TP4-PH Output Addendum

AMALGAMATED INSTRUMENT CO *ABN: 80 619 963 692*
Unit 5, 28 Leighton Place Hornsby *Telephone: +61 2 9476 2244* *e-mail: sales@aicpl.com.au*
NSW 2077 Australia *Facsimile: +61 2 9476 2902* *Internet: www.aicpl.com.au*

Table of Contents

1 Introduction

This addendum to the main TP4-PH manual contains information for the operation of the PI control mode for relay and analog output operation and for serial communications. Refer to the main TP4-PH manual for all other information regarding this instrument

2 Setting up the relay PI controller

The Relay Proportional + Integral Controller can be made to operate in either pulse width control or frequency control mode via the **Rx OPEr** function. Note that the **Rx OPEr** function will not be seen until a value has been set for the low or high alarm e.g. for **R 1Lo** or **R 1H**. The best results are usually achieved by initially configuring as a "Proportional Only" controller and then introducing the Integral functions when stable results are obtained.

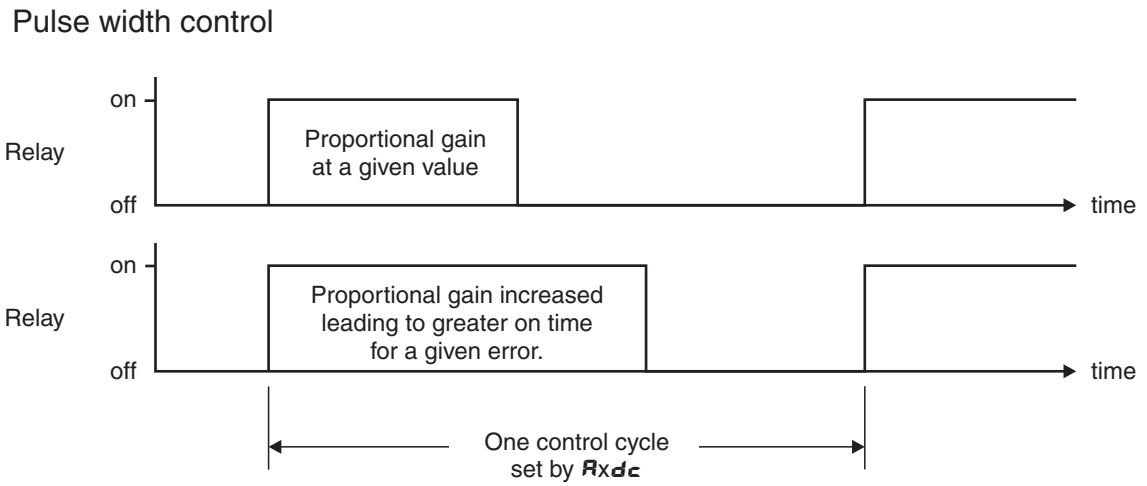
Relay 1 and relay 2 can be set to operate in PI control mode. Any other relays fitted will only operate in normal, non PI operation. The "*x*" in the **Rx OPEr** and other functions indicates the chosen relay i.e. for relay 1 the display will show **R 1 OPEr**, **R 1.SP** etc. The **Rx OPEr** function allows three choices of operating mode for the chosen relay, namely **Rx.AL**, **Rx.tP** and **Rx.Fr**. If **Rx.AL** is selected the chosen relay will operate as a setpoint relay whose operation is controlled by the **RxH**, , **RxLo** etc. settings and the PI control settings will not be seen. See the "Explanation of functions" chapter for details of operation when **Rx.AL** is selected. If **Rx.tP** is selected then the chosen relay will operate in pulse width control mode. If **Rx.Fr** is selected then the chosen relay will operate in the frequency control mode.

Pulse width control - operates by controlling the on to off time ratio of the relay. In a typical application this would be used to control the length of time for which a dosing pump is switched on during a control cycle i.e. the pump or other device will continuously operate for the length of time the relay is activated and will stop operating when the relay is de-activated.

Frequency control - operates by changing the rate at which the relay switches on and off. In a typical control application the frequency control operation is particularly suited for use when one shot dosing is used i.e. the pump or other device puts out a fixed dosing quantity for every pulse received.

2.1 Relay pulse width modulation control mode

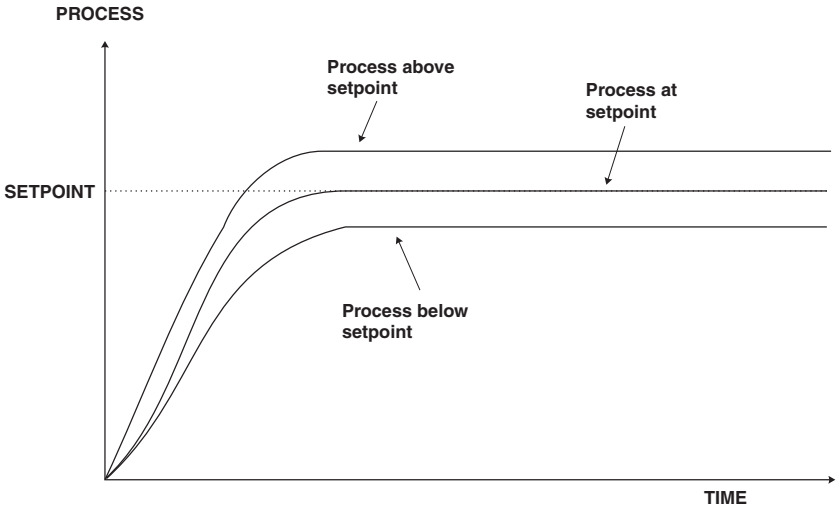
To use pulse width modulation control **Rx.tP** must be selected at the **Rx OPEr** function.



