---+
|   Zm+22     S32         W0                                       W5      |
|                                                                  WSP     |
+---<AND>-----] [---------+-------------------------------<OUT>---+
|    G5       W0.10       |                                        Zm+3    |
|                         |
+---<AND>-----] [---------+
|    G6       W0.11       |
|    WSP                  |
+---<AND>-----] [----] [--+
|   Zm+3      W0.10  W0.11
```

where Zm is the parameter list for the PIDCON Special Function.

In the above example, parameters G5 and G6 would be set earlier in the user software. If a setpoint is entered which is outside these limits it would be set equal to one of the limits.

Example 2

which would provide a Ramp on any new setpoint entered.

4.1.3. REVERSE ACTING/DIRECT ACTING

When bit 3 of the MODEWORD, Zm+1, is set ON, PIDCON operates in REVERSE ACTING mode. In this mode, the output to the plant, Zm+4, decreases as the error increases.

If bit 3 of the MODEWORD, Zm+1, is left OFF, PIDCON defaults to the DIRECT ACTING mode. In this mode, the output to the plant, Zm+4, increases as the error increases.

Note... The DIRECT ACTING/REVERSE ACTING bit affects the output in AUTO mode only. In MANUAL mode, the output is always raised or lowered directly.

4.1.4. DERIVATIVE ACTION Modes

4.1.4.1. MV DERIVATIVE/ERROR DERIVATIVE MODES

PIDCON can be set to operate either in ERROR DERIVATIVE mode or in MV DERIVATIVE mode, using bit 4 of the MODEWORD, Zm + 1, see Table 3. If not set, PIDCON defaults to MV DERIVATIVE mode.

As the names imply, PIDCON calculates its derivative term from the rate of change of error in the ERROR DERIVATIVE mode and from the rate of change of measured value in the MV DERIVATIVE mode.

Use of the MV DERIVATIVE mode reduces the size of transient output changes, compared with the ERROR DERIVATIVE mode, when there is a rapid change of setpoint. However, the ERROR DERIVATIVE mode may be more appropriate for the inner loop of cascade controls.

Note... A genuine step change of setpoint in the ERROR DERIVATIVE mode causes a transient PIDCON output for one execution only. If the derivative gain is a reasonable size, this spike is clipped by the appropriate output limit, and the derivative action is largely lost.

To prevent this occurring, try to avoid step setpoint changes. Use for instance, RAMPGEN to smooth out the changes, or the rate limits on a previous function providing the setpoint. Paradoxically, this may give a faster output response even though the setpoint input signal is changing more slowly.

4.1.4.2. Derivative Scale

The PIDCON Special Function calculates its gains as follows:

$$\text{Proportional Gain} = \frac{P}{100}$$

where P = (Zm+7)

$$\text{Integral Gain} = \frac{I \times t}{100{,}000}$$

where I = (Zm+8)
t = Time in ms since last execution, (Zm+23)

$$\text{Differential Gain} = \frac{D \times 10}{t}$$

Where D = (Zm+9)
t = Time in ms since last execution, (Zm+23)

If Derivative Scale mode is selected, by setting bit 10 of the MODEWORD Zm+1 ON, the Derivative Gain is calculated as above but divided by 10, that is

$$\text{Differential Gain} = \frac{D}{t}$$

4.1.5. BASIC EQUATION/INCREMENTAL EQUATION Modes

When bit 5 of the MODEWORD Zm+1 is set ON, PIDCON operates in the INCREMENTAL EQUATION mode and evaluates the incremental equation (given in Section 3, Specification) to produce its output (again see Section 3).

The BASIC EQUATION mode must always be used if an integral term is not used, that is, if the control is proportional only or proportional + derivative.

When an integral term is included, both modes give exactly the same response as long as the output does not enter limit. However, if the output enters limit, the BASIC EQUATION mode gives a faster response, whereas the INCREMENTAL EQUATION mode gives a response which tends not to overshoot.

4.1.6. OUTPUT Modes

INCREMENTAL OUTPUT mode must be used where, regardless of the value of the setpoint, the control signal to the plant is always of the same value under steady state conditions. This is referred to as the 'home' value of the control signal. With this type of system, the control signal transiently moves away from its 'home' value when the setpoint is changed, but settles back at its 'home' value when steady conditions have been achieved.

ABSOLUTE OUTPUT mode must be used where, under steady state conditions, the control signal to the plant is higher when the setpoint is higher, and lower when the setpoint is lower. In other words, use ABSOLUTE OUTPUT mode where the control signal is roughly proportional to the setpoint.

Examples of a Flow Control Loop

Consider the closed loop control of fluid flow, but with valves having different types of actuators.

Example 1

Valve actuator input in the range 4-20mA, where:

4mA	=	valve fully closed
20mA	=	valve fully open

For a larger flow, a larger control signal is needed, that is, the control signal must increase as the flow setpoint is increased. For this type of valve actuator, PIDCON must be used in ABSOLUTE OUTPUT mode.

Example 2

Valve actuator input in the range 4-20mA, where:

4mA	=	valve closing at maximum rate
20mA	=	valve opening at maximum rate
12mA	=	valve stationary (remains at whatever opening it has reached)

Under steady state conditions, the valve must remain stationary, regardless of the flow required. Valve movements are necessary only when the flow is to be changed, but once the change is complete, the valve must remain at its new opening. The control signal must therefore always return to its 'home' value of 12mA, regardless of the flow setpoint. For this type of valve actuator, PIDCON must be used in the INCREMENTAL OUTPUT mode.

For most applications of INCREMENTAL OUTPUT mode, the 'home' output value is zero, e.g. where the plant signal swings between -10V and +10V. The example given above is to make clear that this is not always so and not a requirement for PIDCON. The INCREMENTAL OUTPUT mode is most often used for driving motor-powered actuators of the type that have no built-in positioning servos, and which therefore require zero input for any steady-state position. It may also be used where the plant provides integral action itself.

4.1.6.1. INCREMENTAL OUTPUT

If bit 6 of the MODEWORD, Zm+1 is left OFF, PIDCON defaults to INCREMENTAL OUTPUT mode. In this mode, PIDCON outputs the change in what the ABSOLUTE OUTPUT would have been since the last scan, from whichever equation has been selected. In INCREMENTAL OUTPUT mode, HILIM, Zm+16, and LOLIM, Zm+15, are ignored and the output of PIDCON is subject only to the normal number limits of ±32767.

4.1.6.2. ABSOLUTE OUTPUT

When bit 6 of the MODEWORD, Zm+1, is set ON, PIDCON operates in the ABSOLUTE OUTPUT mode. In this mode, PIDCON provides an absolute output from whichever equation has been selected.

If on return from MANUAL, bit 8 of FAULTWORD is not set, then the output is limited by the HILIM and LOLIM, Zm+16 and Zm+17. If bit 8 of FAULTWORD is set, the ABSOLUTE OUTPUT is limited by HILIM only. If the ABSOLUTE OUTPUT limits have not been exceeded, the output is also limited by the rate limits L_1 and L_2 set by Zm+17 and Zm+18 where necessary. See Section 4.2 for further details of limits.

4.1.6.3. Contactor Outputs

In some applications, the output of a closed loop control may be necessary to switch contactors on and off with a variable on-to-off time ratio, e.g. for controlling electric heating elements.

For such applications, PIDCON must be used in ABSOLUTE OUTPUT mode but with its output feeding a variable mark-space generator. The example below shows a possible solution. In this section of program, the output from PIDCON is taken into W30, and has been arranged to lie in the range 0 to 1000.

```
|             ONTIME                                          OFFTIME   |
+---<AND>-----<SUB>-----------------------------------------<OUT>--+
|    1000        W30                                          W31    |
|                                                                    |
|  RESET              OFFTIME                                 OUTPUT |
+---]/[-----DELAY-----VALUE-------------------------------------( )--+
|  G60.0      G61        W31                                   G60.1  |
|                                                                    |
|  OUTPUT             ONTIME                                  RESET  |
+---] [-----DELAY-----VALUE-------------------------------------( )---+
|    G60.1    G62        W30                                   G60.0  |
```

The first rung is designed so that there is a constant cycle time of 100 seconds, i.e. 1000 units of 0.1 seconds. The output from PIDCON into W30 sets the on-time; subtracting this from 1000 gives the required off-time. The second rung, with G60.0 initially de-energised, provides an output after a delay equal to the off-time. The output then initiates the on-time delay in the third rung, and the output from G60.1 remains on until this delay has timed out, after which G60.0 resets the off-time delay in the second rung, G60.1 de-energising resets the on-time delay in the third rung, and the sequence re-cycles.

Note... A section of program such as the above should operate on every program scan, and should not be put in a Block operating at a slower repetition interval, so that the DELAYs operate correctly.

4.1.7. Evaluation

When bit 11 of the MODEWORD, Zm+1 is set ON, PIDCON is in the EVALUATE mode. In this mode the function is evaluated, output last scan, Zm+14, is updated and elapsed time since last evaluation, Zm+23, is reset to zero, irrespective of any other mode settings.

When bit 11 of the MODEWORD is set OFF, PIDCON is in the NOT EVALUATE mode. In this mode the output is set to the output on the previous scan and elapsed time since last execution, Zm+23, is not reset. Therefore set bit 11 OFF in the MODEWORD, under conditions where it is required to accumulate elapsed time in Zm+23. If it is required to force the output to freeze at some level and time not to be accumulated in Zm+23, set bit 11 and bit 1 ON in the MODEWORD, to put the function into DDC mode.

