

---

# Appendix D

---

## Practical Exercises for IDC Process Control Workshop

### D.1 Introduction

Our practical exercises are based on the use of the process control training and simulation package PC-ControlLAB supplied to IDC Technologies by Wade Associates Inc.

IDC Technologies acknowledges with thanks the major contribution provided to this training course by the facilities available in this package

The following is a list of the practical sessions we will select dependent on the profile of the class.

Chapter	Exercise
1	Exercise 1 - Flow control loop - basic example for open loop control
4	Exercise 2 - Flow control loop - basic example for closed loop control
5	Exercise 3 - Proportional (P) Control- Flow Chart
5	Exercise 4 - Integral (I) Control - Flow Control
5	Exercise 5 - Proportional and Integral (PI) Control - Flow Control
5	Exercise 6 - Introduction to Derivative (D) Control
5 & 7	Exercise 7 - Practical Introduction into Stability Aspects
1 & 5	Exercise 8 - Further exercises in identifying process responses
8	Exercise 9 - Open Loop Method -Tuning Exercise
8	Exercise 10 - Closed Loop Method -Tuning Exercise
8	Exercise 11 - Exercise in improving 'as found' tuning (damped cycle)
9	Exercise 12 - Cascade Control
9	Exercise 13 – Ratio control
11	Exercise 14 - Combined Feedback and Feedforward Control
11	Exercise 15 - FF and FB Applied to boiler drum level control
12	Exercise 16 - Dead time Compensation in Feedback Control
14	Exercise 17 - Gain scheduling for control of non-linear process

Each of the above practical exercises is described in the following pages in this appendix.

## Before you start

To get started with the exercises you will first need to install PC ControlLAB and the load the exercise data files all as described in Appendix C.

The following is a reference list of the data files required for the exercises.

Chapter	Exercise No	PC Pracs Folder	File Name
1	1	EX-1	IDC Flow 1.mdl
4	2	EX-2	IDC Flow 1.mdl
5	3	EX-3	IDC Flow 2.mdl.
5	4	EX-4	IDC Flow 4.mdl
5	5	EX-5	IDC Flow 4.mdl
5	6	EX-6	IDC Feed Htr.mdl.
5 & 7	7	EX-7	IDC Feed Htr.mdl
1 & 5	8	EX-8	Wade: Folpt.mdl. etc
8	9	EX-9	IDC Feed Htr Ex-9.mdl
8	10	EX-10	Wade: Generic.mdl
8	11	EX-11	Wade: Generic.mdl
9	12	EX-12	IDC Feed Htr casc.mdl.
9	13	EX-13	Wade: ratioflo.mdl
11	14	EX-14	IDC FEEDFWD HTR.mdl
11	15	EX-15	Wade:level2.mdl
12	16	EX-16	IDC Ex 16.mdl
	17	EX-17	Wade: Temp.mdl

The data files provide the process model for each exercise and in some cases strategy files are used to preload controller settings. Details of how to start up each exercise are contained in the instruction steps given in the practical exercise sheets starting overleaf.

---

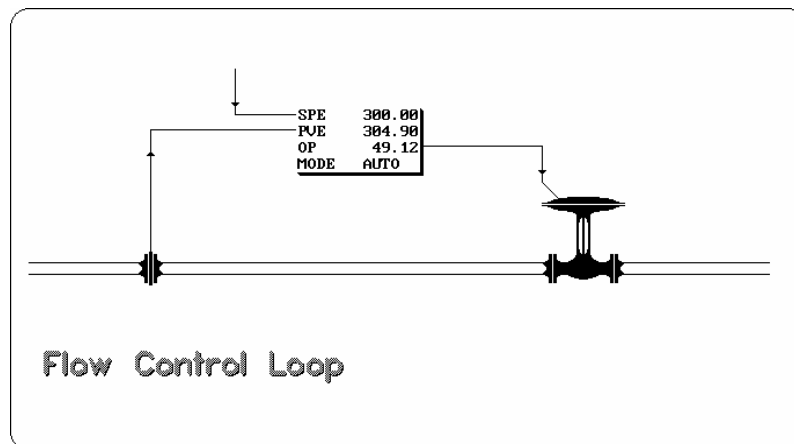
# Exercise 1

---

## Flow Control Loop - Basic Example for Open Loop Control

### D1.1 Objective

This exercise will familiarize the student with the basic concepts of open loop and closed loop control. It provides an opportunity to get a first feel for the open loop response of the process being controlled. A flow control loop as shown in figure 1.1 below will serve as a practical example for this exercise. A flow control loop is generally not difficult to operate and illustrates the basic principles effectively.



**Figure D1.1**  
*Basic Flow Control Loop*

## D1.2 Operation

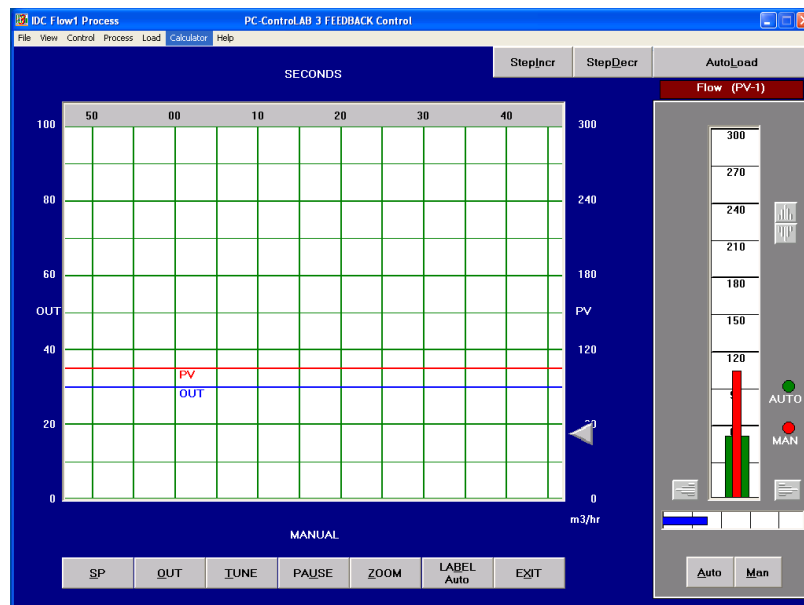
Since this is a relatively simple exercise, it can be used for familiarization with the principles of operation of the simulation software: “PC Control-Lab”. Firstly, you must ensure that you have installed the PC Lab software by following the instructions provided in the introduction. You should also have completed some of the first parts of the tutorial (Under HELP) to ensure you are familiar with the basic controls and operations. Then proceed with the following steps.

Step 1: Start the program in MS Windows by running PC-ControlLab

Step 2: Open the project file for IDC Flow 1.mdl which is a pre-configured flow control loop with first order lag and dead time.

- Click on **Process | Select Model**.
- Select the file location “PC-Pracs” and then subfolder “Ex-1” Highlight “**IDC Flow 1.mdl**” and press **Open**.
- Set the **Horizontal grid scale** to “**Seconds**” (under View). NB: Most exercises in this workshop use the SECONDS scale. This has to be selected at the start of each exercise
- Set the LOAD value to 65% on screen by pressing STEP INCR
- Set the controller mode to **Manual**
- **Use the OUT key to set the OUT value = 30%**

You should now have a display on screen that simply indicates the output value to the control valve is steady at 30% and the PV is steady at 105 m<sup>3</sup>/hr as per Figure D1.2 below



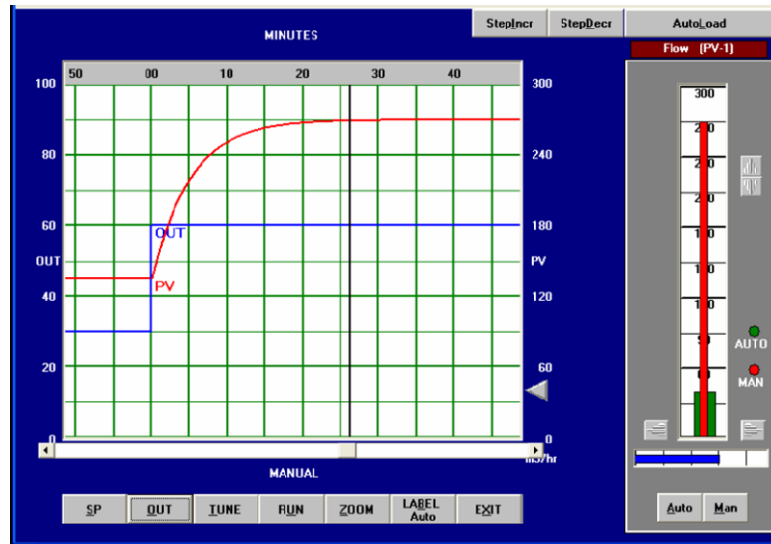
**Figure D1.2**

*Initial display for flow loop manual mode testing showing trend display (left) and controller faceplate display (right).*

This display gives a general idea of the process and displays the major variables PV (process variable scaled 0 to 300 m<sup>3</sup>/hr) and OP (controller output signal to flow control valve scaled in range 0 to 100%). The SP (set point) is shown as an arrow on the right of the trend screen and as a red bar on the controller faceplate.

Note that the operating data with PV, SP and OP appears in a box on the left side whenever you press the “OUT” button. PV data appears in a similar box whenever you press the “PAUSE” button. The trend record screen must be set to a scale in seconds to suit the process model we are going to test.

Firstly, we will observe the general behavior of the process as will be seen on the trend display when the RUN control is pressed. In order to observe the process reaction, as a result of changes in the position of the control valve, keep the control mode in MANUAL with OP=30% and when PV is steady press PAUSE. Then, press OUT and change the value of the output to 60% and press OK. Now press the RUN button to see the response develop on the trend display.



**Figure D1.3**  
*Flow process reaction to step change in output manual mode*

## D1.3 Observation

Observe the Process Variable PV as it changes in value due to the change in OP. There is a basic first order lag response between the output to the valve and the change in flow measured as the PV. This is an idealized response that has been configured into the simulation using the “Builder” tool supplied with the simulation package. This lag represents the response lag of the valve opening as the actuator and positioner respond to follow the controller output signal as well as the lag in the flow sensing stage.