2.2 PI relay control setpoint

Display: Rx.SP
Range: Any display value
Default Value: 0

The control setpoint is set to the value in displayed engineering units required for the control process. The controller will attempt to vary the control output to keep the process variable at the setpoint. Note that the control setpoint value can be reached and adjusted via the "easy access" mode (see "Explanation of functions" chapter) if the ACCS function is set to EASY. This feature could be useful if the setpoint is to be frequently changed.



2.3 PI relay control span

Display: ctrl SPAN
Range: Any display value
Default Value: 100

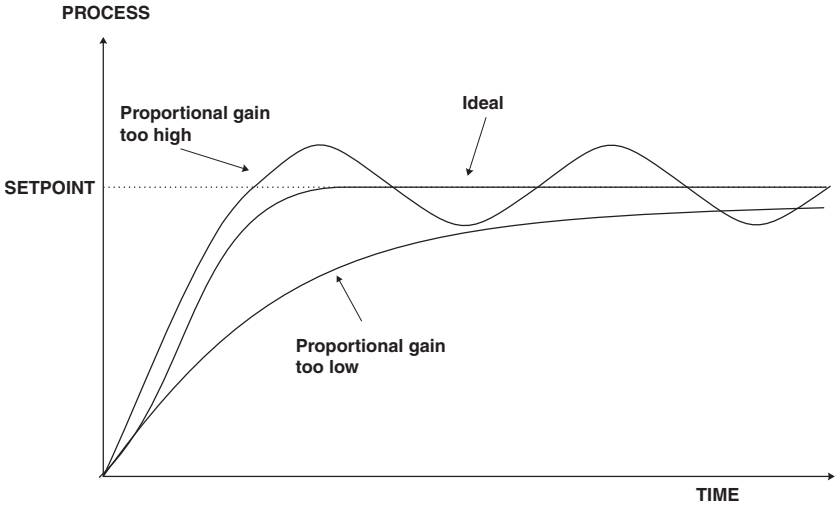
The function of the control span is to define the limit to which the PI control values will relate. The control span value will be common to all control relays i.e. if more than one control relay output is being used then each of these relays operates from the same control span setting. The span value defines the range over which the input must change to cause a 100% change in the control output when the proportional gain is set to 1.000. This function affects the overall gain of the controller and is normally set to the process value limits that the controller requires for normal operation. For example if the control setpoint (Rx.SP) is 7.0 and the ctrl SPAN is 2.0 and Rx.P9 is set to 1.000 then an error of 2.0 from the setpoint will cause a 100% change in proportional control output. For example with Rx.SP at 7.0, ctrl SPAN at 2.0, Rx.P9 at 1.000 and Rx65 at 0.000 a display reading of 5.0 or lower (Rx.SP minus ctrl SPAN) the control output will be at 100% i.e. the relay will be on continuously. The control output will then gradually adjust the on/off time as the display value reaches the setpoint.
For instruments with more than one input where the number of decimal points displayed may vary the control span will take on the value of the main display and so may or may not match the decimal points shown in the input being controlled. e.g. a control span of 2.00 will act as a control span of 20.0 if the input to be controlled is displayed with only 1 decimal point.

2.4 PI relay proportional gain

Display: *Rx.P9*
Range: *-32.767* to *32.767*
Default Value: *0.0 10*

Note: the range value may be restricted if the number of display digits does not allow viewing of the full range.

The proportional value will determine the degree to which the controller will respond when there is a difference (error) between the measured value and the process setpoint. If the proportional gain is increased then for a given error the relay on time will be increased (or decreased if the error is on the other side of the setpoint). The proportional gain action can be reversed by setting a negative gain i.e. with a negative gain the on time will reduce as the error increases. With a proportional gain of *1.000* and an error of *1.0* or more (with control span set at *1.0*) the controller will increase the frequency by 100% if possible. With a proportional gain of *0.500* an error of *1.0* or more (with control span set at *1.0*) will cause the controller to increase the frequency by 50%, if possible. Too much proportional gain will result in instability due to excessive overshoot of the setpoint. Too little proportional gain will lead to a slow response.



This table shows the effect of the output frequency of changing proportional gain and bias with the following settings: ctrl SPAN = 2.0, A 1.dc = 1.0, A 1.1 9 = 0.000			
A 1.SP	**A 1.P9**	**A 1.bS**	**Effect on relay operation**
7.0	1.000	0.0	Reading of 5.0 or below - relay permanently on. Reading of 5.0 to 7.0 - relay pulses with off time increasing as value approaches 7.0. Reading 7.0 or above - relay permanently off.
7.0	1.000	100.0	Reading of 7.0 or below - relay permanently on. Reading of 7.0 to 9.0 - relay pulses with off time increasing as value approaches 9.0. Reading 9.0 or above - relay permanently off.
7.0	1.000	50.0	Reading of 6.0 or below - relay permanently on. Reading of 6.0 to 7.0 - relay pulses with off time increasing as value approaches 7.0. Reading 7.0 - relay pulses at 50% on and 50% off. Reading 7.0 to 8.0 - relay pulses with off time increasing as value approaches 8.0. Reading 8.0 or above - relay permanently off.
7.0	0.500	50.0	Reading 5.0 or below - relay permanently on. Reading 5.0 to 7.0 - relay pulses with off time increasing as value approaches 7.0. Reading 7.0 - relay pulses at 50% on and 50% off. Reading 7.0 to 9.0 - relay pulses with off time increasing as value approaches 9.0. Reading 9.0 or above - relay permanently off.
7.0	- 1.000	50.0	Reading of 6.0 or below - relay permanently off. Reading of 6.0 to 7.0 - relay pulses with on time increasing as value approaches 7.0. Reading 7.0 - relay pulses 50% on and 50% off. Reading 7.0 to 8.0 - relay pulses with on time increasing as value approaches 8.0. Reading 8.0 or above - relay permanently on.