4.1.8. TIME DEPENDENT

When bit 12 of the MODEWORD, Zm+1, is set ON, the PIDCON operates in the AUTO TIME DEPENDENT mode. In this mode any user input to Zm+23 is ignored and Zm+23 is automatically calculated using the contents of the host controllers data table E8 and the Count at Last Call, Zm+29. The location Zm+23 is always reset to zero after each evaluation in AUTO mode irrespective of the status of the AUTO TIME DEPENDENT/USER TIME DEPENDENT bit 12 of the MODEWORD. In the AUTO TIME DEPENDENT mode, care must be taken to ensure that the Time Since Last Execution does not exceed 65.536 seconds, otherwise spurious values for the I and D terms may be calculated.

When bit 12 of the MODEWORD is set OFF, PIDCON operates in the USER TIME DEPENDENT mode. In this mode the user input parameter, elapsed time since last evaluation, Zm+23, is used to scale the integral and derivative gains internally, the value in Zm+23 being interpreted as units of milliseconds. In this mode a special case exists when the time since last evaluation is zero; when the Incremental equation is selected, the equation is evaluated normally, but with the internal integral and derivative gains set to zero. When the Basic equation is selected, the equation is not calculated and the output is set to the output of the previous scan.

Note... When operating in the User Time Dependent mode, the Time at last Call, Zm+29, continues to be updated. This allows the function to be easily switched back into Auto Time Dependent Mode.

When operating in DDC mode, the Time Since Last Execution, Zm+23, continues to be calculated as normal.

4.2. Limits and Faults

4.2.1. Fault Word Indication

If a fault is detected:

- bit 15 of the MODEWORD, Zm+1, is set ON
- a specific bit of the FAULT WORD, Zm, is set ON, see Table 4

In this context, a 'fault' means anything that prevents PIDCON evaluating its output normally. For instance, the term 'fault' includes the output going into limit, which is likely to happen during normal operation. The term 'fault' therefore does not necessarily mean an operational fault condition.

Bit 15 of the MODEWORD is set OFF at the beginning of each execution of PIDCON. If bit 15 of the MODEWORD is OFF on exit from PIDCON, no faults were detected on that scan. The only circumstance when this does not occur is when the Return-from-Manual bit, Zm bit 8, has been set ON in the FAULT WORD, Zm, after evaluating PIDCON in the MANUAL mode.

Table 4 shows the meaning of particular bits of the FAULT WORD, Zm, being ON and what output PIDCON gives under fault conditions. If bit 9 in the FAULT WORD, Zm, is ON, at least one of the input data table values is outside the allowed range, see section 4.2.7. In this case, PIDCON defaults to the NOT EVALUATE mode, and the rung remains at the output of the previous scan, Zm+24. The Time Since Last Execution, Zm+23, continues to be accumulated.

4.2.2. Range Limits

The Working Setpoint value, Zm+3, is always subject to its full range limits, 0 to 32767. While the Working Setpoint value is in limit, bit 7 is set ON in the FAULT WORD, Zm for the purpose of indication only.

4.2.3. Rate Limits

When PIDCON is in the AUTO mode and ABSOLUTE OUTPUT mode is selected, the output is subject to rate limits.

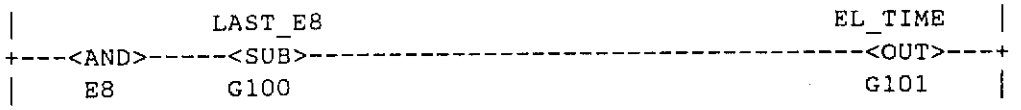
If the output is moving away from zero and the rate-limit-away-from-zero set by L_1 is exceeded, the output is set to the output on the previous scan, plus the rate-limit-away-from-zero term, that is:

(Zm+4) = (Zm=24) + (Zm+17)

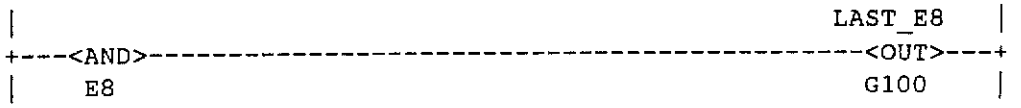
Bit 0 in the FAULT WORD, Zm, is set ON to indicate the fault.

If the output is moving toward zero and the rate-limit-toward-zero set by L_2 is exceeded, the example ladder below will allow the output to be set to the output of the previous scan minus the rate-limit-toward-zero term.

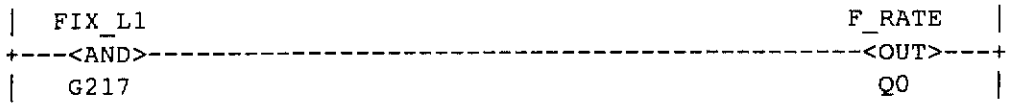
Calculate time since last execution by subtracting last value of E8 from the current value.



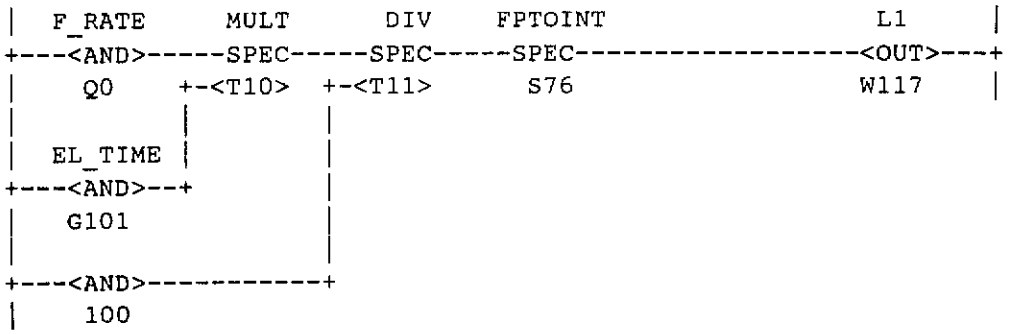
Save the current value of E8 ready for next time.



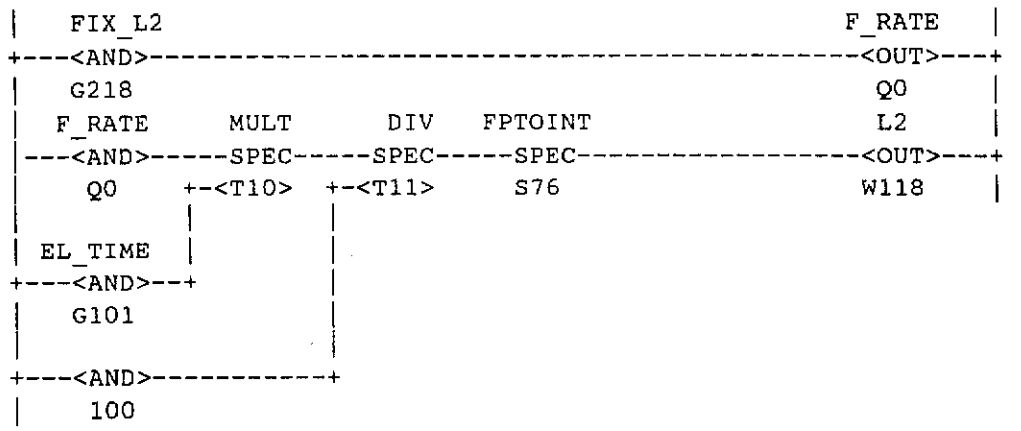
Convert the non time-dependent rate limit value into a floating point number.



calculate the time-dependent rate limit by multiplying by elapsed time (in ms) and dividing by 100, then convert the result back to integer.



Now do the same for the rate limit away from zero.



that is:(Zm+4) = (Zm+24) - (Zm+18)

Bit 1 in the FAULT WORD is set ON to indicate the fault.

4.2.4. Integral Component Limits

When PIDCON is in the AUTO mode and the BASIC EQUATION mode is selected, the integral component of the output is subject to the same range limits as the output itself, (Zm+15, Zm+16) as long as the return-from-manual bit in the FAULT WORD, Zm bit 8, is not set.

If the integral component of the output lies within the bounds specified by Zm+15 and Zm+16, the return-from-manual bit is set OFF if it was previously ON. This sequence provides integral de-saturation, otherwise known as anti-reset windup.

4.2.5. Output Limits

4.2.5.1. ABSOLUTE Output Limit

When PIDCON is in the AUTO mode its output is subject to output limits and output rate limits. If the output would be greater than HILIM, Zm+16, then:

- The output, Zm+4, is set equal to HILIM
- Bit 5 in the FAULT WORD, Zm, is set ON for indication

If the output would be less than LOLIM, Zm+15, then:

- The output, Zm+4 is set equal to LOLIM
- Bit 4 in the FAULT WORD, Zm, is set ON for indication

Rate limits are dealt with in Section 4.2.3. When PIDCON is in the MANUAL mode, its output is subject to a number limit of 0 to 32767.

A closed loop system only remains in closed loop mode operationally as long as the plant responds to what the control signal demands of it. However, it is possible that, with a well tuned 3-term controller, the controller may demand more from the plant than it is capable of. For instance, suppose the signal levels have been chosen so that the output from a PIDCON function is in the range 0 to 10000 (giving numbers which can be easily interpreted when monitoring as % output . Suppose that the PIDCON output has been scaled with a LINCON and fed to an analogue output module so as to give a 4 to 20mA output to the plant corresponding to an output of 0 to 10000 from PIDCON.