Two very important process parameters can already be extracted from this response, which is the basis of an “open loop test” of the process. These are:

- 1) The open loop gain of the process
- 2) The time constant of the first order lag

**Find the open loop gain  $K_p$  by measuring:**

$$K_p = \frac{\% \text{ change in PV}}{\% \text{ change in OP}}$$

Because there is a control valve involved you cannot be sure that the process gain will be the same at all flow rates through the system. Therefore the value of  $K_p$  should be tested at different flow rates.

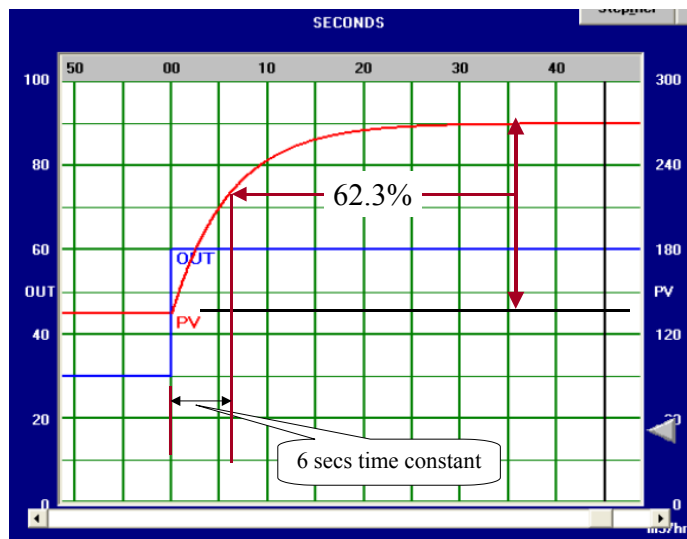
Carry out the following gain test in 20% increments starting at OP=10%

- 1) Record PV(10) when OP = 10%
- 2) Increase OP to 30% and allow the PV to stabilize
- 3) Record PV (30) when OP = 30%
- 4) Calculate Kp from the above equation.
- 5) Repeat steps 1 to 4 for (30% to 50%) and (50% to 70%).

Remember to change the value of PV from the engineering scale of 300m<sup>3</sup>/hr to the percentage of full scale value

You should find that the gain Kp is reasonably constant = 1.5 over the range of operation. Very often this is not the case of course, particularly at the low and high ends of the control valve opening but for the moment we have set up an approximately linear relationship between 0 and 300m<sup>3</sup>/hr.

**Now find the time constant of the first order lag** in the process (process includes the valve). You can do this by repeating the step response from 30% to 60% OP in Manual and observing the time trace.



**Figure D1.4**  
*Finding the time constant of the first order lag*

The lag is found by measuring the time from the initial movement of the PV until the PV reaches 62.3% of the new steady state value. Once you have most of the curve on screen press the PAUSE button and use the cursor on the trend display to read off the PV and time values. Record the original PV as PV1 and the new steady state value of PV as PV2. Then calculate the target PV point for the time constant as:

$$PV_{tc} = PV1 + 62.3\% (PV2 - PV1).$$

In the standard example you should be able to show that the time constant is 5.0 seconds as shown in Figure D1.4. Note that the response curve reaches approximately 99% of its final value after 4 time constants.

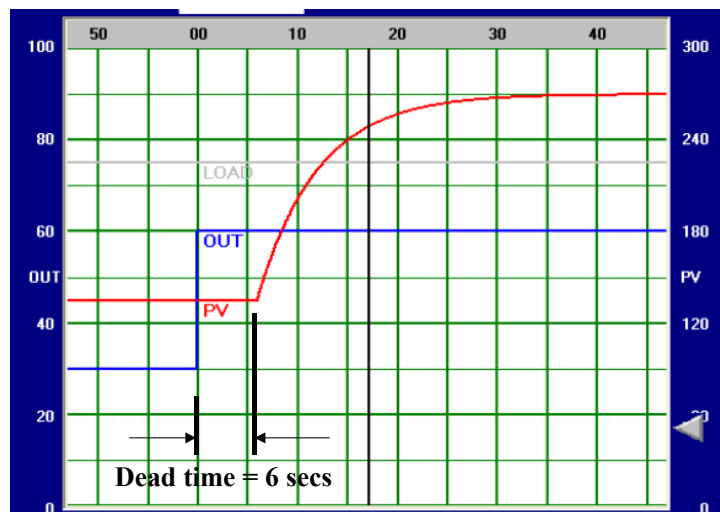
### Changing parameters

The PC-Control Lab software allows you to inspect and change most parameters of the process being simulated. You should now experiment with changing and testing the gain and time constant values in the model by clicking on the command “ Process—Change Parameters”. Repeat the above the above tests to verify your chosen values.

**Inserting dead time.**

**Caution:** Dead time parameters in PC-Lab are stated in minutes. For example 0.1 minutes = 6seconds. Do not raise dead time above 1.0 minutes when operating in with the display grid set to SECONDS as this may cause corruption of the simulation. If this occurs simple reload PC-Lab.

Most processes exhibit some degree of dead time or “transport delay” in which the change in output to the process (more correctly the manipulated variable MV) has no effect at all on the PV until the change has had time to travel through a time period. (Opening the floodgates on a dam will not affect downstream residents until the surge has traveled to their neighborhood). In this simulation you may insert and observe the effect of dead time on a step change by increasing the parameter value from 0.01 to 1.0 or even higher. Figure D1.5 shows an example when dead time has been set to 6 seconds (0.1 minutes). This will make any a control loop much more difficult to stabilize so it should be restored to the minimum value of 0.01 before proceeding with the next stage of this practical.



**Figure D1.5**  
Response to step change with dead time in the process

## D1.4 Open loop control

You can observe that without the advantage of automatic feedback from the PV your ability to drive the PV to a target value depends on knowing the gain of the process and zero offset or knowing what loading is placed on it.

Loading is the general term used in the simulation for an external disturbance on the process or for a measurement offset. The load on this particular process is the system pressure drop across the valve and flow meter. The normal pressure drop applies when the load parameter, displayed on the screen as a grey line, is at 50%. To observe the effect of a reduction in system pressure:

- 1) Set the OP to 30% and allow the PV to stabilize, then just press the *StepDecr* button.
- 2) Step the pressure down again until the grey line is on 40%. Observe that the flow falls each time as you might expect. The change in system pressure drop has changed the gain of the process

Now adjust the OP value to deliver  $PV = 180 \text{ m}^3/\text{hr}$ . Now increase the pressure drop by pressing the *StepIncr* button and observe the change in PV. Clearly, unless you know the exact amount of the disturbance you cannot easily restore the PV to  $180 \text{ m}^3/\text{hr}$  unless you use feedback from the PV.

If the disturbance is caused by random noise in the process (in this case it could be a supply pressure deviation or a downstream pressure deviation) or in the measurement you will also not be able to maintain the PV at the target value by open loop control. For example: Insert noise by pressing the *AutoLoad* button and observe the random movement of the PV.

### **Conclusion for open loop control**

Open loop control can only be used when the process characteristics are unchanging and undisturbed by external factors or if these factors can be accurately compensated out. In all other cases predicting the required output to achieve the desired PV will give poor results.



---

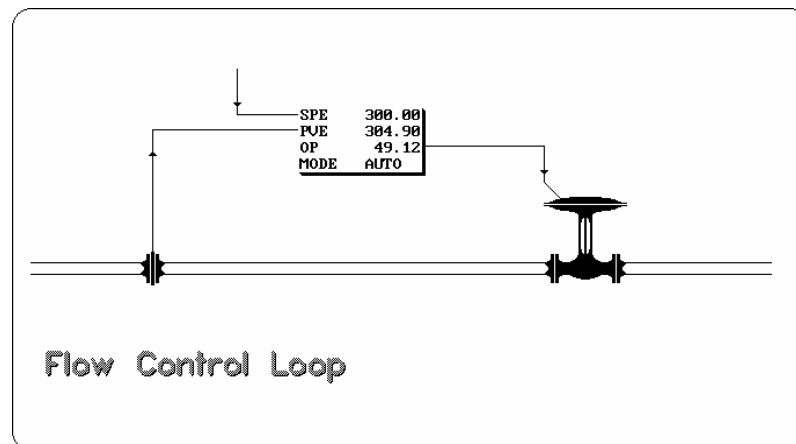
## Exercise 2

---

# Flow Control Loop - Basic Example for Closed Loop Control

### D2.1 Objective

This exercise will familiarize the student with the basic concepts of closed loop control. It provides an opportunity to get a first feel for closed loop control and the open loop response of the process being controlled. A flow control loop as shown in Figure D2.1 will serve as a practical example for this exercise. A flow control loop is generally not difficult to operate and illustrates the basic principles effectively.



**Figure D2.1**  
*Basic Flow Control Loop*

### D2.2 Operation

This exercise follows directly from Exercise 1 in which you will have seen how to install and operate the PC ControlLab simulation tool. Also in Exercise 1 you will have loaded the model IDC Flow1.mdl and tested its open loop response for gain and time constant. Make sure that you have completed these steps before proceeding with this exercise.

## D2.3 Operation in Auto mode

Prepare the model IDC Flow 1 for closed loop control by loading it from file folder Ex-2. Select the file location “PC-Pracs” and then subfolder “Ex-2” Highlight “**IDC Flow 1.mdl** ”and press **Open**. Set the grid scale to seconds.

For some initial practice with the automatic control mode some suitable loop tuning values for controller gain and integral (reset) action must be loaded. Press the “Control” key on the toolbar and select the dropdown for “Retrieve strategy and tuning”. Select the file location to Ex-2, select: “IDC Flow Auto 1.stg” and open it. Now press the TUNE key and verify that you have Gain = 1.5 and Reset = 0.2 minutes per repeat.

The normal practice in setting a control loop to automatic is to first make sure that the set point (SP) is made equal to the present PV to avoid any disturbance as the automatic control action is started. Set point tracking has already been enabled in this strategy model so if you set the output to a starting value the set point will self-adjust to match the PV. Execute the following steps.