2.5 PI relay integral gain

Display: A*x*.1 9

Range: -32.767 to 32.767

Default Value: 0.000

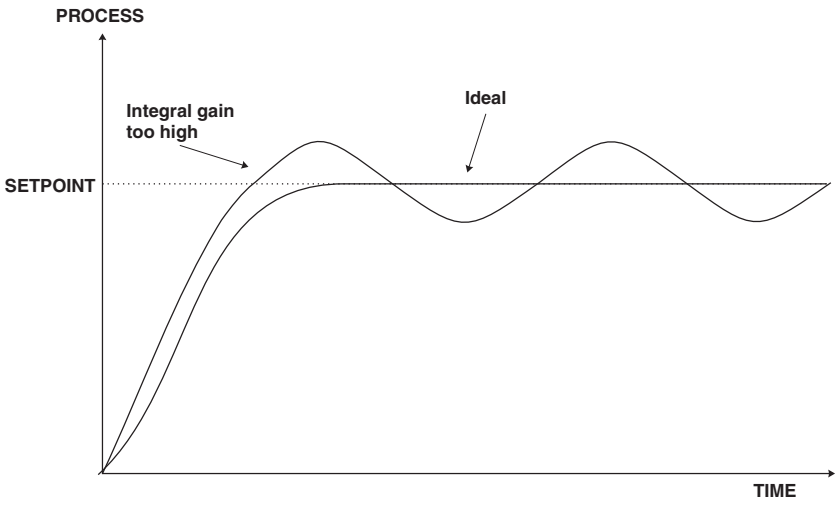
Note: the range value may be restricted if the number of display digits does not allow viewing of the full range.

The Integral action will attempt to correct for any offset which the proportional control action is

unable to correct (e.g. errors caused by changes in the process load). When the integral gain is correctly adjusted the control output is varied to maintain control by keeping the process variable at the same value as the control setpoint. Since the integral gain is time based the output will gradually increase if the error does not decrease i.e. if the measured value remains constant and there is an error (a difference between the measured value and the setpoint) then the frequency will be increased compared to the previous frequency output. The higher the proportional gain, the greater the degree by which the on to off ratio will be affected i.e. the response will be greater at higher integral gain settings. With an integral gain of **1.000** an error of **1.0** or more (with control span set at **1.0**) will cause the integral action to try to correct at the rate of 100% minute. With an integral gain of **0.200** an error of **1.0** or more will cause the integral action to try to correct at the rate of 20% per minute. Too high an integral gain will result in instability. To low an integral gain will slow down the time taken to reach the setpoint. The optimum setting will depend on the lag time of the process and the other control settings. Start with a low figure (e.g. **0.200**) and increase until a satisfactory response time is reached. The integral gain figure has units of gain/minute. The integral action can be reversed by setting a negative gain figure, note that the sign of the integral gain must match the sign of the proportional gain. The integral control output follows the formula:

$$Integral\,control\,output = \frac{error \times Ig \times time\,(seconds)}{60} + previous\,integral\,control\,output$$

Where Ig is the integral gain set via **Rx.I 9**.



2.6 PI relay integral control high limit

Display: **Rx.I H**
Range: **0.0** to **100.0**
Default Value: **100.0**

The maximum limit can be used to reduce overshoot of the control setpoint when the control output is increasing i.e. rising above the setpoint. Other than this the limit operates in the same manner as the low limit described previously.

2.7 PI relay integral control low limit

Display: Rx.I L
Range: 0.0 to 100.0
Default Value: 100.0

The minimum limit can be used to reduce overshoot of the control setpoint when the control output is being reduced i.e. falling below the setpoint. The low limit reduces the available output swing by a percentage of the maximum output. Without a limit the integral output can be very large at the time the setpoint is reached and a large overshoot of the will then result. Settings available are from 0.0 to 100.0 (%). If the limit setting is too high then overshoot will result. If the setting is too low then the integral output can be limited to such an extent that the setpoint cannot be maintained.

Start with a low value such as 20.0 and increase or decrease the value until a satisfactory result is obtained. The advantage of using separate low and high limits is that in many applications the response is very one directional e.g. the system may respond very quickly to a heat input but may cool down at a much slower rate. Separate high and low limit settings allow independent limiting of the integral control swing below and above the setpoint so a smaller minimum limit can be set to limit swings below the setpoint to compensate for the slower cooling time.

The minimum and maximum limits are used in conjunction with the output bias setting to maintain the control process setpoint value. For example with a bias (Rx.bS) set at 50%, minimum limit set at 20% and a maximum limit of 30% the actual bias when the process is at the setpoint may be anywhere between 30% and 80% i.e. Integral control is being used to alter the bias setting in order to maintain the process at the setpoint. In this case the minimum term will allow the bias to drop to a value between 50% and 30% in order to maintain the setpoint. The maximum term will allow the bias point to rise to a value between 50% and 80% in order to maintain the setpoint.