When the setpoint is changed, the output of PIDCON could go up to perhaps 15000. However, it is unlikely that the analogue output module would provide 30mA and, even if it could, the increased signal might not have any effect on, for example, a valve actuator. Once the valve is fully open, any further increase in signal does not make any difference.

In such cases, if limits are not used, any large change in setpoint can cause undesirable overshoots in the measured value. This is because of the time taken for the output of PIDCON to return to the working range of values. Until this happens, the control signal to the plant remains at its maximum value or above, and the measured value is driven too high.

Similar effects can occur, to a lesser extent, if rate limits are not used when in the ABSOLUTE OUTPUT mode.

Therefore, limits should be set according to the following rules:

(a) Set limits on the 3-term control function itself, and not on any subsequent functions operating on the control signal, such as LINCON.

(b) Choose limit values so that following hardware or software does not saturate. That is, any hardware may be driven to its limit but no harder, and software program functions must not be driven to maximum or limit number values.

In the ABSOLUTE OUTPUT mode, rate limits should be set so that the rates of change of the control signal are not faster than the plant can follow. The table values L_1 and L_2 determine how fast the control signal can change.

4.2.5.2. INCREMENTAL Output Limit

When PIDCON is in the AUTO mode, its output is subject to the number limits of -32768 to +32767. When the output is at this limit, it is indicated by bit 10 in the FAULT WORD, Zm, being set ON.

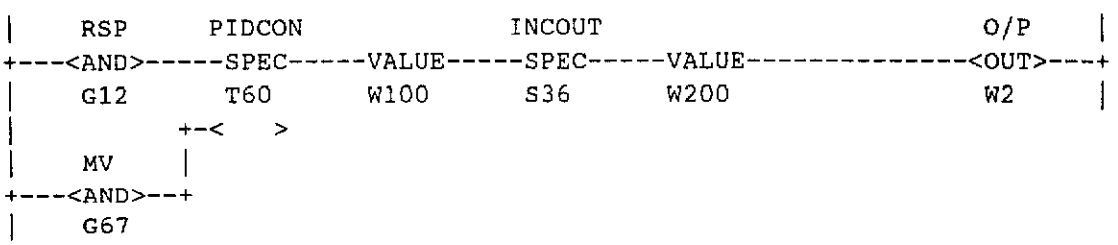
When the PIDCON is in the MANUAL mode, its output is subject to the same number limits of -32768 to +32767.

INCREMENTAL OUTPUT mode, as explained in section 4.1.8, is intended for use with actuators where the control signal always returns to a 'home' value under steady state conditions.. Such actuators, in fact, integrate the difference between the control signal and its 'home' value. INCREMENTAL output is normally used where its 'home' output value is zero.

At first sight it may appear that the same rules for limit settings should be applied as for the ABSOLUTE OUTPUT mode. That is, that limits should be set so as to prevent the control signal trying to drive the actuator faster than it can respond. This may be satisfactory in practice with an actuator which has very little friction and stiction.

However, with many actuators, if the setpoint to the control loop is changed or if there is a disturbance affecting the measured value, the system may come to a steady state condition. In this case PIDCON, set for INCREMENTAL OUTPUT, gives an output too small to overcome friction, with the actuator having stopped short. Proper reset action is then lost and the actuator could overheat if left standing with a steady current flowing through it.

To overcome any such problem, the Special Function INCOUT could be added, following PIDCON, to the ladder diagram. See also section 4.2.10, e.g.



4.2.6. Deviation Limit

If the magnitude of the calculated ERROR, Zm+19, is greater than the DLIM limit, Zm+13, then bit 6 in the FAULT WORD is set ON to indicate the closed loop control is operating outside specified tolerance and ladder code should be used to take corrective action. If DLIM is negative, the error will always be greater than the deviation limit.

4.2.7. Data Errors

In addition to faults caused by the output of PIDCON being in one of the limits given above, there are other fault conditions which prevent PIDCON from being able to evaluate its output normally.

If bit 9 in the FAULT WORD is ON, it indicates that one of the user input parameters is outside the permitted range. When this fault is detected, PIDCON defaults to the NOT EVALUATE mode. In this case, the output, Zm+4 is set to the output of the previous scan, Zm+24, but the time since last execution, Zm+23, continues to be accumulated.

The following conditions lead to FAULT WORD bit Zm bit 9 being set:-

Rung Input (Remote Setpoint)	X	<0
Ratio	Zm+5	<0
Local setpoint	Zm+12	<0
Manual raise/lower	Zm+14	>3200
LOLIM	Zm+15	<0
HILIM	Zm+16	<0
Rate limit	Zm+17	<0
Rate limit	Zm+19	<0
Elapsed time	Zm+23	<0

Note... The FAULT WORD is cleared at the start of each scan, with the exception of the return-from-manual bit, which remains set if previously set

4.2.8. Measured-Value-Out-of-Range Error

If the rung input Y, measured value, is less than zero, the PIDCON Special Function responds by setting bit 11 in the FAULT WORD, see Table 4. The MV data table location, Zm+2, is set to zero and the PIDCON Special Function operates normally with this zero measured value.

Note... It is up to the user software to detect this bit and take appropriate action, such as putting the Special Function into DDC mode and holding the present value of output or suiciding the output. Such a fault may indicate the failure of the MV transducer or perhaps a wiring fault.

4.2.9. Working Setpoint Limit

If the PIDCON Special Function is selected to work with a Remote Setpoint, the Working Setpoint, Zm+3, is calculated as shown in Section 4.1.2.1. The result of this calculation may fall outside the range 0 to 32767 and under these circumstances, the following actions are taken.

1. WSP >32767

The Working Setpoint is set equal to 32767 and bit 7 of the FAULT WORD is set ON.

2. WSP <0

The Working Setpoint is set equal to zero and bit 7 of the FAULT WORD is set ON. This fault flag is for indication only and PIDCON output continues to be calculated as normal.

4.2.10.Characteristics of INCOUT

The INCOUT function only gives outputs which lie between upper and lower limits of windows, symmetrical about zero. The upper limits operate as normal limits; the lower limits provide a dead zone. The dead zone should be set so that signals smaller than any to which the actuator can respond are blocked.

However, the chief property of INCOUT is that it gives an output whose integral is equal to that of the input from PIDCON, subject to the output being within the limit window. This is more easily seen in the diagram below:

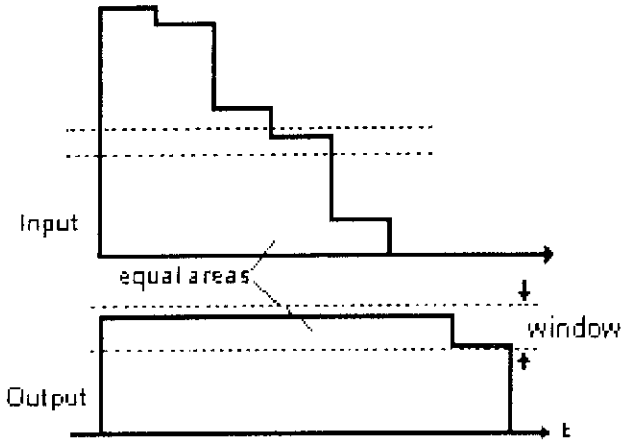


Figure 2 Characteristics of INCOUT

Note... If the input area had been fractionally less, the last output value on the last program scan shown might have fallen below the lower limit window. In this case, the actual output on this scan would have been zero. The value not outputted would be held over until PIDCON provided some further input to INCOUT

INCREMENTAL OUTPUT can be checked using the incremental equation given in section 3. On a step change of setpoint, PIDCON normally gives a large positive output on the first program repetition interval, followed by a large negative output on the second interval. INCOUT enables these two 'spikes' to be used in the output to the plant without being chopped off by limits or being disregarded by the actuator.

4.3. Operation of Deadband Facility

If a small error persists due to external effects, the control loop may hunt. This is more likely on integral control, but is also possible even without integral control due to quantization errors in the DAC on the output, and in the ADC on the measured value. To prevent this, PIDCON includes a deadband facility, allowing a value DBAND, Zm+10, to be set, which operates on the error signal as follows:

If | error | <= DBAND then : error = 0

If | error | > DBAND then : error - DBAND

If | error | <- DBAND then : error + DBAND

The DBAND parameter only works properly if positive values are used, care should be taken to ensure no negative values are entered since the parameter is not range checked.

4.4. Bumpless Transfer

Both the BASIC and INCREMENTAL equations allow bumpless changeover to closed loop control. Returning from LOCAL to REMOTE is the only circumstance in which bumpless transfer does NOT take place, see section 4.1.2.1.

4.5. Initialisation

PIDCON automatically adjusts the sizes of the integral and derivative gains, taking into account the time since it was last executed. To do this, PIDCON uses table location Zm+29 to store what the time was on its last execution. The next time it is executed, PIDCON reads the time from the host controller's data table E8, and uses Zm+29 to find the time interval since last execution, Zm+23.

The time since last execution is used in PIDCON's calculations for the output as shown by the equations in section 3. It is also used for rate limit calculations.

This technique makes PIDCON easy to program, as it is not necessary to know anything about program cycle times, and it doesn't matter if PIDCON is operated in a Block. However, it does mean that:

- PIDCON must be initialised
- PIDCON must not be allowed to run for more than 65 seconds without being operated, otherwise it needs to be re-initialised.

4.5.1. First Execution

When the program first runs, Zm+29 may contain zero, or a time value remembered from when the controller was last powered up. When PIDCON operates, it can therefore calculate a spurious value for the time since last execution, and set extremely large gains.