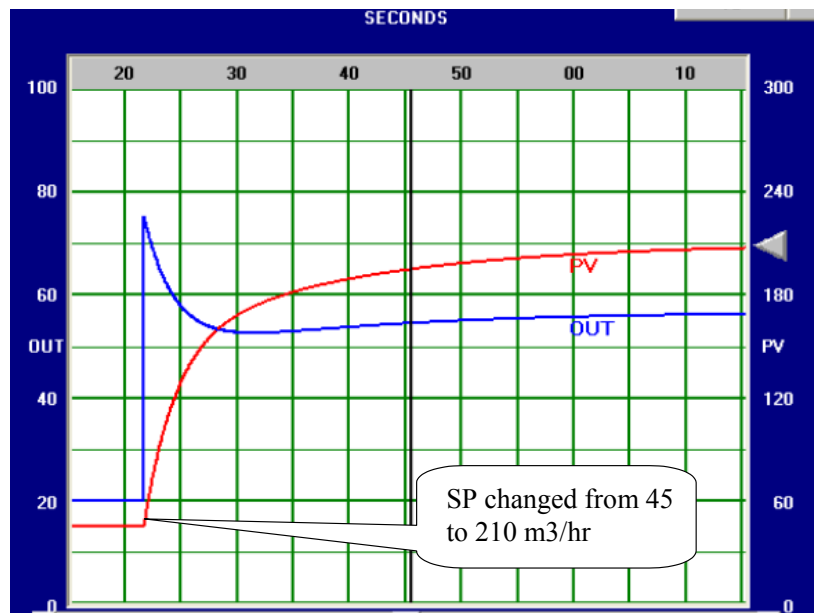
Step 1: Keep the controller in MAN and the display in RUN. Set OP = 20%.

Step 2: Note that the PV and SP settle at 45 m<sup>3</sup>/hr.

Step 3: Change the controller from MANUAL to AUTOMATIC mode.

Step 3: Change the set point to 210 m<sup>3</sup>/hr by using the SP button and typing in the new value.

Step 4: As soon you press OK you will see the response to this step change in the set point. You should obtain the response shown in Figure D2.2.



**Figure D2.2**  
*Basic flow loop auto response to step change in SP*

## D2.4 Observation

As soon as the set point is changed the controller responds by driving the OUTPUT to reduce the error between SP and PV and the PV is steadily driven to the new Set Point. The controller tuning has been set up for proportional plus integral action control, hence you can see in Figure D2.2 that the proportional action brings the PV towards the SP but does not meet it until the integral action has increased the output over time until the error is zero.

## D2.5 Operation

You should now experiment on your own with changing the tuning parameters and repeating step changes to see the effect on the responses. The tuning parameters we are concerned with here are simple the controller gain and the integral action. Press TUNE on the display to access these parameters. If you want to get back to the settings used for the test run in Figure D2.2 simple reload the strategy file using the command CONTROL- RETRIEVE STRATEGY,MODEL and TUNING.

Since a flow control loop has no intrinsic stability problems, most effects can be observed clearly. You may also try introducing some dead time into the model as discussed in D2.3 above. Find how much dead time can be tolerated before the response becomes unstable.

This completes Exercises 1 and 2. By this stage you should be quite familiar with the basic features of the PC-Control Lab 3 facility.



---

## Exercise 3

---

# Proportional (P) control

### D3.1 Objective

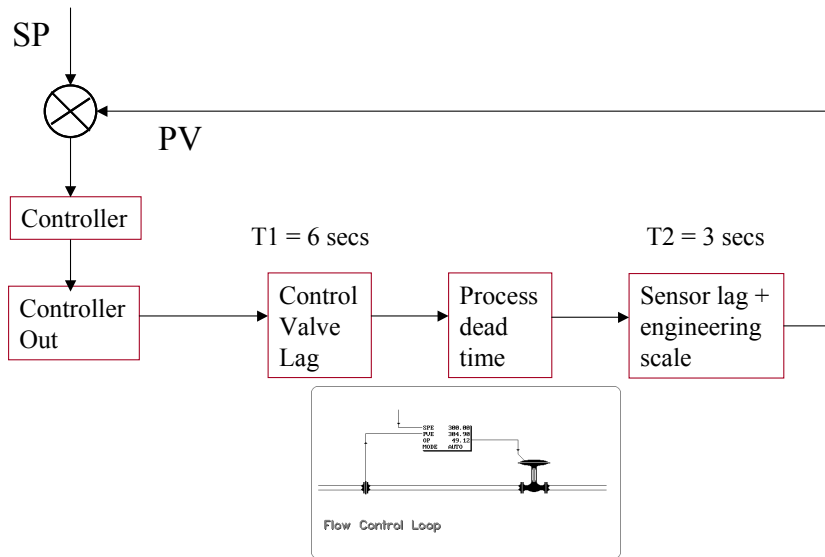
This exercise will introduce the main control action of controllers - Proportional Control. Special emphasis is placed on the fact that there is a remaining offset condition, if proportional control is used solely. We shall see that a steady state offset between Set point and Process variable exists whenever proportional only control is used in a non-integrating process such as flow control. We shall see that this offset reduces as the control loop gain is increased but it will also be seen that instability will eventually occur as gain values are increased.

### D3.2 Process model

This exercise will use the same basic example for closed loop flow control that we used in Exercise 1. There is however a slightly more realistic model is to be used for this and subsequent related exercises. The process is shown in block diagram form in Figure D3.1. The output of the controller is applied to a first order dynamic lag stage representing the control valve, which also has a stage gain of 1.5. There is a dead time element representing small transport delays in the valve and in the process and then there is another dynamic lag stage representing the flow sensor with a certain amount of noise filtering. Effectively the overall process behaves as two dynamic lags in series plus a very small amount of dead time. This is very typical of a practical flow control loop and it means that the process behaves as very well damped 2nd order system until its loop gain becomes quite high.

The initial parameters used for the model in this exercise were:

- Valve lag time constant: 0.2 minutes
- Dead time: 0.001 minutes
- Sensor lag time constant: 0.05 minutes



**Figure D3.1**  
Process Model for Flow control loop

To install this model in the PC-ControLab simulation start the simulator programme and then select: **Process / Select Model/ PC-Pracs/Ex-3/IDC flow 2.mdl**.

Note that this model can display PV in the range 0 to 100% or it can display PVE in the range 0 to 300 m<sup>3</sup>/hr. For this exercise we have used PV in 0 to 100% for simplicity. Therefore set the display range as follows: **View/Display Range/Percent of span**

## D3.3 Operations

### D3.3.1 Operations to demonstrate and calculate offset in proportional only control

Once you have opened IDCFlow2.mdl you should install the control strategy file for this exercise and proceed as follows:

**Step 1:** Select Control/Retrieve strategy and tuning/ IDC Flow prop2.stg. Set the scroll display rate to "seconds".

**Step 2:** Select Control options and confirm that the Control Algorithm has been set to Proportional Only.

**Control action in Auto will now be defined by the algorithm  $OP = K1 (SP - PV)$**

**Where  $K1$  = proportional loop gain. (i.e controller gain x process gain)**

**Step 3 :** Select TUNE and ensure that Gain =1 and Man Reset = 0.00. ( Man reset is used optionally to provide a fixed bias in the output when in manual mode).

**Step 4.** Set the LOAD to 75% by using the STEPINCR button.. (This offsets an internal bias of 75% within the model used for other tests).

**Step 5:** Check the relationship between OP and flow PV by setting the mode to MAN. Then check.

Op = 0% , PV = 0%  
 OP = 20 % , PV = 30%  
 OP = 50 % , PV = 75%

These figures will confirm the process gain (due to the valve) is 1.5. What you have now is a simple control valve arrangement where flow is zero when the valve is shut and where full range flow is achieved well before the valve is fully open. What valve opening is needed to achieve 100% flow? Note also in this model the valve gain is constant over the operating range and hence process gain does not change. This gain would be expected to change in most installations.

**Step 6:** Still in Man set OP = 20% and SP = 50%. Allow the simulation to run so that the PV value is steady at 30%.

**Step 7:** Switch the mode to AUTO. Note that no disturbance should occur because the deviation between SP and PV is 20% and the loop gain (K1) is 1.5.

$$\begin{aligned} \text{OP} &= \text{K1} (\text{SP}-\text{PV}) \\ &= 1 \times (50 - 30) \% = 20\% \end{aligned}$$

**Step 8:** Change SP to 80% and observe the response to a step change in the SP. Figure D3.2 shows the response you should expect on the simulation

Notice that the PV value settles down before reaching the value of SP. The remaining difference between SP and PV is called Proportional **Offset**. It can also be seen that the control action (OP) moves almost instantaneously (proportional control) in accordance with changes to PV or SP (Equation Type A). The step change of SP has caused the value of OP to make a step change accordingly. Immediately after this, one can observe the output to exactly follow the value of PV, but in reverse direction to the PV as the error (SP-PV) decreases.

Therefore, the trend display shows both the proportional change of output based on change of SP (Step change), followed by proportional change of output based on change of PV (Exponential approach to a steady value).