2.8 PI relay control output bias

Display: Rx.bS
Range: 0.0 to 100.0
Default Value: 50.0

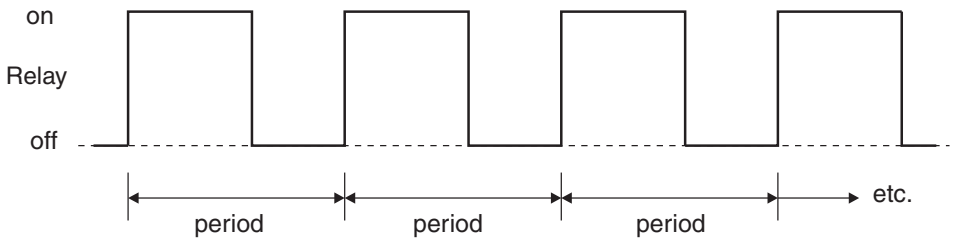
The control bias sets the ideal steady state output required once the setpoint is reached. Settings are in % from 0.0 to 100.0. When set at 0.0 the relay will be de-activated for the entire control period when the measured input is at the setpoint (depending on proportional and integral gain settings). If set at 50.0 then the relay operation frequency will on for 50% and off for 50% of the duty cycle time when the measured input is at the setpoint. If set at 100.0 then the relay will activated for the whole time whist the measured input is at the setpoint.

2.9 PI relay control cycle period

Display: Rx.dc
Range: 0 to 250
Default Value: 10

Displays and sets the control period cycle from 0 to 250 seconds. The control period sets the total time for each on/off cycle. This time should be set as long as possible to reduce wear of the

control relay and the controlling device.



2.10 Setting up the PI pulse width controller

1. Set the Rx OPEr function to Rx.tP.
2. Set the control setpoint Rx.SP to the required setting.
3. Set the control span ctrl SPAN to the required setting.
4. Set the proportional gain Rx.P9 to an arbitrary value e.g. 0.500.
5. Set the integral gain Rx.I 9 to 0.000 (i.e. off).
6. Set the low and high integral Rx.I L and Rx.I H limits to an arbitrary value e.g. 20.00.
7. Set the bias Rx.bS to 50.0.
8. Set the cycle Rx.dc period to 20 seconds.

Initialise the control system and monitor the control results. If the original settings causes process oscillations then gradually decrease the proportional gain until the oscillations decrease to an acceptable steady cycle. If the original settings do not cause process oscillations then gradually increase the proportional gain until a steady process cycling is observed.

Once the steady cycling state is achieved note the difference between the display value and the control setpoint value. Gradually increase or decrease the bias value until the displayed value matches (or cycles about) the control setpoint value.

Gradually increase the integral gain until the process begins to oscillate. Then reduce the integral gain slightly to regain the control without this added oscillation.

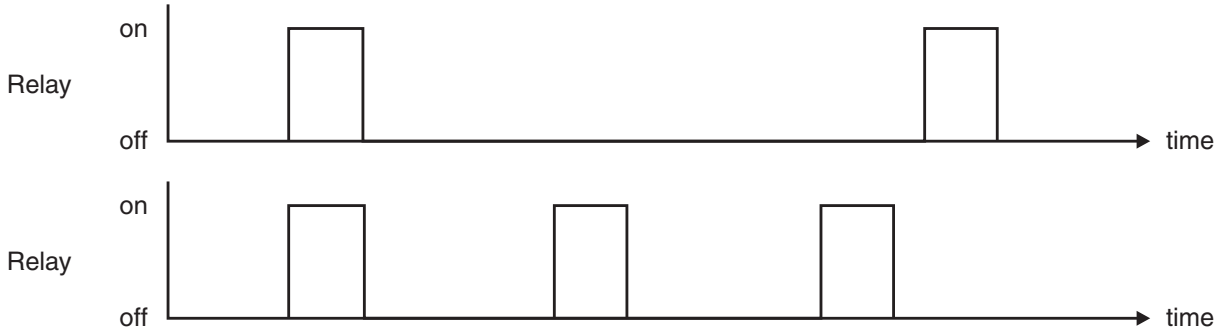
Create a step change to the process conditions and observe the control results. It may be necessary to fine tune the settings and use integral limits to obtain optimum results.

Set up sequence	**Symptom**	**Solution**
Proportional gain	Slow response	Increase proportional gain
Proportional gain	High overshoot or oscillation	Decrease proportional gain
Proportional bias	Process above or below setpoint	Increase or decrease bias as required
Integral gain	Slow response	Increase integral gain
Integral gain	Instability or oscillations	Decrease integral gain

2.11 Relay frequency modulation control mode

To use pulse width modulation control **Ax.Fr** must be selected at the **Ax OPEr** function. In frequency modulation mode the relay on time is fixed. A minimum relay off time can also be set. The control program will vary the actual off time to suit the error seen between the setpoint and the measured temperature at the time. For example if extra dosing is needed to reach the setpoint then the off time will be reduced resulting in more on pulses per period of time i.e. the frequency of the pulses is controlled to allow the setpoint to be maintained.

Frequency control - pulse frequency varies according to settings and control requirement



Frequency PI control operation has many functions in common with PI pulse width control, refer to the appropriate sections as shown below for these common functions.
Ax.SP (Control setpoint) - refer to section 2.2
ctrl SPAN (Control span) - refer to section 2.3
Ax.Pg (Proportional gain) - refer to section 2.4
Ax.I g (Integral gain) - refer to section 2.5
Ax.I L (Integral control low limit) - refer to section 2.7
Ax.I H (Integral control high limit) - refer to section 2.6
Ax.bS (PI control bias) - refer to section 2.8
Ax.dc (PI control cycle period) - refer to section 2.9. In frequency mode this function sets the minimum off time. If set to **0** the relay will be disabled. The control program can extend the off time to maintain the setpoint but not reduce it. If a 100% error is seen then the pulse rate will be at its maximum i.e. the off time will equal **Ax.dc**. If a 50% error is seen there will be a pulse every 2 times **Ax.dc**. For a 25% error there will be a pulse every 4 times **Ax.dc** and for a 10% error there will be a pulse every 10 times **Ax.dc**.