Consequently, PIDCON output should be suppressed the first time it operates. This can be done by setting the EVALUATE/NOT EVALUATE bit 2 in the MODEWORD, Zm+1, OFF. PIDCON then only reads the time, and its output remains unchanged. This bit can be set ON for subsequent program cycles.

4.5.2. Re-initialisation

If there is a PIDCON Special Function in a Block which only executes for certain phases of plant operation, it is quite possible that PIDCON may not have been operated during the previous 65 seconds. In this case, a spurious value of PIDCON output could be caused, as the time read from the controller could have been 'once round the clock' or more.

In this case, initialisation must be included in the Block so that PIDCON operates the first time with the output not evaluated but held, and is only subsequently released.

4.5.3. Single Cycle Operation

For test purposes, PIDCON may be required to operate in a single cycle mode, so that the output changes can be observed, one execution at a time. When a controller containing PIDCON is set to single cycle mode, the time is artificially incremented by 100ms on each operation, rather than operating in real time.

5. PRINCIPLES OF CLOSED LOOP CONTROL

If PIDCON is being used it is assumed that the user is familiar with closed loop control theory. If not, the manual 'Closed Loop Control Using GEM80 Programmable Controllers', Publication No. T451, can be purchased from CEGELEC INDUSTRIAL CONTROLS, or refer to any standard textbook on the subject. However, the following practical points should be borne in mind:

5.1. Accuracy

Steady state accuracy of a closed loop control is the sum of the accuracies of the following:

- Transducer converting the measured variable into an electrical signal
- Any signal conditioning equipment, scaling and/or isolating this signal
- Analogue to digital conversion of the measured value
- Any scale factor or linearization calculations on the measured value

Note... Inaccuracies in the control signal to the plant do not enter into the system accuracy calculations. The resolution of this control signal (the smallest possible change in it) can, however, be important. This topic is dealt with later.

5.2. Response

The response depends on:

- Characteristics of the plant
- Response of the measuring transducer
- Placement of the measuring transducer
- Analogue and digital filtering

Note... If, for example, a temperature transducer is placed a long distance downstream of where the heat is fed in on a temperature control, any change of heat input may not be sensed by the transducer for several seconds. Therefore a fast control loop becomes impossible

5.3. Signal Levels

The control signal to the plant, and the measured value, are normally both analogue signals. However, within the GEM80 controller, both signals are treated as numbers.

When programming a GEM80 controller, it may be tempting to convert signals into numbers which are directly meaningful for monitoring, e.g. to convert a signal from a thermocouple into a number equal to the temperature in °C or °F. However, this may reduce the resolution obtainable with the control loop, and may also cause a dither type of fluctuation.

For instance, if a temperature in the range of 0 to 100°C is converted into a number in the range 0 to 100, the system is only capable of setting the temperature in increments of 1°C. If instead, the temperature is converted into a number in the range 0 to 10000, the temperature can be controlled in 0.01° increments.

Similarly, if there is a valve actuator with a positioner, and if a 3-term control function that provides the control signal to it is programmed with the range of the output number only 0 to 100, the valve can only be set to discrete positions 1% apart.

It may well be that, to maintain the measured value equal to the setpoint, the valve should be at a position halfway between two steps. In this case, the controller causes the valve to dither up and down 1% so as to obtain the correct mean setting.

The numbers used for the setpoint, measured value and control signal should therefore be sufficient to give the resolution required. Where it is necessary to provide 'real values' for monitoring purposes, it is preferable to use simple multiples, e.g. 10 or 100 times. If this is not feasible, separate rungs in the program should be used, specially for monitoring.

5.4. Program Repetition Interval

To achieve the fastest response, the GEM80 program could be arranged so that PIDCON updates the value of the control signal to the plant on every cycle through the program.

However, it is usually undesirable to update the control signal more frequently than necessary, because PIDCON involves several internal calculations, and consequently takes longer to perform than most other functions. Refer to the execution times given in the Technical Manual for specific models of GEM80 controller.

Therefore, it is recommended that the sections of program containing PIDCON are included within BLOCKs, initiated at regular intervals, where the time interval is such as to make only a marginal difference to the closed loop response. The illustration below indicates the effect of the control signal being updated only at the repetition interval as shown below:

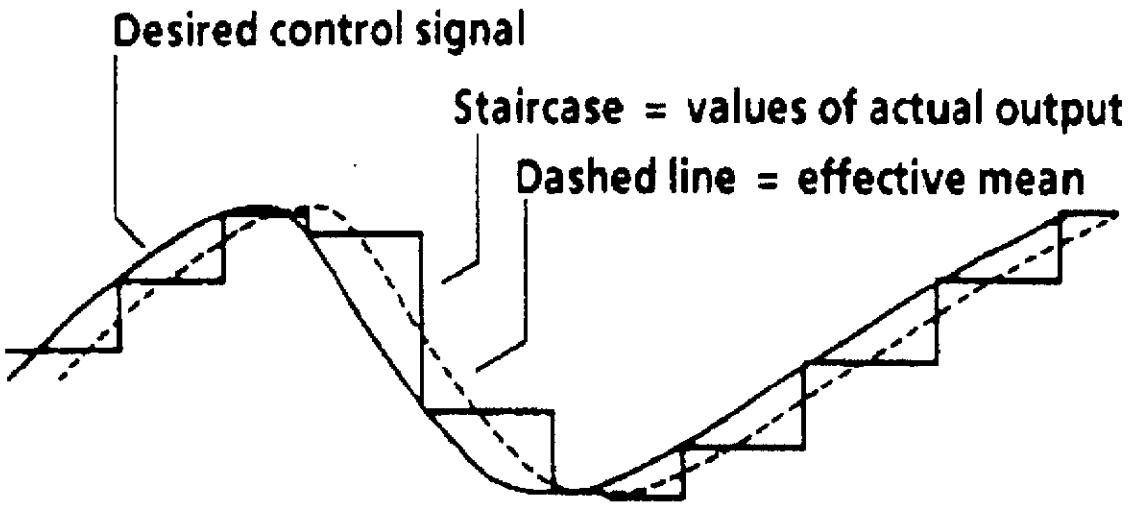
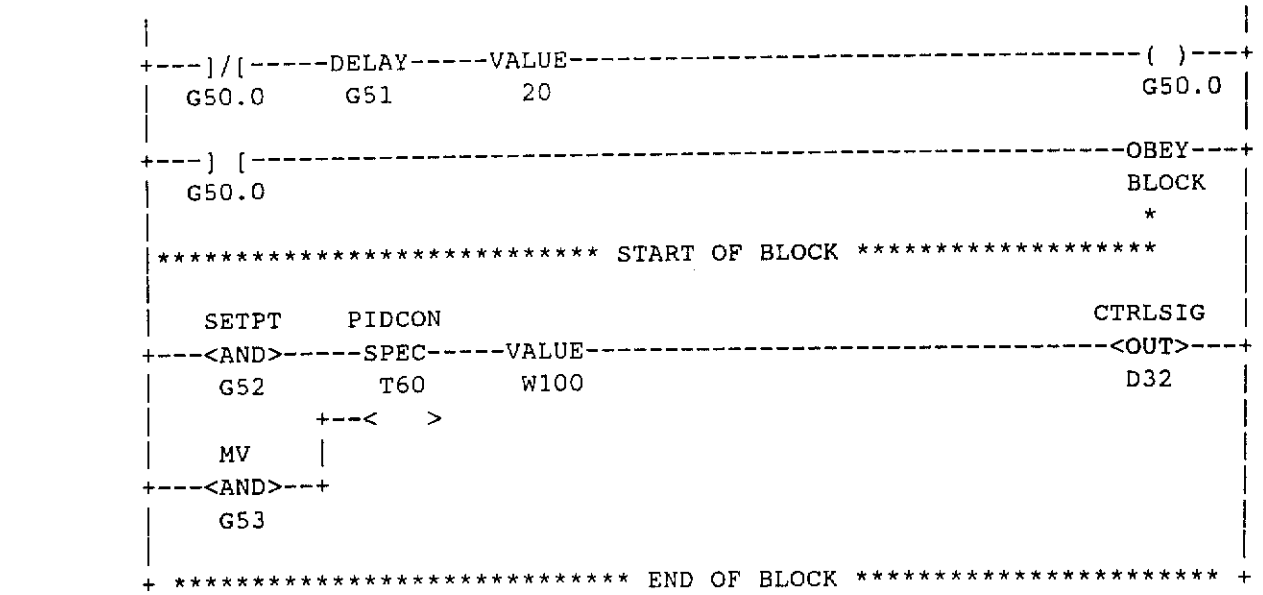


Figure 3 -Control Signal Updated at Repetition Interval

Effectively, the control signal to the plant is delayed by half the repetition interval.

5.4.1. Time Dependent BLOCK Example

Using the DELAY function, PIDCON can be put in a BLOCK as in the following example:



In the example above, the coil G50.0 energises after a delay of 2 seconds (20 units of 0.1s). However, it operates for one program cycle only, as the contact of G50.0 in the same rung resets the DELAY on the next program execution. De-energising G50.0 in the next rung therefore causes the BLOCK containing PIDCON to be operated at 2-second intervals, and to be skipped on the other program cycles while the DELAY is timing out.

As the repetition interval for the BLOCK depends only on the VALUE set for the delay, it does not matter if the configuration data in the P-table of the controller is subsequently altered for the program cycle time; the BLOCK is still initiated at 2-second intervals.