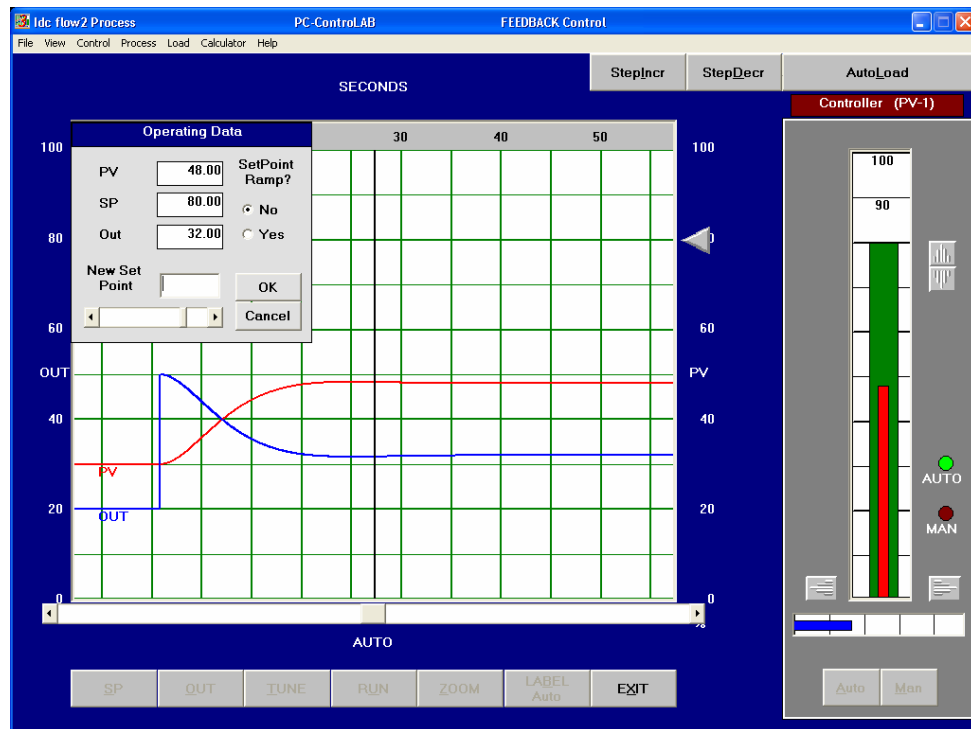
You can predict the new PV and OP values by using the above equation and by knowing the process relationship  $\text{PV} = \text{Gp} \times \text{OP}$ . where Gp is the process gain

$$\text{OP} = \text{PV}/\text{Gp} = \text{K1}(\text{SP}-\text{PV})$$

Re-arranged this states:  $\text{PV} = \text{SP} \cdot (\text{Gp} \cdot \text{K1} / (1 + \text{Gp} \cdot \text{K1}))$

So if K1 = 1, Gp = 1.5 and SP = 80% , PV = 48%

Therefore a steady state offset of 32 % occurs in this example.



**Figure D3.2**

*P only. Step change in SP with  $K_I = 1.0$*

**Step 9:** This step change exercise should now be repeated as you increase the controller gain  $K_I$  using the TUNE controls.

**Exercise :** Calculate the PV and offset between PV and SP for values of controller gain of 2, 5 and 10. For each gain setting create the same step response in AUTO from SP = 30% to SP = 80%. Record the PV values you obtain and compare with your predicted values.

**Observation:** Note how the response becomes increasingly unstable as the gain is increased. This is due to the 2nd order lag characteristic in the process model and places a limitation on the amount of gain you can safely apply. Hence the offset problem cannot be eliminated by proportional control only.





**Figure D3.3**

*P only. Step change in SP with  $K_I = 10$ . Overshoots developing as gain is increase. Offset is reduced.*

### D3.3.2 Operation to demonstrate the effect of longer lag times on response

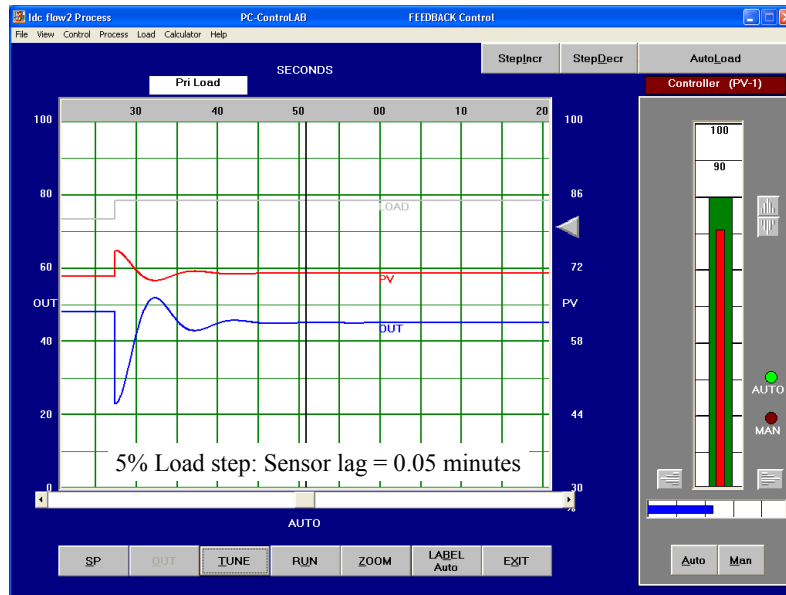
Repeat the above exercise with slightly longer lag times in the process model and using a controller gain of 5. Increase the lag time in the sensor by using the Process/change parameter command on the simulator.

Now is a good time to experiment with the Zoom facility in the simulator. We suggest you set the minimum PV display to 30% and leave the maximum at 100%.

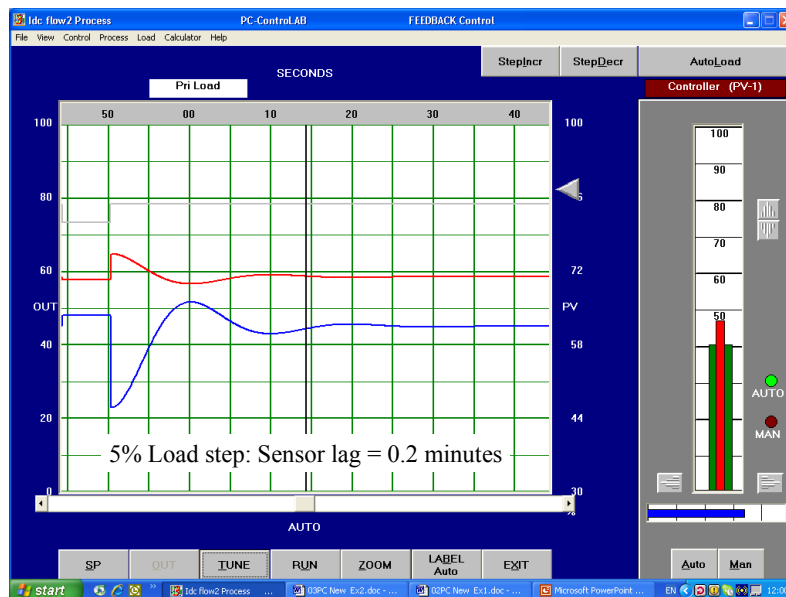
For this test use the “StepIncr” and “StepDecr” buttons to impose 5% load changes on the process. The damped overshoot responses can be seen in Figure D3.4 and Figure D3.5 where the lag time of the sensor has first been set to 0.05 minutes ( 3 seconds) and then changed to 0.2 minutes ( 12 seconds).

#### Observation

Observe that the overshoot responses do not increase in amplitude but the frequency is substantially reduced by the slower process characteristic. The stability is not affected because the proportional only control action does not introduce any phase shift in the response. This was not going to be the case when we revert to P+I control.



**Figure D3.4**  
*P only. Sensor lag = 0.05 min. Overshoot frequency approximately 10 seconds*



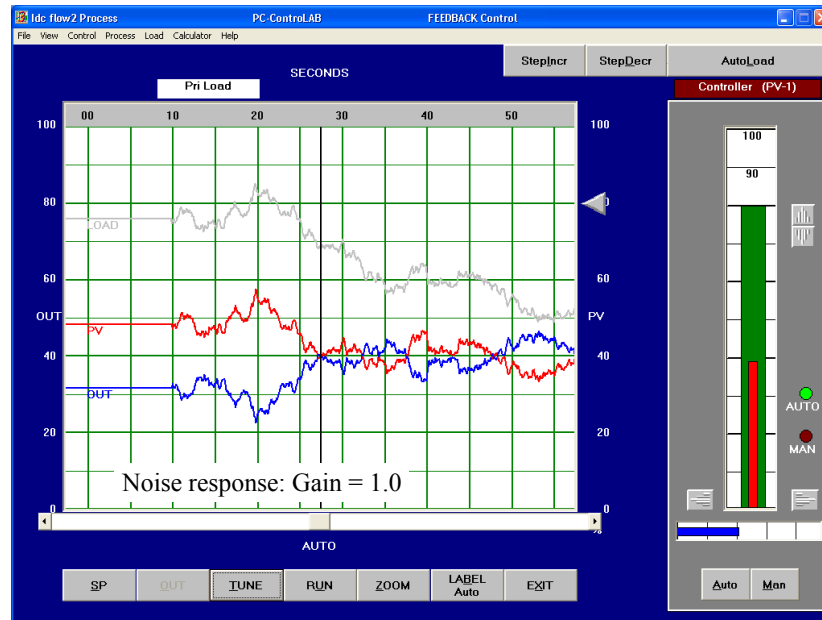
**Figure D3.5**  
*P only. Sensor lag = 0.2 min. Overshoot frequency approximately 20 seconds*

### D3.3.2 Operation to demonstrate the effect of process noise in high gain loops

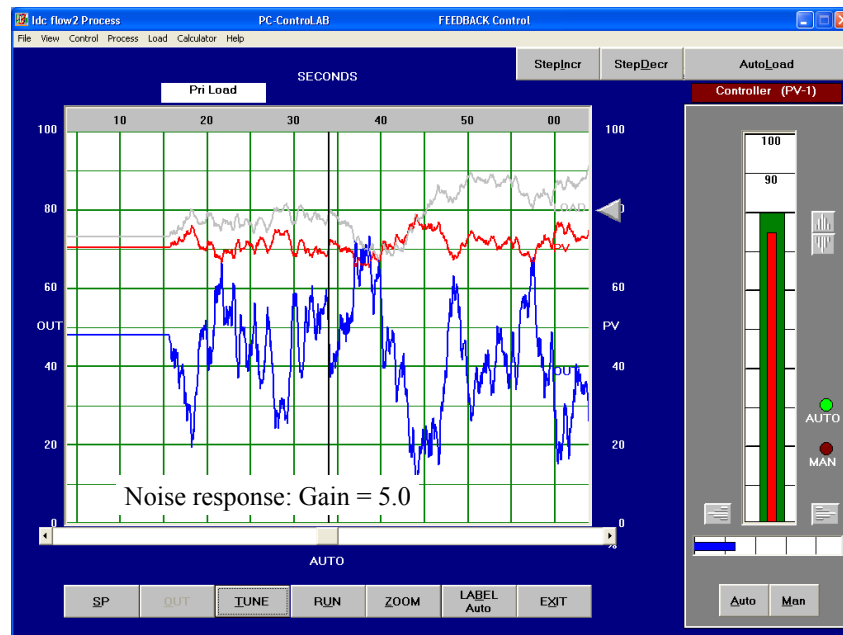
In this exercise we can try out the facility in the simulator to inject random disturbances into the process by using the “Auto load” button. This will demonstrate the effect of noise in the automatic control loop.