This table shows the effect of the output frequency of changing proportional gain and bias with the following settings: ctrl SPAN = 2.0, A 1.dc = 1.0, A 1.1 9 = 0.000			
A 1.SP	**A 1.P9**	**A 1.65**	**Effect on relay operation**
7.0	1.000	0.0	Reading of 5.0 or below - relay pulses at maximum frequency. Reading of 5.0 to 7.0 - relay pulses with frequency decreasing as value approaches 7.0. Reading 7.0 or above - relay permanently off.
7.0	1.000	100.0	Reading of 7.0 or below - relay pulses at maximum frequency. Reading of 7.0 to 9.0 - relay pulses with frequency decreasing as value approaches 9.0. Reading 9.0 or above - relay permanently off.
7.0	1.000	50.0	Reading of 6.0 or below - relay pulses at maximum frequency. Reading of 6.0 to 8.0 - relay pulses with frequency decreasing as value approaches 8.0. (period increased by 50% at 7.0 compared to minimum period e.g. if minimum period is 4 seconds the period at 7.0 will be 6 seconds) Reading 8.0 or above - relay permanently off.
7.0	0.500	50.0	Reading 5.0 or below - relay pulses at maximum frequency. Reading 5.0 to 9.0 - relay pulses with frequency decreasing as value approaches 9.0. (period increased by 50% at 7.0 compared to minimum period e.g. if minimum period is 4 seconds the period at 7.0 will be 6 seconds) Reading 9.0 or above - relay permanently off.
7.0	- 1.000	50.0	Reading of 6.0 or below - relay permanently off. Reading of 6.0 to 8.0 - relay pulses with frequency decreasing as value approaches 8.0. (period increased by 50% at 7.0 compared to minimum period e.g. if minimum period is 4 seconds the period at 7.0 will be 6 seconds) Reading 8.0 or above - relay pulses at maximum frequency.

2.12 PI relay on duration

Display: A*x*.dr
Range: 0.0 to 25.0
Default Value: 1.0

Displays and sets the control relay on duration from 0.0 to 25.0 seconds. If set to 0.0 the relay will be disabled. The duration should be long enough to ensure that the device being controlled receives an acceptable on pulse.

2.13 Setting up the PI frequency controller

1. Set the A*x* OPEr function to A*x*Fr.
2. Set the control setpoint A*x*.SP to the required setting.

3. Set the control span **ctrl SPAN** to the required setting.
4. Set the proportional gain **Ax.Pg** to an arbitrary value e.g. **0.500**.
5. Set the integral gain **Ax.I g** to **0.000** (i.e. off).
6. Set the low and high integral **Ax.I L** and **Ax.I H** limits to an arbitrary value e.g. **20.00**.
7. Set the bias **Ax.bS** to **50.0**.
8. Set the cycle **Ax.dc** period to **20** seconds.
9. Set the relay on time **Ax.dr** to an arbitrary value e.g. **1.0**

Initialise the control system and monitor the control results. If the original settings causes process oscillations then gradually decrease the proportional gain until the oscillations decrease to an acceptable steady cycle. If the original settings do not cause process oscillations then gradually increase the proportional gain until a steady process cycling is observed.

Once the steady cycling state is achieved note the difference between the display value and the control setpoint value. Gradually increase or decrease the bias value until the displayed value matches (or cycles about) the control setpoint value.

Gradually increase the integral gain until the process begins to oscillate. Then reduce the integral gain slightly to regain the control without this added oscillation.

Create a step change to the process conditions and observe the control results. It may be necessary to fine tune the settings and use integral limits to obtain optimum results.

Set up sequence	**Symptom**	**Solution**
Proportional gain	Slow response	Increase proportional gain
Proportional gain	High overshoot or oscillation	Decrease proportional gain
Proportional bias	Process above or below setpoint	Increase or decrease bias as required
Integral gain	Slow response	Increase integral gain
Integral gain	Instability or oscillations	Decrease integral gain

3 Serial communication and datalogger options

3.1 Serial Operation and Commands

A choice of one of three serial communication modes can be made in the TP4. The modes of operation available are: **d. SP**, **Cont**, **POLL** or **A.buS**. **A.buS** is a special binary output for communications with optional download and data logger Windows compatible software for use with a PC, consult the booklet supplied with this Windows software for details. The other modes are described below.

3.2 d. SP - Image display mode

This is a one way communications mode in which the display value on the instrument is sent via RS232 or RS485 as raw data in the following format:

$$< ESC > IXYYYYY$$

Where	$< ESC >$	is the ASCII Escape character (27 Dec., 1B Hex.)
	I	is the ASCII character "I" (73 Dec., 49 Hex.)
	X	is the number of image bytes in ASCII (31 to 38 Hex)
	$YYYYY$	is the raw 8 bit display data

This information is output every display update (typically approx. 4 times per second). The number of image bytes sent depends on the number of display digits present. This mode is suitable only when the receiving unit is produced by the same manufacturer as the TP4. The most common usage would be to provide slave displays for the measuring instrument. The slave displays would automatically detect the image mode data and display the correct value accordingly. The data is in seven segment display image i.e. Bit 0 is segment A, Bit is segment B etc.