5.4.2. Programs Containing Several 3-term Controls

If there is a system containing several 3-term controls, in order not to slow down the program cycle time too much, different control loop BLOCKs need to be initiated on different cycles through the program. This can be programmed easily using the SEQR (Sequencer) function. If all the control loops have to operate at the same repetition interval this is straightforward, but, if the loops are required to operate at different intervals, a timing diagram should be drawn.

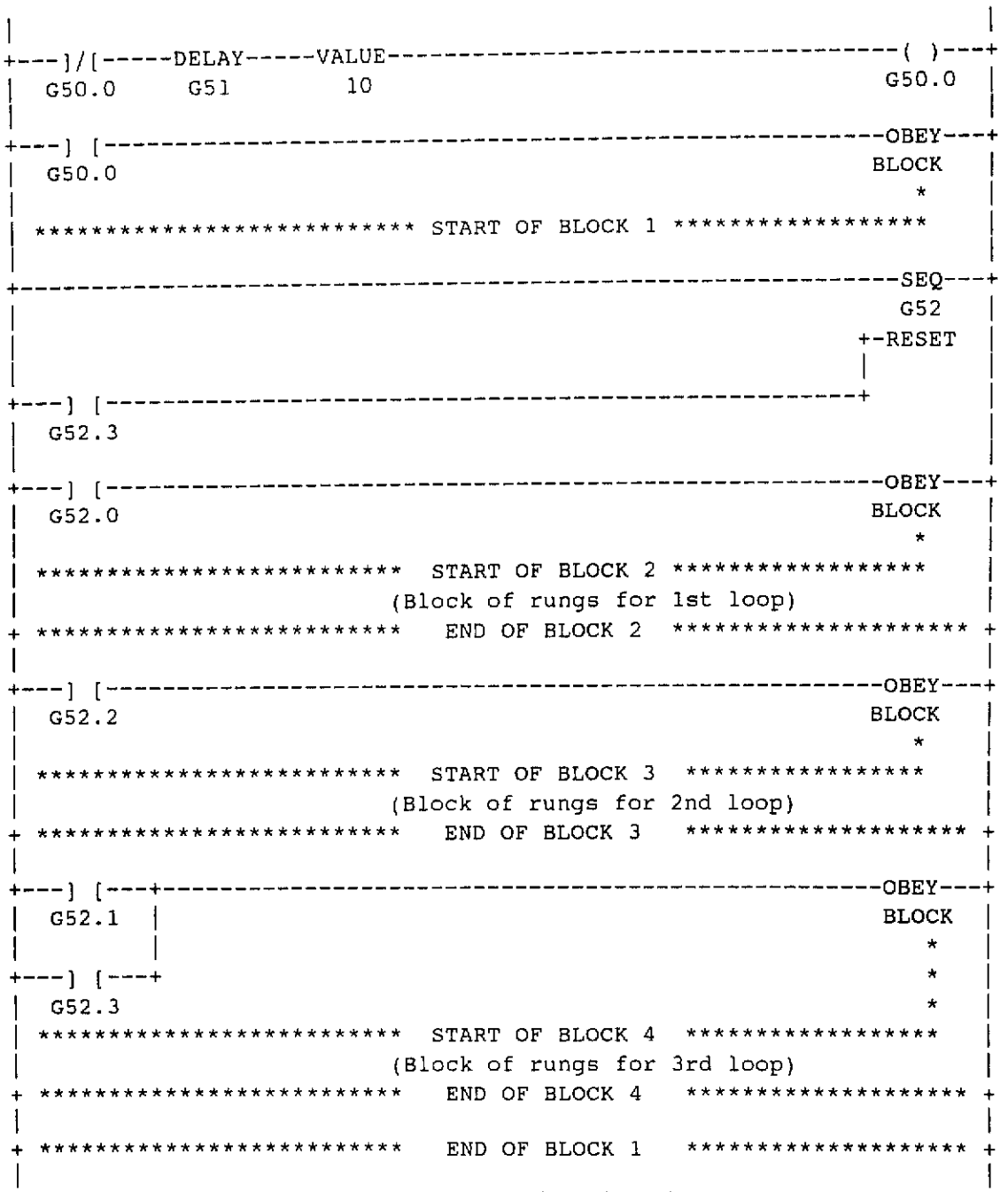
Example

Assume 3 loops. Two loops can be operated at a 4-second repetition interval, but the remaining loop is required to operate at 2-second intervals.

One possible solution would be to arrange the operation of the loops as follows:

Time (seconds)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1st loop	X				X				X				X		
2nd loop			X				X				X				X
3rd loop		X		X		X		X		X		X		X	

In this timing diagram only one loop operates each second. It also shows that the cycle of operations repeats every 4 seconds. Consequently, a rung is needed with a 1-second delay, operating into a Sequencer with 4 steps, after which the Sequencer must be reset so that the cycle of operations can repeat. A suitable ladder diagram would be:



The above section of program is not the only possible method of programming the given set of loops, but probably the neatest.

Note... It is essential that each loop operates at a regular interval, and that the following alternative timing diagram would be wrong because it would invalidate this rule for the 3rd loop.

Time (seconds)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
1st loop	X				X				X				X			
2nd loop		X				X				X				X		etc.
3rd loop			X	X			X	X			X	X			X	

Instead of operating at regular 2-second intervals, as in the first timing diagram, the 3rd loop would operate at intervals of 1-second and 3-seconds alternatively, which would result in a somewhat unpredictable performance.

Note...

1. Other methods of initiating loops at regular intervals are possible, using the timing markers E0.0 or E0.1. However, there may be problems with trying to count timing markers, depending on what the program cycle time is. See GEM80 Programming Manual, Publication No. T300. The method above, using DELAY, works in general, regardless of the program cycle time.

2. The same techniques can be used for time-dependent blocks for other types of function besides 3-term control. When a timing diagram is drawn, it may be found useful to include other functions which can be operated at regular intervals. This ensures that no program scan is more heavily loaded, therefore taking longer than other program scans.

5.5. Determination of Controller Settings

When PIDCON is incorporated into the program, values must be entered to set the proportional gain, the integral time and the derivative time. These values can be determined by one of four methods.

5.5.1. Measurement Method

In this method, the plant is run in open loop and the response measured to a small step change of input. This response is used with standard formulae to find settings.

Record or plot the measured value to a small step change of control signal as shown below:

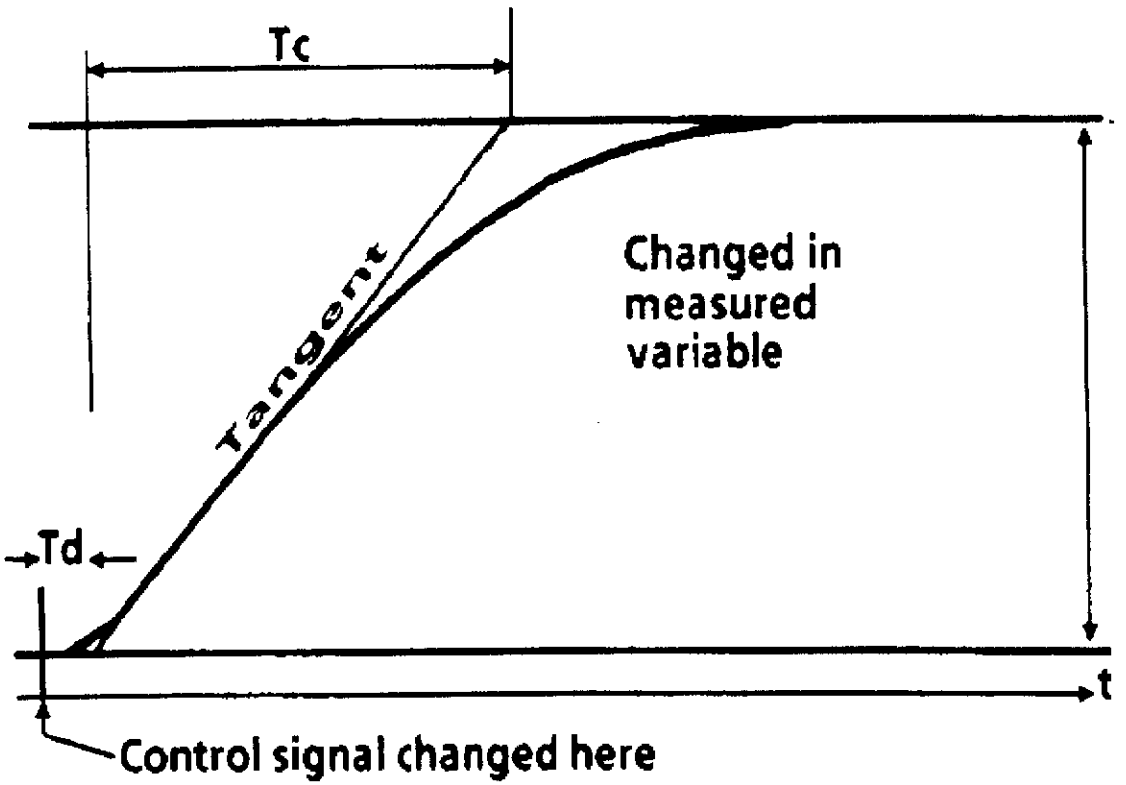


Figure 4 - Measured Value Plot

Note... When conducting such a test, use a control signal large enough to make the plant operate somewhere in the middle of its normal working range prior to applying a step change. For instance, with a valve actuator, the valve should already be open to some extent when the step change of control signal is applied, and not starting from its closed position. See, for instance, the worked example is section 11. This avoids non-linearities which are often found in plant characteristics at the extremes of the control range.