- Step 1:** Set the controller gain (TUNE button) to 1.0 and restore the process lags to 0.2 for the time constant and 0.05 for the sensor.
- Step 2:** Make SP = 80% adjust the load offset to approximately 75% as for the first exercise. Place the controller into Auto

- Step 3:** Once the PV is stable press the Auto Load button and see the degree of disturbance to the PV whilst the OP is changing moderately to try to correct for the errors.
- Step 4:** Increase the controller gain to 5.0 and see the increase OP disturbances whilst PV disturbance is not reduced very much.



**Figure D3.6**  
*P only. Sensor lag = 0.05 min. Gain = 1. Showing response to process noise*



**Figure D3.7**  
*P only. Sensor lag = 0.05 min. Gain = 5.0. Showing response to process noise*

## D3.4 Conclusion

This exercise has demonstrated a number of basic features of feedback control using the proportional-only algorithm.

- With increasing values of Gain, we obtain smaller values for OFFSET. This shows, that it may be desirable to use high values for Gain to minimize the Offset. However the offset can never be eliminated by proportional control.
- As gain increases, overshoots occur but in proportional-only control quite high gains may be used before complete instability occurs. The frequency of overshoot decreases with increased lag times as you would expect since the natural frequency of the process becomes slower.
- If a control loop has a tendency to be unstable, stability problems put limits on the increase of Gain. Even if no stability problem exists, the value of Gain should be kept as low as possible. This avoids unnecessary amplification of noise.

---

## Exercise 4

---

# Integral (I) Control - Flow Control

### D4.1 Objective

This exercise will introduce the Integral Control action of controllers. Special emphasis is given to the task of eliminating the remaining offset term of proportional control. It will also show the slower control action of Integral control, compared with Proportional control.

### D4.2 Process and control models

This exercise will use the same process model for closed loop flow control that we used in Exercise 3. Recall that this model has two first order lags representing the control valve and the process as well as a very small and usually negligible dead time element.

Note that this model can display PV in the range 0 to 100% or it can display PVE in the range 0 to 300 m<sup>3</sup>/hr. For this exercise we have used PV in 0 to 100% for best clarity.

The controller action for integral only control provides an output that is the time integral of the error between Sp and PV. Hence the output will continue to change as long as error exists and the controller will normally drive the PV to be exactly equal to the SP.

The control algorithm is described in the following box:

$$OP = \frac{1}{\tau_{int}} \int_0^T (SP-PV).dt + (\text{Initial OP at } T = 0)$$

Where  $\tau_{int}$  is the integral time in seconds or minutes  
 In controller tuning it is called the reset action and is specified as minutes per repeat (of error).

This is alternatively expressed as

$$\frac{1}{\tau_{int}} \text{ Repeats (of error) per minute}$$

For example: If  $Sp-PV = 10\%$  and  $\tau_{int} = 2$  minutes the output of the controller will change by  $10\%$  every 2 minutes.  
 Or we can say the reset rate will be  $10/2 = 5\%$  per minute.

### D4.3 Operations

Using the above information we can now set up the simulation using the Flow2 model to examine the integral action values and effects.

#### Test No 1: Open loop integral action.

- Step 1:** Start the simulator programme and then select: **Process / Select Model/ PC-Pracs/Ex-4/IDC flow4.mdl.**
- Step 2:** **Select Control/Retrieve strategy and tuning/ PC-Pracs/Ex-4/IDC Flow4 Int Only.stg.** Set the scroll display rate to “seconds” and set mode = MAN. Check that Control/Options/ Integral-only has been set. Press the **StepIncr** control to raise the Load value to 75% as observed on the trend screen.
- Step 3:** Check that SP Tracking is ON by going to Control/Control Options/SP Tracking.
- Step 4:** TUNE: Integral gain = 2.0. Manual Reset = 0.0
- Step 5:** Place the loop into AUTO and SP =30%. Allow the response to stabilize and then PAUSE the simulation. .
- Step 6:** Create a temporary open loop situation by increasing the dead time in the process to 1.0 minutes. (Change Parameter/Dead time).
- Step 7:** Change SP = 60% to create a 30 % control error. Then RUN and observe the integral action ramp. Calculate the expected ramp rate and check that this confirmed by the ramp rate response on the display. Check your result with Figure D4.1 overleaf.

### Test No 2: Close loop step response

**Step 1:** Change the dead time back to 0.01 to minimize its effect. .

**Step 2:** Place the loop into AUTO and SP =30%. Allow the response to stabilize and then PAUSE the simulation.

**Step 3:** Change the SP =60% again and repeat the step change. Observe the exponential approach of PV to the SP and note the slow speed of this response. Confirm with Figure D4.2.

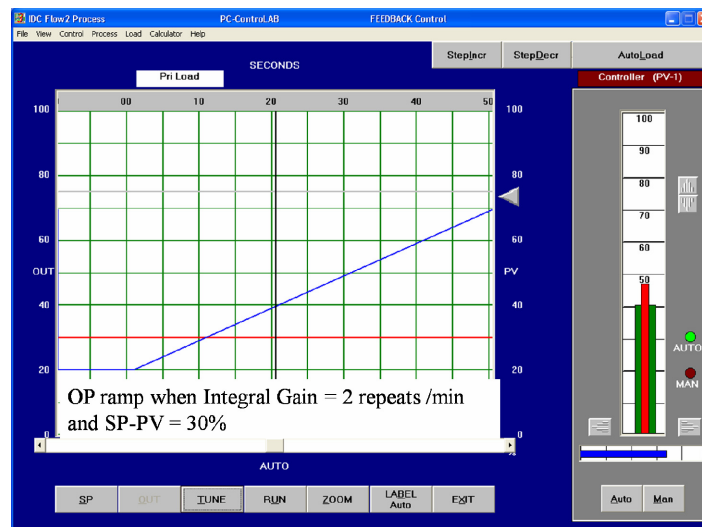
This test shows the great value of Integral (I) action in eliminating offsets. Change the SP to any value in the range and see that PV is brought to exactly the same value.

### Test No 3: Close loop step response with high integral gain

**Step 1:** Repeat all the steps of test 2 but change the integral action gain in stages upwards to see the increasing degrees of overshoot. Note the phase lag between PV and OP that cause the oscillatory action. Once the OP correction at 180 degrees phase lag is greater than the error at that point (i.e. loop gain > 1 at 180 degrees) the control system will sustain oscillation and expand its responses to the limits of PV and OP. The control is now totally unstable. See Figure D4.3 for a marginally stable response when integral action is 10.0 repeats/minute.

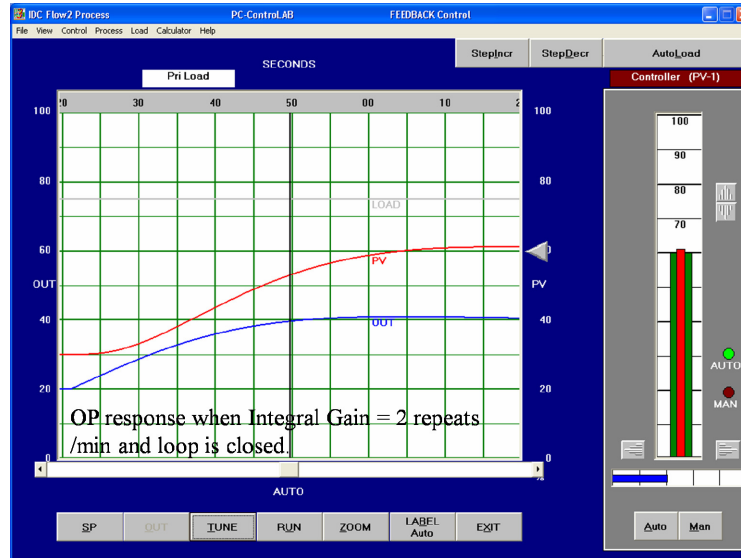
## D4.4 Example responses

Figure D4.1 shows the basic integration rate if no PV feedback occurs. The rate of change of output = Integral gain (repeats per minute) x Error (SP-PV). In this example gain = 2 and error =30%. Hence the ramp rate is 1% per second.



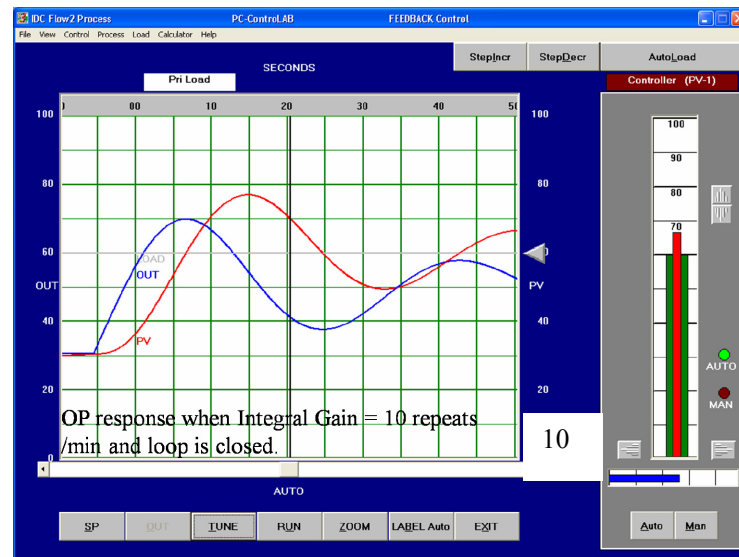
**Figure D4.1**  
*Basic integral action ramp*

Figure D4.2 shows the typical lag response when the loop is closed and the PV approaches the SP and the error is diminished to zero. Minimal overshoot occurs in this case when integral gain = 2 repeats per minute.