3.3 Cont - Continuous output mode

In this mode the display value is continually sent via the RS232/485 interface in ASCII format with 8 data bits + 1 stop bit. Data will be updated at slightly less than the sample rate for the instrument being used. The standard format for **Cont** mode is as follows:-

$$< STX > XYYYYY, XZZZZZ, XTTTTT < CR >$$

Where	$< STX >$	is the ASCII Start of text character (2 Dec., 02 Hex.)
	X	is an ASCII Space character (32 Dec., 20 Hex.) for a positive value or
	X	is the ASCII character "-" (45 Dec., 2D Hex) for a negative value
	$YYYYY$	is the display value in ASCII for channel 1
	$ZZZZZ$	is the display value in ASCII for channel 2
	$TTTTT$	is the display value in ASCII for temperature
	$< CR >$	is the ASCII Carriage return character (13 Dec., 0D Hex.)

e.g.: If the display is showing 123456 then the instrument will send 02 31 32 33 34 35 36 0D (HEX) to the host.

3.4 POLL mode commands

This mode requires a host computer. PLC or other device to poll the instrument to obtain display or other information or reset various setpoint parameters. Special communications software such as a terminal program is required when using POLL mode. Data is in ASCII format with 8 data bits + 1 stop bit. When polling the instrument it is essential that the command characters are sent with less than a 10mS delay between them. This normally means that each command line must be sent as a whole string e.g. a command such as $< STX > PA < CR >$ is sent as one string rather than $< STX >$ on one line followed by P etc. If testing using a "terminal" program or other software this is normally achieved by allocating a command string to a function key. Whenever the function key is operated the whole string is sent. The format used is ASCII (8 data bits + 1 stop bit) so, for instance, if address 1 is used then the string $< STX > PA < CR >$ must be put into the terminal program as: ∧BP!∧M where:

∧B	is the ASCII character for $< STX >$ (2 Dec, 02 Hex)
P	is the command line to transmit the primary display value (80 Dec. 50 Hex)
!	is the ASCII character for address 1 (33 Dec of 21 Hex)
∧M	is the ASCII character for $< CR >$ (13 Dec. 0D Hex)

Typical formats for the host command is as follows:-

$< ST > CA < CR >$ (Standard read etc.)
$< STX > CA < CR > N < CR > XYYYYYY$ (Set Value Command)

Where:	$< STX >$	is Start of Text Character (2 Dec, 02 Hex, ∧B ASCII)
	C	is the command character (see following commands)
	A	is the unit address (Range: 32 to 63 Dec, 20 to 3F Hex, "SPACE" to ? ASCII the address is offset by 32 Dec, 20 Hex)
	$< CR >$	is Carriage Return (13 Dec, 0D Hex, ∧M ASCII)
	N	is the setpoint number in ASCII e.g.: 1 for alarm 1 etc.
	X	SPACE for positive and "-" for negative
	$YYYY$	YY is the setpoint value in ASCII

The POLL commands available and instrument responses are as follows:

1. Transmit primary display value: $< STX > PA < CR >$
   e.g. ∧BP!∧M using a terminal program (address 1). Instructs unit to return the primary display value. The primary value is the channel 1 display value. Format of returned data is: $< ACK > PAXYYYYY < CR >$

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex)
	P	echo command received "P" (80 Dec, 50 Hex)
	A	is the responding units address
	X	SPACE for positive and "-" for negative
	$YYYYY$	is the display value in ASCII
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex)

   If the decimal point is non zero then it will be sent in the appropriate place as "." (46 Dec, 2E Hex).

2. **Transmit secondary display value:** $< STX > SA < CR >$
e.g. ∧BS!∧M using a terminal program (address 1). Instructs unit to return the secondary display value which is the remote input value. If no remote input function is set then the secondary value is the same as the primary value. If a remote input is set for **Hi** , **Lo**, **Hi Lo**, **P.HLd** or **d.HLd** then the value for the selected function will be returned e.g. if set to **Hi Lo** the high value followed by the low value will be sent (separated by a comma). Format of returned data is: $< ACK > SAXYYYYY < CR >$

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex)
	S	echo command received "S" (83 Dec, 53 Hex)
	A	is the responding units address
	X	SPACE for positive and "-" for negative
	$YYYYY$	is the value in ASCII
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex)

3. Transmit tertiary display value: $< STX > TA < CR >$
e.g. ∧BT!∧M using a terminal program (address 1). Instructs unit to return the tertiary display value. The tertiary value is the temperature display value. Format of returned data is: $< ACK > TAXYYYYY < CR >$

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex)
	T	echo command received "Q" (84 Dec, 54 Hex)
	A	is the responding units address
	X	SPACE for positive and "-" for negative
	$YYYYY$	is the display value in ASCII
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex)

If the decimal point is non zero then it will be sent in the appropriate place as "." (46 Dec, 2E Hex).

4. Transmit quadranary display value: $< STX > QA < CR >$
e.g. ∧BQ!∧M using a terminal program (address 1). Instructs unit to return the quadranary display value. The quadranary value is the channel 2 display value. Format of returned data is: $< ACK > QAXYYYYY < CR >$

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex)
	Q	echo command received "Q" (81 Dec, 51 Hex)
	A	is the responding units address
	X	SPACE for positive and "-" for negative
	$YYYYY$	is the display value in ASCII
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex)

If the decimal point is non zero then it will be sent in the appropriate place as "." (46 Dec, 2E Hex).