From this plot, measure three values as follows:

1. The gain of the plant:

$$A = \frac{\text{change in measured value}}{\text{change in control signal}}$$

2. The effective deadtime. Obtain this by drawing a tangent as shown, and find where it cuts the initial value of the measured value.

   Td = Effective deadtime

3. The effective time constant. Obtain this by extending the tangent to where it cuts the final value of the measured value and find the difference in time between the two ends of the tangent as shown

   Tc = Effective time constant

These three values are used in determining the 3-term controller settings and the repetition interval.

Note... If the plant is one in which the INCREMENTAL OUTPUT mode has to be used, it is not possible to produce a response like that shown, because the plant output continues to change until the control signal is set back to its home value. In this case, to conduct a test, change the control signal for a short time (e.g. 1 second) then return it to its 'home' value. Denoting the length of time the control signal is applied by Tt, the formula of the gain plant is:

$$A = \frac{\text{change in measured value}}{\text{change in control signal}} \times \frac{1}{Tt}$$

If the measured deadtime Td looks to be close to the test time Tt, the test should be performed again with a larger change of control signal for a shorter test time.

Closed Loop Response

If the plant is now closed loop controlled, using a 3-term control function, and with a very short update time which can be ignored, the fastest response which is normally obtainable for a small step change in setpoint is as shown below:

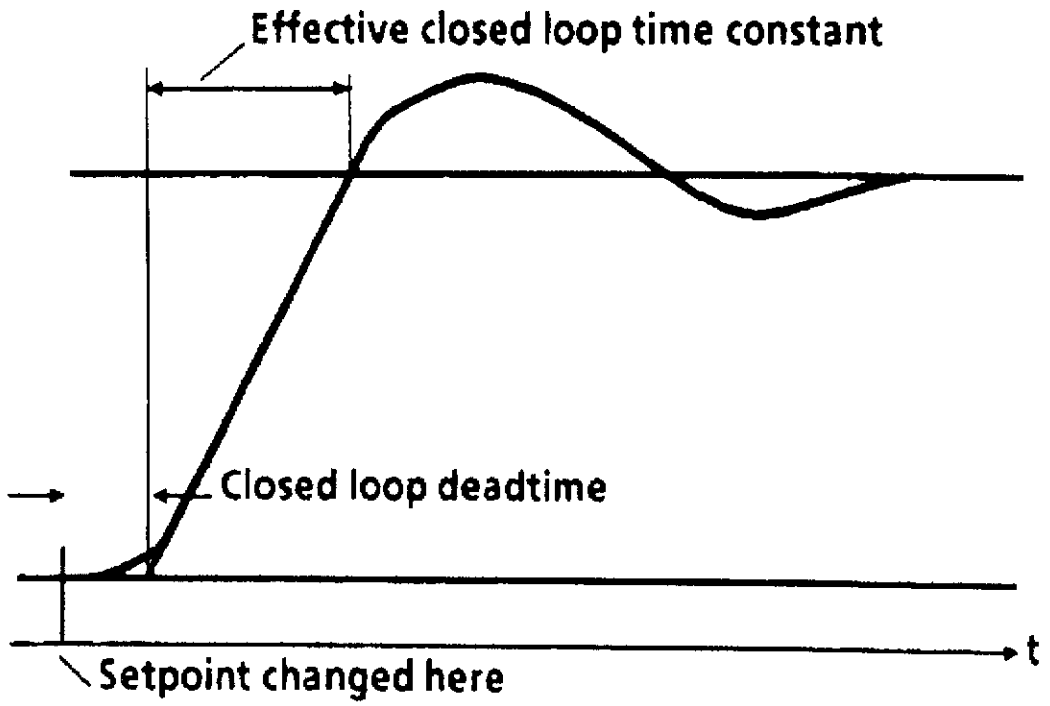


Figure 5 - Closed Loop Response Curve

The following approximations apply:

- The deadtime of the closed loop response is about the same as that of the open loop response.
- The effective time constant of the closed loop response is about the same as the deadtime.
- If the 3-term control function is tuned up any more, it is not possible to obtain any real improvement in the speed of response. Instead, the response becomes more oscillatory or unstable.

Proportional, Integral and Derivative Gains

Knowing the values of plant gain A, plant deadtime Td, and plant time constant Tc measured from the open loop response, settings can be determined using the following rules of thumb:

$$\text{Repetition interval} \qquad T = \frac{Td}{2}$$

In many cases, Td is so long that this condition is easily met, even with PIDCON placed in a block executed at intervals of several seconds. However, if Td is fairly short and if T is of the same order as Td/2, allow for the sampling delay by treating the effective plant deadtime as:

$$\text{Effective plant deadtime } Td' = Td + \frac{T}{2}$$

There is always a delay of one program execution between the time when the controller reads the setpoint and measured value inputs and the time when it sets the control signal output to the plant. For most process control applications, this time is negligible, but for very fast loops the execution time Te should still be added, making the effective plant deadtime into:

$$\text{Effective plant deadtime } Td' = Td + \frac{T}{2} + Te$$

Thus, for a plant with a 10-second time delay, operated by PIDCON within a BLOCK operated at 1-second intervals, and a program cycle time of 20ms, an effective deadtime of 10.52 seconds is produced. In this case, the extra 0.02 seconds makes negligible difference.

Rules of Thumb

For 2-term PI control

$$\text{Proportional gain} = \frac{1}{A} \times \frac{Tc}{Td'} \times 0.9$$

$$\text{Integral time } TI = \frac{Td'}{0.5}$$

$$\text{Derivative time } TD = Td' \times 0.5$$

Note... $$\text{Proportional band \%} = \frac{100}{\text{proportional gain}}$$

5.5.2. Calculation Method

If there is sufficient design data about the plant to be able to calculate its transfer functions, it may be possible to analyse the closed loop system by whatever methods are most familiar (Nyquist, Bode, etc.) to find the required controller settings.

Compared with a pure analogue control system, the only extra factor to be considered is the time delay effect of the program repetition interval. As shown previously, an additional time delay (transport lag) is generated by using a digital programmable controller of T/2 + Te which would not be present in an analogue system.

If the closed loop system is to have a bandwidth of 1 radian/second, and if Te is negligible, choosing T = 1/2 would give an extra 14° phase shift at the bandwidth frequency. Shortening the repetition interval would reduce the phase shift pro rata. Alternatively, if PIDCON is operated on every program execution so that T is equal to Te, it would be necessary to make Te = 1/6 to produce the same 14° phase shift.

From the analysis, it should be possible to determine the values needed for proportional gain, integral time (and derivative time, if derivative action is needed) and the program and BLOCK repetition intervals to give the type of response required.

5.5.3. Knowledge Method

Where a similar plant is GEM80 controlled, the values of the settings can be copied directly, at least for initial trials. Where a similar plant is analogue controlled, or where an analogue control loop is being replaced with a GEM80 system, the same settings can be used for the proportional gain, integral and derivative times, but a suitable BLOCK repetition interval must still be established.

Rules of Thumb

For 2-term PI control

$$T = 0.15TI$$

where TI is the integral time

For 3-term PID control

$$T = 0.25TI$$

where TI is the integral time

Note.. These formulae assume that the integral action has been increased to give more or less the quickest reset action to a setpoint change which gives a satisfactory response. If this is not so, it may be necessary to make the repetition interval somewhat shorter than implied by the above rules of thumb.

5.5.4. Cut-and-Try Method

This is the least methodical of the four methods in which an initial low gain is increased until the response becomes oscillatory. If the user is inexperienced in setting up closed loop controls, it is recommended that the measurement method detailed earlier is used.

The experienced user who has a 'feel' for the plant being controlled may, however, attempt to short cut open-loop testing of the plant and go direct to entering first-try table values. How these table values relate to the proportional gain, integral and derivative times is given is section 5.6. Further details are given in the worked examples in section 5.7.

5.5.5. Response not fast enough

Occasionally it may be found that the response obtainable from a closed loop system controlled by PIDCON is not as fast as that needed for the application.

Do not forget that the response obtainable depends on the system deadtime. In most cases, this is determined by the plant, see the Measurement Method described earlier. In a few cases it could be caused by the GEM80 program cycle time. Whatever the cause, the only way in which the response can be speeded up is by reducing the deadtime. The improvement in response time obtainable is proportional to the reduction in deadtime.

5.6. Table Values

The values which must be set for PIDCON are related simply to the values obtained for proportional gain, integral time and derivative time.

Table value P	=	proportional gain in units of 0.01 (e.g. value required is 5000 for a proportional gain of 50).
Table value I	=	P/TI (e.g. value required is 50 for an integral time of 100 seconds when P = 50000
Table value D	=	P/TD (e.g. 10000 for a derivative time of 2 seconds with P = 5000).

5.7. Worked Examples

5.7.1. Measurement Method

Temperature control. Output from a GEM80 feeds a valve in a steam heating pipe. Measured value comes from an RTD transducer.

Both the measured value and the control signal to the plant are scaled using LINCON Special Function S11 to be in the range 0 to 32000.

A test is conducted of running the valve with a control signal value of 12000, which is then increased as a step change to 13500. The corresponding change in the measured value is from 12375 to 12975. The response shows an effective deadtime of 12 seconds and an effective time constant of 140 seconds.