**Figure D4.2**  
Typical closed loop response in Integral only control

Figure D4.3 shows the response when integral action gain is too high for the process, leading to overshoots and oscillation. Note the phase shift in which the OP correction lags behind the PV.



**Figure D4.3**  
Typical closed loop response in Integral only control where gain is too high.

## D4.4 Conclusion

Integral control will control the PV towards the SP precisely, without any OFFSET. The trade-off for this, when compared with Proportional control, is a lagging behavior, which results in slower control. It will be shown later, that Integral control decreases the stability of the loop, if an intrinsic stability problem exists within the control loop.



---

## Exercise 5

---

# Proportional and Integral (PI) Control - Flow Control

### D5.1 Objective

This exercise will use the same process flow model that we used in exercise 4 but it will introduce a combined Proportional and Integral Control action of controllers. Special emphasis is given to the elimination of the remaining offset of proportional control without loss of control speed. It will be shown that the combination of Proportional and Integral control maintains the speed of control as it exists with Proportional control only, but without the disadvantage of an OFFSET term.

### D5.2 Operation

**Step 1:** Start the simulator programme and then select: **Control/Retrieve strategy and tuning/PC-Pracs/Ex-5/IDC Flow5 P+I.stg.**

Set the scroll display rate to “seconds” and set mode = MAN. Set View/Display = Percent of span. Check the Load value is 60% as observed on the trend screen.

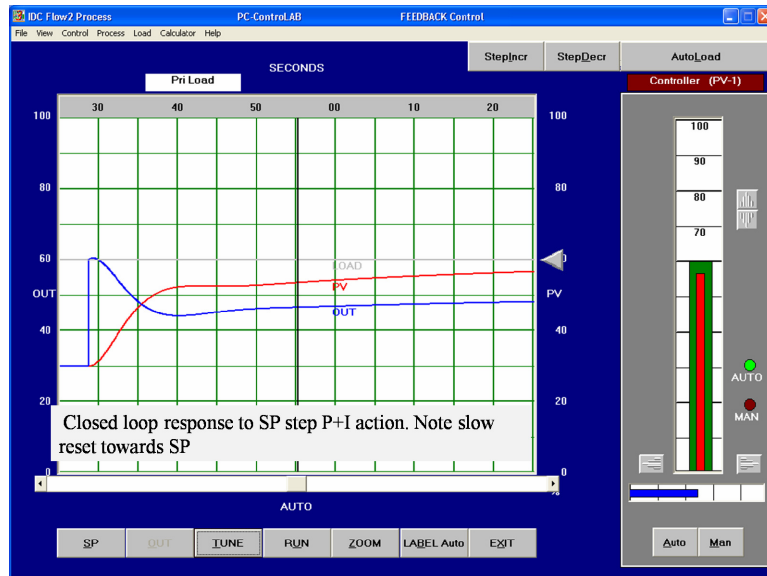
**Step 2:** Check that SP Tracking is ON by going to Control/Control Options/SP Tracking.

**Step 3:** Keep the control mode in MAN and check that the control option has been set to “PID Non-interacting”. This algorithm keeps the proportional gain and integral gain and derivative gain terms separate.

$$OP = (SP - PV) \times (K1 + Kint (\text{Integral}) + Kder (d/dt))$$

**Step 4:** Check under TUNE that: Gain = 1.0 and Reset = 2.0 repeats/min

**Step 5:** Set OP = 30% and SP = 30% and switch to AUTO. Now test the step response to a change in SP from 30% to 60%. See Figure D5.1 for an example of what you should see.



**Figure D5.1**  
*P+I step response with low gain settings*

### D5.3 Observation

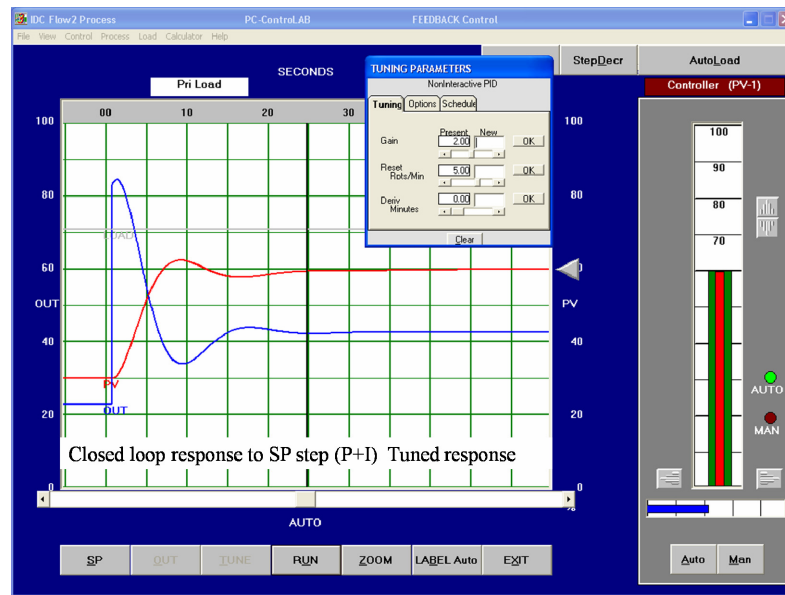
Note how the proportional action response is typically as we have seen in exercise 3 with a substantial offset between PV and SP. The offset is then gradually reduced by the integral action term driving the OP slowly upwards until the error is zero.

The slow speed of response may not be acceptable for many applications so the tuning strategy now comes into play. The proportional gain may be increased substantially to speed up response and reduce offset as we have seen in Exercise 4. However we have already seen that noise and instability problems will limit the gain.

### D5.4 Tuning Operations

The normal strategy is to increase gain to its sensible limit and then apply a small amount of integral action to rest the error to zero. You may now experiment with this approach on the simulator. For these tests it is convenient to use the STEPINCR and load decrement buttons to drive 5 % step changes into the process load.

You should start by reducing the reset action to a minimum and testing the response in proportional control only. Write down your suggested maximum gain for a response that has some degree of overshoot but seems to be reasonably stable. Then begin to apply a small amount of reset gain until the responses arrive at the set point fairly quickly but without excessive overshoots. Compare your results with the typical example in Figure D5.2.



**Figure D5.2**  
*P+I step response with tuned settings*

## D5.5 Conclusion

Combined Proportional and Integral control turns out to be the best choice of control, if no stability problem exists within the control loop. As a general rule, (with few exceptions) flow control loops have no stability problems. This is the reason why the majority of flow control loops use PI-Control.



---

## Exercise 6

---

# Introduction to Derivative (D) Control

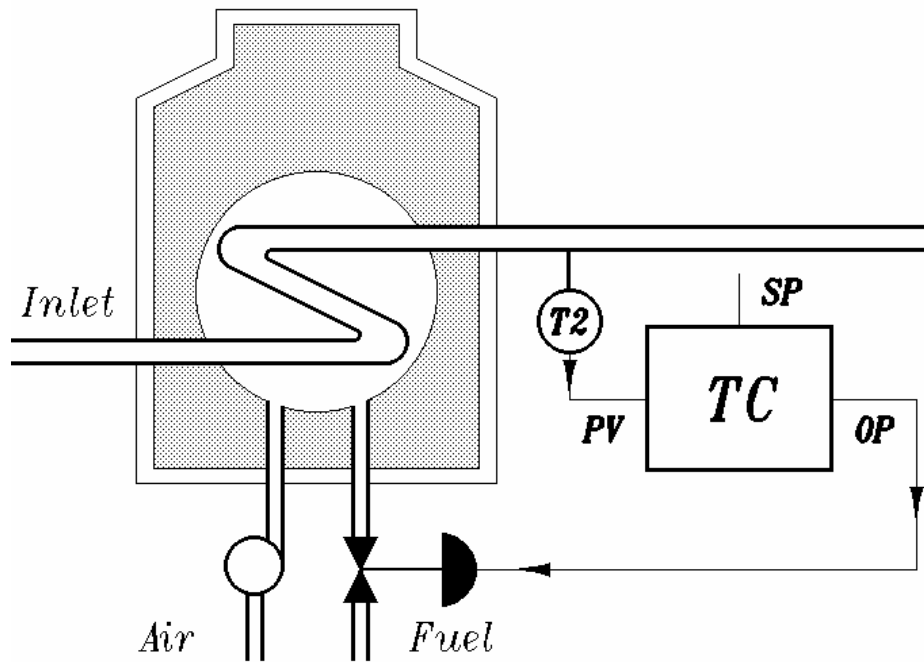
### D6.1 Objective

This exercise will introduce the Derivative action in feedback controllers. Derivative action is required only in control loops with stability problems. Derivative control counteracts the instability (oscillation) of control loops. As such it is an additional control action. Derivative Control is always combined to form a PD or PID-Control loop system.

As flow loops generally do present severe stability problems, a more typical general process model is used to demonstrate D-Control. Figure D6.1 shows a single loop to control temperature in a feed heater process. This is an example for a control loop, where a stability problem can be expected due to more complex dynamic lags.

The process model to be used is held in the file IDC Feed Htr.mdl. The model represents the dynamic lags of the gas flow, heat transfer stage and feed temperature sensing linked in series. A small dead time element is also included to represent the time for the heated material to travel to the output point of the heater where the temperature sensing is located.

An interesting feature of this process is that the gain of the open loop stage from fuel flow input to product temperature at the exit changes according to the feed flow rate. For our first exercises therefore we shall avoid non-linearity by fixing the feed rate at 60%. This type of non-linearity is commonly encountered and it makes loop tuning difficult until the characteristics are recognized. We shall conduct the next series of tests with model operating at 60% feed.