5. **Read low alarm setpoint:** $< STX > LA < CR > N < CR >$
e.g. ∧BN!∧M2∧M to read alarm 2 low setpoint using a terminal program (address 1). Instructs unit to return the low alarm setpoint value. Format of returned data is:
$< ACK > LANXYYYYY < CR >$

Where: $< ACK >$ is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
L echo command received "L" (76 Dec, 4C Hex)
A is the responding units address
N is the relay number in ASCII
X SPACE for positive and "-" for negative
$YYYYY$ is the setpoint value in ASCII
$< CR >$ is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

6. **Read high alarm setpoint:** $< STX > HA < CR > N < CR >$

   e.g. ∧HN!∧M2∧M to read alarm 2 high setpoint using a terminal program (address 1). Instructs unit to return the high alarm setpoint value. Format of returned data is: $< ACK > HANXYYYYY < CR >$

   Where: $< ACK >$ is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
   H echo command received "H" (72 Dec, 48 Hex)
   A is the responding units address
   N is the relay number in ASCII
   X SPACE for positive and "-" for negative
   $YYYYY$ is the setpoint value in ASCII
   $< CR >$ is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

7. **Set low alarm setpoint:** $< STX > lA < CR > N < CR > XYYYYY < CR >$
   e.g. ∧lN!∧M1∧M1000∧ to set alarm 1 low setpoint to 1000 using a terminal program (address 1). Instructs unit to set the low alarm setpoint value.
   Format of returned data is: $< ACK > lANXYYYYY < CR >$

   Where: $< ACK >$ is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
   l echo command received "l" (108 Dec, 6C Hex)
   A is the responding units address
   N is the relay number in ASCII
   X SPACE for positive and "-" for negative
   $YYYYY$ is the setpoint value in ASCII
   $< CR >$ is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

8. **Set high alarm setpoint:** $< STX > hA < CR > N < CR > XYYYYY < CR >$
   e.g. ∧hN!∧M1∧M5000∧ to set alarm 1 low setpoint to 5000 using a terminal program (address 1). Instructs unit to set the high alarm setpoint value. Format of returned data is: $< ACK > hANXYYYYY < CR >$

   Where: $< ACK >$ is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
   h echo command received "h" (104 Dec, 68 Hex)
   A is the responding units address
   N is the relay number in ASCII
   X SPACE for positive and "-" for negative
   $YYYYY$ is the setpoint value in ASCII
   $< CR >$ is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

9. **Transmit instrument model and software version:** $< STX > IA < CR >$ e.g. ∧BI!∧M using a terminal program (address 1). Instructs unit to return the instrument model and software version.
   Format of returned data is: $< ACK > IACCX.X < CR >$

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
	I	echo command received "I" (73 Dec, 49 Hex)
	A	is the responding units address (offset by 32 Dec e.g. "!" is address 1)
	CC	a 2 character identifier e.g. LC means loadcell input
	$X.X$	is the software version number e.g. 4.6
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

10. **Invalid command:** If the command received from the host is invalid the unit will return the following: $< ACK >?A < CR >$, where:

Where:	$< ACK >$	is Acknowledge (6 Dec, 06 Hex, ∧F ASCII)
	?	echo command received "?" (63 Dec, 3F Hex)
	A	is the responding units address (offset by 32 Dec e.g. "!" is address 1)
	$< CR >$	is a Carriage Return (13 Dec, 0D Hex, ∧M ASCII)

3.5 Data logger

The data logger is an optional addition to the instrument. This section applies only to instruments fitted with the data logger option. If the data logger is being used with the Windows compatible software provided then refer to the separate "Download Software User Guide" booklet.

Operation of the data logger
The data logger memory will store the hours:mins:secs, day:month and year together with the Channel 1, Channel 2 and Temperature readings at the time of log update. The log update time may be set at the **LoGUPdt** function. If an input is overranged when logged then the overrange value (**----**) will be logged for that channel for as long as the overrange value is present. Readings taken during power failure will not be logged. The log memory is set up in a circular format. Once the top of memory is reached the log data will overwrite the start of memory (overwriting the oldest record). The recording time available will vary depending on the memory size fitted and the update time selected. The table below shows maximum recording times.
Data is transmitted in comma separated format making it compatible with many commercially available databases/spreadsheets. Time information is downloaded in Julian time format which is again compatible with many databases/spreadsheets. The internal clock is battery backed. Downloaded log records are in the form of the time followed by the logged record for each channel at that time.
Downloaded information is transmitted via the serial output option board in RS232 or RS485 format, thus a serial output option must be fitted on all instruments with data logging software.

Data logger Windows software
Data logger software compatible with Windows 95, 98, 2000, NT and XP is provided for use with the data logger (not tested and may not be compatible with Vista). A separate user booklet for the software is also provided. Consult this user manual for details of software setup. The data logger can also communicate using standard serial polling commands, these are listed under the heading "Serial Command Format" in this chapter.

TP4-PH datalogger table - maximum logging times (approximate)

Time between logs	**32k memory** days:hours:min	**128k memory** days:hours:min
10 seconds	0:07:31	1:06:07
20 seconds	0:15:03	2:12:16
30 seconds	0:22:36	3:18:24
1 minute	1:21:12	7:12:48
2 minutes	3:18:24	15:01:48
3 minutes	5:15:36	27:14:24
4 minutes	7:12:48	30:03:12
5 minutes	9:10:00	37:16:00
6 minutes	11:07:12	45:04:48
10 minutes	18:20:00	75:08:00
15 minutes	28:06:00	113:00:00
20 minutes	37:16:00	150:16:00
30 minutes	56:12:00	226:00:00
60 minutes	113:00:00	452:00:00

Data logger polling functions
Usually data is downloaded using the Windows program supplied with the data logger but the data logger can be also polled via a PC etc. using the commands below. Functions which are used when

the data logger option is fitted are accessible only via CAL mode.