Therefore:

$$A = \frac{12975 - 12375}{13500 - 12000} = \frac{600}{1500} = 0.400$$

Td = 12 seconds

Tc = 140 seconds

Rule of thumb for repetition interval:

$$T = \frac{Td}{2} = 6s$$

Assume that T = 5s. The program execution time is likely to be much shorter than this and can be ignored. Therefore:

$$Td' = Td + \frac{T}{2}$$

$$= 12 + \frac{5}{2}$$

$$= 14.5s$$

Using 3-term control, the PID terms are:

$$\text{Proportional gain} = \frac{1}{A} \times \frac{Tc}{Td'} \times 1.2$$

$$= \frac{1}{0.400} \times \frac{140}{14.5} \times 1.2$$

$$= 28.97$$

$$\text{Integral time TI} = \frac{Td'}{0.5}$$

$$= \frac{14.5}{0.5}$$

$$= 29s$$

$$\text{Derivative time TD} = Td' \times 0.5$$

$$= 14.5 \times 0.5$$

$$= 7.25s$$

Lastly calculate the required table values:

P = proportional gain in units of 0.01

= 2897

$$I = \frac{P}{TI}$$

$$= \frac{2897}{29}$$

$$= 99.896$$

= 100 to the nearest integer

$$D = P \times TD$$

$$= 2897 \times 7.25$$

$$= 21003$$

5.7.2. Calculation Method

Examples are not given here, since analytical methods for the design of closed loop systems and worked examples can be found in many textbooks.

5.7.3. Cut-and-Try Method

Make a first estimate at the repetition interval, bearing in mind that this should be not more than half the plant deadtime. Assume a value of 1 second.

To start with, integral and derivative actions are not required so leave the table values for I and D set to zero. See also section 4.1.5 concerning the use of the Integral Equation without any I and D terms.

Now set in some low, arbitrary value for P so as to give a low proportional gain initially. If the control signal and measured value have been scaled using LINCON so that their span utilises most of the permissible number range, it is unlikely that P need be less than 100 as a first try.

Increase P by stages. This speeds up the plant response to setpoint changes. Eventually, an optimum setting is reached beyond which the plant response becomes more oscillatory than is acceptable. Assume that the optimum setting for P is 3520.

While going through the above procedure for setting up P, the measured value does not exactly match the setpoint, but is always somewhat less. The next requirement is some integral action to reset the measured value exactly equal to the setpoint. However, because integral terms tend to have a destabilising influence, the P term must first be reduced to 90% of the optimum value previously found. That is, set P to 3520 x 0.9 = 3168.

Start to increase the value of I and for each new trial setting of I, do not increase it to more than twice its previous setting. Suppose the optimum value is found to be 98.

In the case of 2-term PI control, that is all we need to do. However, it is worth checking whether the repetition interval, which was initially set to 1 second, could be made longer.

Using the formula for I in section 5.6 backwards, with the P and I settings, the integral time set for PIDCON is:

$$TI = \frac{\text{Table value for P}}{\text{Table value for I}}$$

$$= \frac{3168}{98}$$

$$= 32.2 \text{ seconds}$$

For 2-term control with a well tuned controller, it should be possible to use a repetition interval approaching 0.15TI (see 5.5.3)

or 0.15 x 32.3 = 4.85 seconds

Allowing for the fact that the 2-term control may not have been tuned precisely to optimum, increase the repetition interval to 4 seconds. All that is necessary is to alter the value that defines the DELAY time on the BLOCK enclosing PIDCON, or whatever method has been used for setting the PIDCON repetition interval. Altering this repetition interval should not make any significant difference to the response of PIDCON and the loop it is controlling.

If it is considered that a derivative term would be worthwhile and not introduce too much noise, the 2-term control can be converted into a 3-term control. Unlike reset action, derivative action tends to make a control more stable. Once D has been increased to its optimum setting, it should then be possible to increase P by about 30% and I by about 100%. So, with the D term added, it should be possible to set:

P = 3168 x 1.3 = 4118
I = 98 x 2 = 196

Obviously, these higher P and I values must not be set until after the D-term has been added, otherwise an oscillatory or unstable system could result.

6. EXERCISES

With a GEM80 Controller and an 8902 Simulation Unit, try out PIDCON using the following simple exercises, in which the plant is simulated within the GEM80 program itself. The user may, if so desired, alter the characteristics of the simulated plant.

In both exercises, the setpoint is taken from the thumbwheel switches (A3), and the response of the measured value is displayed on the LCD numeric display (B2).

6.1. Absolute Output Exercise

Set the controller program repetition interval to 100ms by making P1 = 100.

The plant may be simulated by the following rungs:

```
|                                                                          |
+---<AND>-----<AND>-------------------------------------------<OUT>---+
|    P1000      W1000                                          G1000    |
|                                                                       |
|  OUTPUT              OUTPUT                                           |
+---SPEC-----VALUE-----SPEC-----VALUE-----------------------<OUT>---+
|     S8         W3        S8        W13                      W23     |
|                                                                       |
+---<AND>-----------------------------------------------------<OUT>---+
|     500                                                      W1      |
|                                                                       |
+---<AND>-----------------------------------------------------<OUT>---+
|     600                                                      W11     |
|                                                                       |
+---<AND>-----------------------------------------------------<OUT>---+
|     700                                                      W21     |
|                                                                       |
|  OP1/RU    TCONST           TCONST           TCONST           MV1/RU|
+---<AND>----SPEC----VALUE----SPEC----VALUE----SPEC----VALUE--<OUT>-+
|     G10       S60      W0        S60      W10       S60      W20     G12  |
```

The necessary set up of the Special Function has been included within the ladder rungs; refer to the Software Data Sheet for further details.

Explanation:

The rungs represent the plant as a series of three time constants in cascade. When several time constants are put in cascade, always put the larger ones last and the smaller ones first in the rungs.

In order to display the Measured Value on the LCD the following rung should be inserted:

```
|  MV1/RU     BCDOUT                                              DISPLAY  |
+---<AND>-----SPEC-----------------------------------------------<OUT>---+
|     G12         S2                                                B2     |
```

This rung simply takes the output of the previous rung; the Measured Value and converts it using BCDOUT into a suitable form to drive the display.

Before trying to put a closed loop control round the 'plant', try to get the feel of the plant response by running it open loop, using the thumbwheel switches to drive directly into G10, using the following rung:

```
|   INPUT      BCDIN                                                OP1/RU    |
+---<AND>-----SPEC-------------------------------------------------<OUT>---+
|    A3        S1                                                   G10     |
```

For a large change of input signal, for example if the thumbwheel is altered from 0000 to 1000, the number displayed changes slowly for the first second or two, then speeds up to nearly a constant rate, after which the rate slows, the last figure taking a considerably long time to change.

To use the measurement method as given in section 5.5.1 it is necessary to make a plot of the plant response to find the deadtime and the effective time constant. One way to do this is to route the measured value output from G12 to an analogue output (e.g. D32) to which a pen recorder could be connected. Another way is to write a section of program in the GEM80 that copies the contents of G12 every 5 seconds, say, into a sequence of table locations that can subsequently read off as a data list on the Programmer screen, and then be plotted as a graph.

To use the cut-and-try method of section 5.5.4, put in the following rungs:

```
|                                                                          |
+---]/[-----DELAY-----VALUE-------------------------------------( )---+
|  G0.0       G1          5                                      G0.0    |
|                                                                        |
+---] [-----------------------------------------------------------OBEY---+
|  G0.0                                                         BLOCK    |
|                                                                 *      |
| ***************************** START OF BLOCK ******************       |
|                                                                        |
|   INPUT     BCDIN     PIDCON                                   P1/RU   |
+---<AND>-----SPEC------SPEC-----VALUE--------------------------<OUT>---+
|    A3         S1        T60       W100                        G10     |
|                      +--<   >                                          |
|                      |                                                 |
|  MV1/RU              |                                                 |
+---<AND>--------------+                                                 |
|    G12                                                                 |
|                                                                 EVAL   |
+-----------------------------------------------------------------( )---+
|                                                                W101.11 |
|                                                                        |
|                                                                ATMDEP  |
+-----------------------------------------------------------------( )---+
|                                                                W101.12 |
|                                                                        |
+ ***************************** END OF BLOCK ********************** +
|                                                                        |
```

Before proceeding any further delete the rung above which drives the plant control signal G10 directly from the input thumbwheel A3.

It is now necessary to set up the data table values for PIDCON as given in Table 2 at the beginning of this data sheet:

MODEWORD	W101	=	@4D (see below)	
RATIO	W105	=	1000	
BIAS	W106	=	0	
P	W107	=	160	) You may want
I	W108	=	15	) to try
D	W109	=	4	) different values
DBAND	W110	=	10	
DLIM	W113	=	32000	
MANUAL	W114	=	100	
LOLIM	W115	=	0	
HILIM	W116	=	32000	
L1	W117	=	600	
L2	W118	=	600	

MODEWORD

W101.0	ON	AUTO
W101.1	OFF	Not DDC Mode
W101.2	ON	REMOTE Setpoint
W101.3	ON	DIRECT
W101.4	OFF	MV Derivative
W101.5	OFF	BASIC Equation
W101.6	ON	ABSOLUTE Output
W101.7	OFF	Not Raise
W101.8	OFF	Not Lower
W101.9	OFF	Slow
W101.10	OFF	Not DERIVATIVE SCALE
W101.11		Set by ladder rungs
W101.12		Set by ladder rungs
W101.13	OFF	INTERNAL Setpoint
W101.14		Not used
W101.15		No fault

The other table locations contain intermediate values used or calculated by PIDCON itself and should not be written to.