**Figure D6.1**  
Feed heater single loop control

## D6.2 Operation

Open PC-ControLAB and load the process model IDC Feed Htr.mdl. Load the control strategy file IDC Feed Htr PID.stg.

This model and the control strategy have been tuned for a tight response to the model of a feed heater as shown in Figure D6.1 but for this exercise we are going to simply demonstrate the effects of derivative control so we shall temporarily reduce the P and I tuning components.

Test 1: Will show the effect on the controller output of applying the derivative term to the error between SP and PV. This occurs if you are using the option to apply to derivative to error as in the type A equation in the manual

- **Step 1:** In MAN mode use Control options to select Deriv on Error. Select Reset action off. Select interactive PID control option. Set the scroll rate to Minutes using the View command. Set load to 0% using the StepDecr button. (In this model Load is applied only to the temperature measuring stage to simulate a bias or to inject process disturbances whilst leaving the feed steady at 60%)
- **Step 2:** TUNE and set Gain = 0.5 and Deriv = 2 minutes. Set the term derivative gain to 10.0 using the TUNE/OPTIONS tab. This introduces a substantial low pass filter in the PID equation making this response easier to observe and less sensitive to higher frequencies
- **Step 3:** Set OP = 36 % and see that PV settles at approximately 500C. If not, adjust the STEPDECRES to achieve this balance. Set the SP to equal the PV
- **Step 4:** Switch to AUTO. There should be no disturbance at this point since the error is zero.

Now we have a situation where any rapid change in error will drive the output in response to the rate of change of error. If the derivative time = 2 minutes this means for example that we can calculate the amount of output as follows:  $OP = (\% \text{ error per minute}) \times 5 \text{ per minute}$

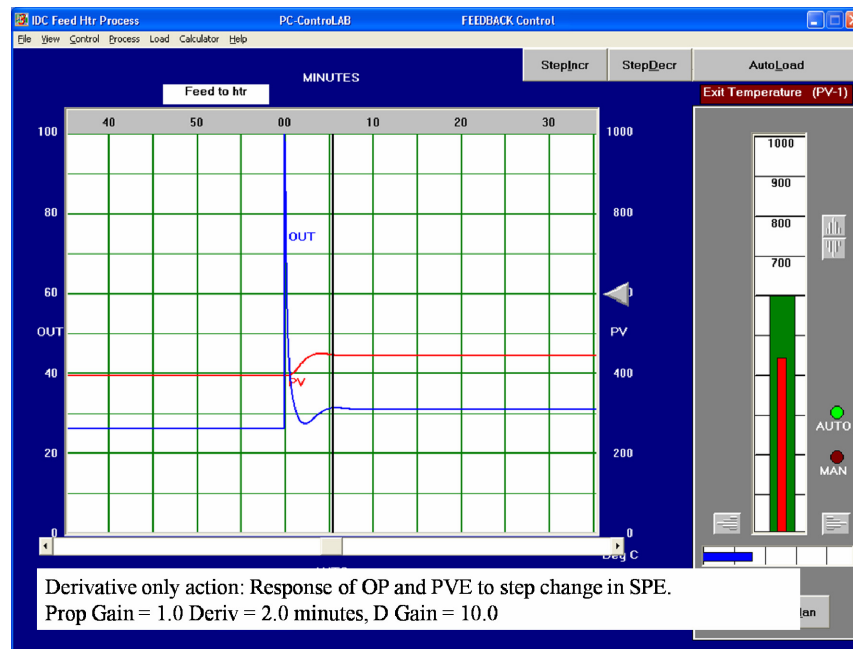
For example: 10% /minute increase in error will create  $OP = 10\% \times 2 = 20\%$ .

- Step 5: In order to observe D-Control, step change SPE from 500 degC (50% of range) to 600degC (60% of range). Figure D6.2 shows the response you should obtain.

### D6.3 Observation

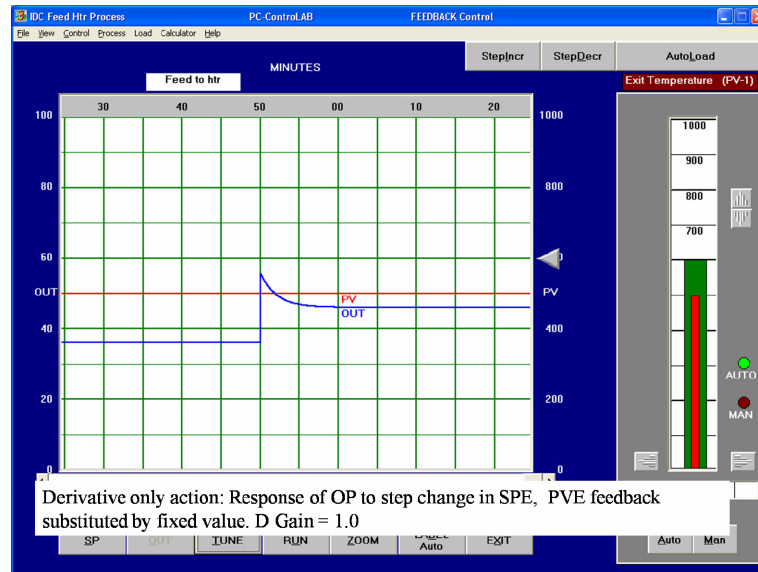
It is observed that Derivative control calculates a controller OP value based on the Rate of Change of PVE-SPE. As is explained in greater detail in the chapter "Digital Control Principles", D-Control of an "Interactive PID Controller" is in actual fact a PD-Control action with a Low-Pass-Filter. Therefore, the initial OP change as a result of the step change of SPE is not a Unit-Pulse-function (Needle shaped pulse with magnitude); it is in fact a limited step of OP, followed by an exponential decay. As a P-component always exists within this control action, PVE settles down with an OFFSET.

The time constant of the low pass filter is described as the Derivative gain and you can adjust this with the TUNE/Options control. (Normally a value of 10 is used in these simulations.) To see this more clearly you can temporarily disconnect the PVE signal and substitute a fixed value using the commands: Control/Measurement Options. Now the response of OP is purely due to the derivative action. The result of doing this can be seen in Figure D6.2.



**Figure D6.2**

Step response for SPE change 500 to 600 DegC showing derivative action on OP and resulting PVE response.



**Figure D6.3**

Step response for SPE change 500 to 600 °C showing derivative action on OP. PVE feedback has been disconnected and derivative gain set to 1.0

## D6.4 Operation

Repeat the above exercise with different values of TDER and K. Observe the clipping of OP at the OPHI and OPLO limits, if large values of K are used. You may set the OP limits to 5% (low) and 95% (high) to more clearly see the effects.

## D6.5 Conclusion

Derivative Control is based on the rate of change of PV-SP and is not designed to bring the value of PV to the value of SP. The sole purpose of D-Control is to stabilize an intrinsically unstable control loop by introducing a phase lead into the output response.



---

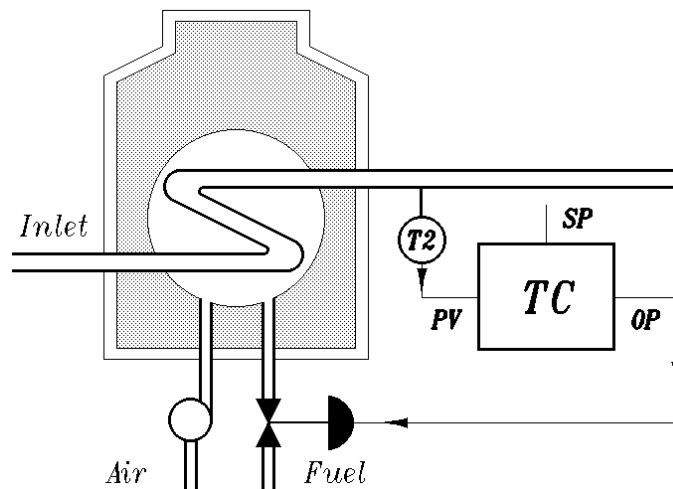
## Exercise 7

---

# Practical Introduction into Stability Aspects

### D7.1 Objective

The objective of this exercise is to demonstrate the direct relationship between process lag and stability, and to make students aware of the great difference between noise and stability in practice. We shall use the same process model that we used for Exercise 6. Please note the comments made on the characteristics of the feed heater process.



**Figure D7.1**  
*Feed heater process. Load is applied as flow through the heating tube.*

## D7.2 Operation

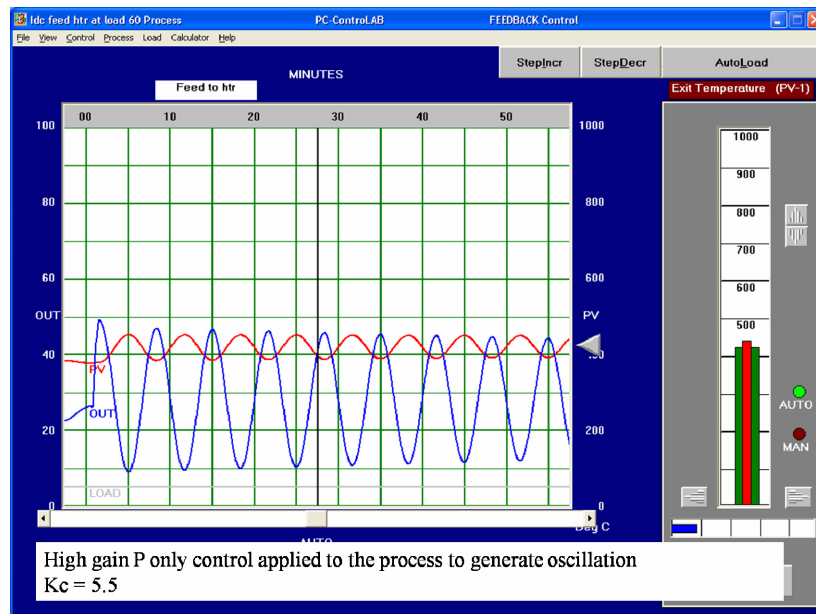
Open PC-ControLAB and load the process model IDC Feed Htr.mdl. Load the control strategy file IDC Feed Htr PID.stg. In order to see the stability problem of closed loops in practice and without disturbances ensure that the Auto Load switch is off.