LOG UPdt (select log update time) Displays and sets the time period between each log sample. Available selections are: 0.10 (10 seconds), 0.20 (20 seconds), 0.30 (30 seconds), 1.00 (1 minute), 2.00 (2 minutes), 3.00 (3 minutes), 4.00 (4 minutes), 5.00 (5 minutes), 6.00 (6 minutes), 10.00 (10 minutes), 15.00 (15 minutes), 20.00 (20 minutes), 30.00 (30 minutes) or 60.00 (60 minutes).
Note: The data log memory (see CLr LOG below) must be cleared whenever the log update time is changed or the date and time is changed.

CLr LOG (clear data log memory) This function clears the data log memory, to clear the memory press then release ▲ and ▼ simultaneously, the display will show CLr ? asking if you really want to clear the memory. If you wish to clear memory then press then release ▲ and ▼ simultaneously again. The log memory will then be cleared and the log period reset, the display will indicate PrOG LOG to confirm this. Once the memory is cleared all previously logged records will be lost from the instruments memory, if the CLr ? message is reached and it is not wished to clear the log memory then pressing and releasing either F or P will abort the function.

SEt rtc (set time) Displays and sets the current time in hours and minutes (24 hour format HH.MM) e.g. set as 17.20 for 5:20 pm.

SEt dAtE (set date) Displays and sets the current date in days and months (DD.MM format). The months will roll over automatically (up at the end of the month, down at the beginning of the month) as the day is scrolled up or down.

SEt YEAr (set year) Displays and sets the current year (YYYY format). Valid years settings go up to 2037 (valid Julian time format years).

Serial Command Format
Instruments using the data logger option are provided with extra software functions to the standard instrument. This section describes these extra functions.

Initial Setup
Select the baud rate, parity and address as required. The serial output mode function (0.Put) must be set to POLL when using the data logger.

Transmit Record Block: <STX>DA<CR>D<CR>TTTTTTTTTT<CR>NNNN<CR>
Where: TTTTTTTTTT is the start time of the block (in Julian time format). NNNN is the number of records to be sent. Instructs the unit to send a block of logged data via the serial interface. The returned data format is: <ACK>DAD<CR> followed by NNNN records in the format:-
TTTTTTTTTT,S1111,S2222,S3333,S4444,S5555,S6666,S7777,S8888<CR> where:

TTTTTTTTTT is the start time for each record (in Julian time format). If TTTTTTTTTT (time in "Transmit Record Block" request) is sent as 0 then the records will start at the earliest time in log memory.

S is the sign (<SPACE> for positive values and "-" for negative.)

1111, 2222 etc. are the values for each channel.

Values will only be transmitted for active channels. Invalid readings from any channel will be received as the overrange value (----) for that channel. If the start time requested is not present in the log then <ACK>DA?<CR> will be returned.

Transmit All Logged Data: <STX>DA<CR>A<CR>

Instructs the unit to transmit the entire data log. All log records since the last log memory reset will be sent to the host. The unit will respond with <ACK>DAA<CR> followed by all log record sent in the same format as above (Transmit Record Block)

Transmit System Time: <STX>DA<CR>T<CR>

Instructs the instrument to transmit the current time in Julian time format as follows:- <ACK>DAT TTTTTTTTTT<CR>

Transmit the Log Start Time: <STX>DA<CR>S<CR>

Instructs the instrument to transmit the log start time i.e. the time stamp on the first record in the log. Note that if the memory has "wrapped around", i.e. has started to overwrite existing logged records, that the log start time will not be the original time the log started (since this time stamp and associated log record has been overwritten). The returned data format is: <ACK>DAS TTTTTTTTTT<CR>

Transmit the Log Update Time: <STX>DA<CR>U<CR>

Returns the current log update time as set in the log memory. The returned time may be different to the **dLAY** time if there has been no log reset since the **dLAY** function was changed. The returned data format is: <ACK>DAU NNNN<CR>, where NNNN is the update time in seconds.

Transmit the Log Memory Size: <STX>DA<CR>M<CR> Returns the size of the log memory in records. The returned data format is: <ACK>DAM NNNN<CR>, where NNNN is the number of records for that memory size e.g. an 8K memory will return 508.

Set the System Time: <STX>DA<CR>t<CR>TTTTTTTTTT<CR>

Set the instrument system clock to Julian time TTTTTTTTTT. If the command is successful then <ACK>DAt<CR> will be returned. If the Julian time is invalid then <ACK>DA?<CR> will be returned.

Set the Log Update Time: <STX>DA<CR>u<CR>NNNN<CR>

Set the log update time to NNNN seconds. Note that the new time will not apply until a log reset is performed. If the command is successful then <ACK>DAu<CR> will be returned. If the update time is invalid then <ACK>DA?<CR> will be returned. Valid times are as shown in the **dLAY** function explanation.

Reset the Log Memory: <STX>DA<CR>R<CR>RESET<CR>

This command will reset the log memory. This will erase all current records and reset the log update time if it has changed. As this will result in a loss of data the command must be sent exactly as it appears or the memory will not be reset. If the command is successful then <ACK>DAR<CR> will be returned to indicate that the memory has been reset. If the command is invalid then <ACK>DA?<CR> will be returned.