Try the effect of changing the setpoint on A3 and observe the plant response on B2. Try altering the values of P, I and D in W107 to W109. It should not be difficult to make the system unstable if the values are made too high.

For the calculation method of section 5.5.2, use whichever method is most convenient.

6.2. Incremental Output Exercise

Simulate the plant by the same rung as before, but insert as the final element before the OUT to G12, an ADD with location G12:

```
| OP1/RU  TCONST        TCONST        TCONST        MV1/RU  MV1/RU   |
+--<AND>---SPEC---VALUE--SPEC---VALUE--SPEC---VALUE--<ADD>---<OUT>---+
|   G10     S60     W0    S60     W10   S60     W20    G12     G12    |
```

Set PIDCON to the INCREMENTAL OUTPUT mode by making W101.6 = 0. Then try running in a similar manner to the ABSOLUTE OUTPUT mode exercise. The W-table, W115 to W118 that set the limits is ignored in the INCREMENTAL OUTPUT mode.

Explanation:

The additional ADD in the plant simulation rung means that, under steady state conditions, the control signal in G10 must be zero. For this condition, the rung does not add anything to G12 so that G12 remains at whatever value it has reached without any control signal input. This is the sort of plant characteristic that INCREMENTAL OUTPUT mode is intended to control, where, after a change of setpoint, the control signal always comes back to its 'home' value; in this case, zero.

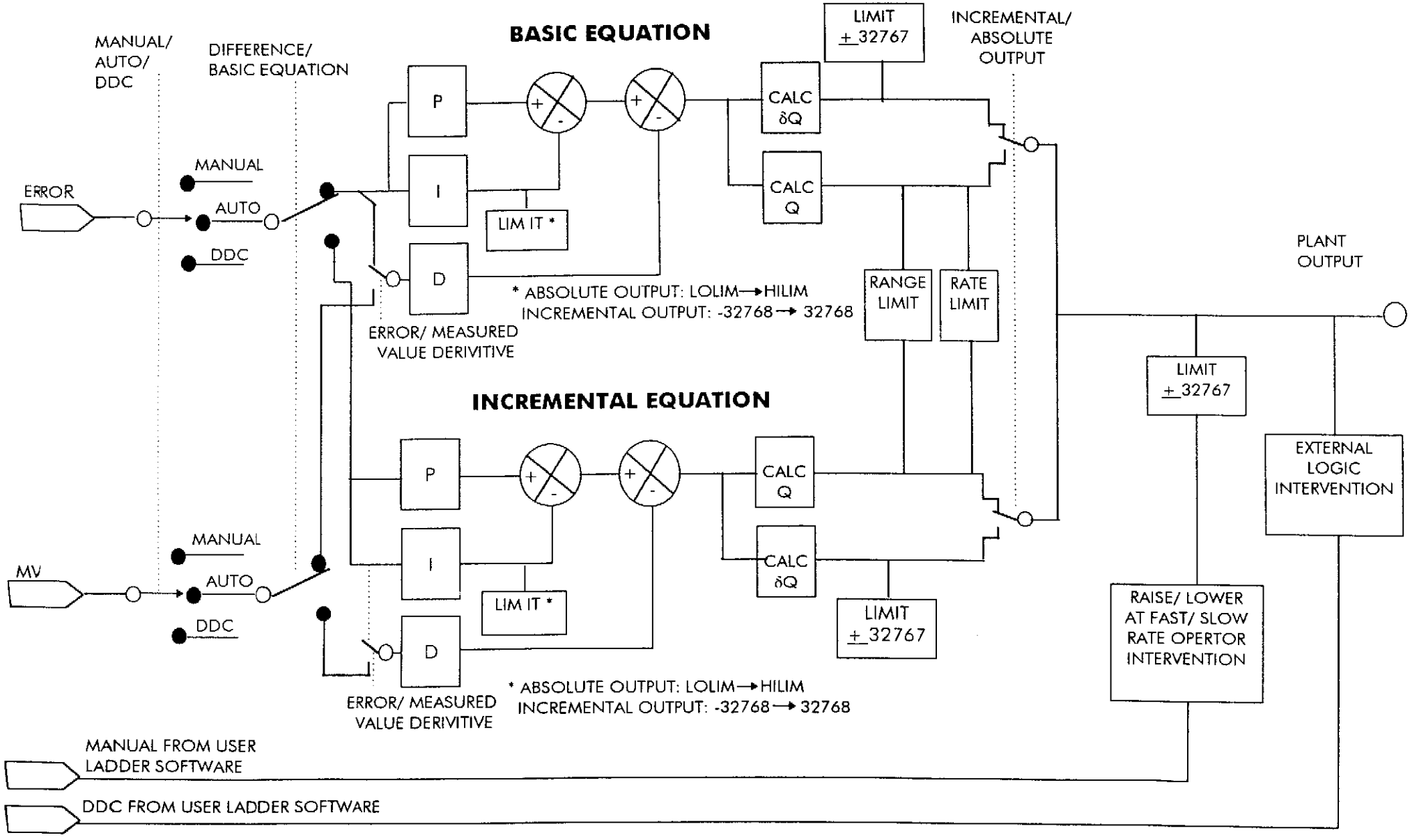


FIGURE 6 PIDCON MAIN FUNCTIONS

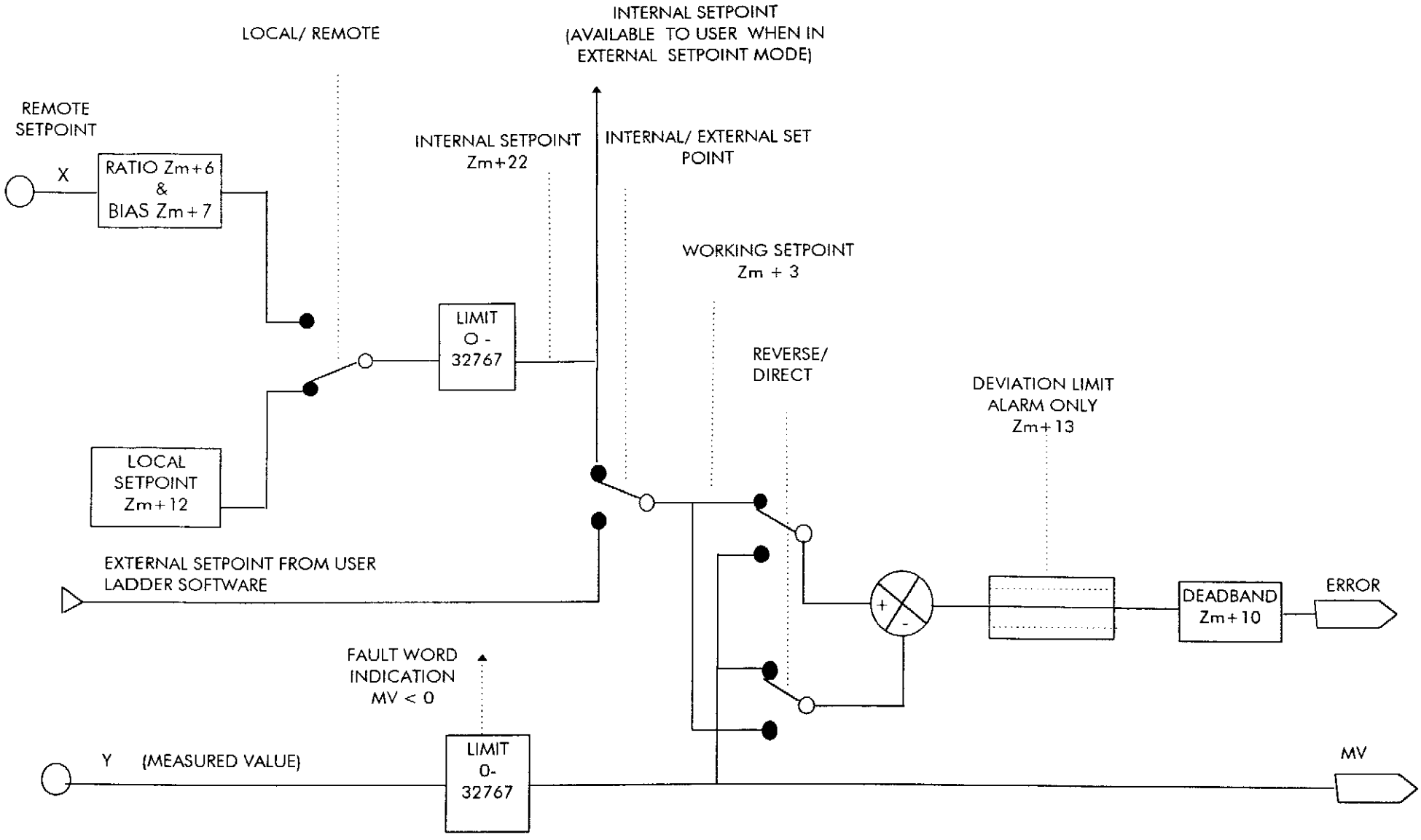


FIGURE 7 PIDCON INPUT STAGE

Publication No.T1300 - Issue 4

ALSTOM

DRIVES & CONTROLS

West Avenue, Kidsgrove, Stoke-on-Trent
Staffordshire, ST7 1TW, UK
Tel: +44 (0) 1782 781000
Fax: +44 (0) 1782 781001

Culemeyerstraße 1
D-12277 Berlin
Tel: +49 (0) 30 7496 2656
Fax: +49 (0) 30 7496 2494

5, avenue Newton
BP215, 92142 Clamart Cedex France
Tel: +33 (0)1 46 29 15 30
Fax: +33 (0)1 46 29 15 29