- **Step 1:** Call up the TUNE display and set  $K = 5.5$ , and  $Deriv = 0$ . Go to TUNE/OPTIONS tab to turn off the integral action (i.e. Reset Action Off).
- **Step 2:** Set the control to MAN and set  $OP = 36\%$ . Set Load to 0 % and note that PVE stabilizes at approximately 500 degC.
- **Step 3:** Move the SPE to 500degC (or keep SP tracking ON).

The value of proportional gain you have set makes it possible to create a continuous oscillation, purely based on the lag of the process.

- **Step 4:** Switch to AUTO and then operate the StepIncr switch once to insert a 5% step load change on the process to create the disturbance.

See Figure D7.2 for the expected response.



**Figure D7.2**  
Oscillation with P only control, showing 180 degree phase shift

## D7.3 Observation

Observe that the process is driven into a state of continuous oscillation with the OP signal being 180 degrees out of phase from the PVE. With these tuning values, no integral and no derivative control action takes place and the controller will neither increase nor decrease the phase shift within the control loop. Therefore, only the process has caused the existing phase shift of 180°.

In this situation, the controller performs P-Control only. As shown in the chapter "Stability and Control Modes of Closed Loops", continuous oscillation exists if Loop Gain is 1 and Loop Phase Shift is 180°. Since our controller Gain  $K$  is 5.5 we can calculate the process Gain at the resonant frequency.  $K_p = 1/5.5 = 0.182$ . (Gain is attenuated through the process lags as frequency rises).

## D7.4 Operation

In the following operation, we will add derivative control to the existing situation of continuous oscillation, in order to find out what effect it will have on stability. Change Deriv in the TUNE Display to 0.5 minutes. This should result in a Trend Display as shown in Figure D7.3.

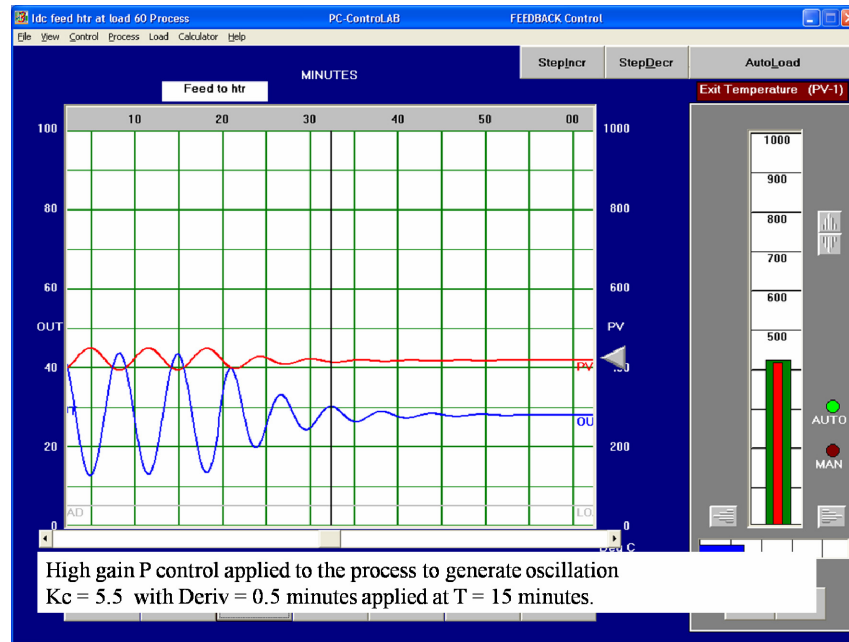


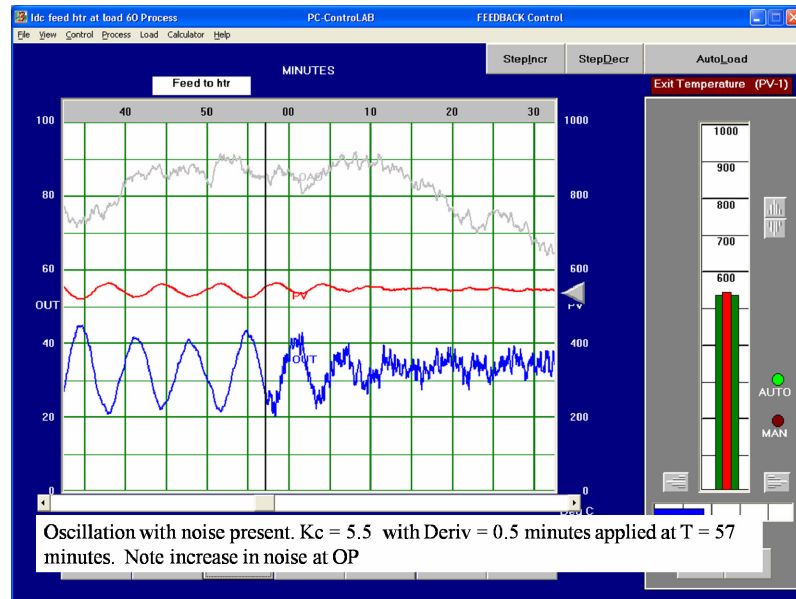
Figure D7.3  
Transition from P-control to PD-control

## D7.5 Observation

The left side of the Trend Display (Figure D7.3) shows continuous oscillation with P-Control only. The right side of the Trend Display shows the effect of added D-Control. The trend shows obviously faster control combined with a suppression of the oscillation. This means that **D-Control has a stabilizing effect.**

## D7.6 Operation

It is vital for every Process Control Engineer to have experience in the effects of Process Noise and D-Control. (The value of K-NOISE is different with different processes.) Now repeat the same operation as before ( $K = 5.5$ , TDER + 0, Reset Action OFF), in order to create continuous oscillations and then press AutoLoad (but now superimposed by noise). Then add D-Control (TDER + 0.5). Figure D7.4 shows the Trend Display resulting from this exercise.



**Figure D7.4**  
Transition from P-control to PD-control of a noisy process

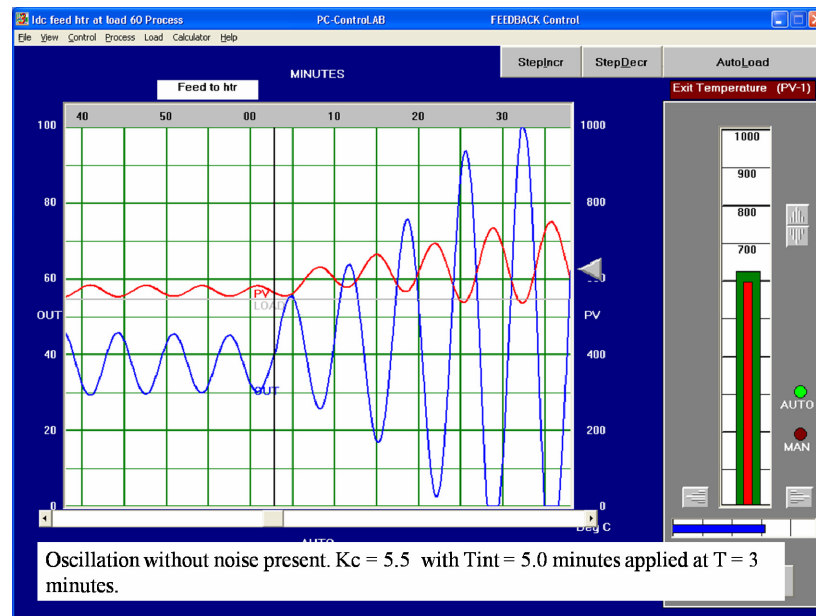
## D7.7 Observation

It should be clear that D-Control still has some stabilizing effect but it also multiplies noise as can be seen in the OP trace.

## D7.8 Operation

Now repeat the exercise as above and turn derivative term back to 0.0, but change to PI-Control, (not PD-Control as before), in a continuous oscillating situation created by P-Control.

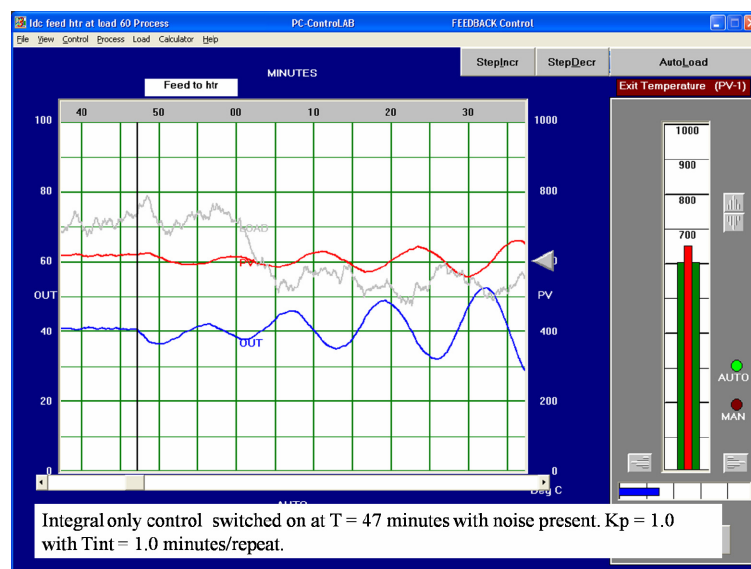
- Step 1:** In MAN: First set the load to approx 50%. Set OP =36 and see that PV settles at about 550°C.
- Step 2:** Keep reset action switched off and then set to AUTO and induce the oscillation.
- Step 3:** Switch on AutoLoad to induce noise
- Step 4:** Set TINT = 5.0 minutes /repeat (reset = 0.2 repeats/ minute).
- Step 5:** Use TUNE/Options to switch on reset action and then observe the response, which should look like Figure D7.5.



**Figure D7.5**  
Transition from P-control to PI-control of a noisy process

Then start again with continuous oscillation, created by P-Control only and change to mainly I-Control only by taking the following steps:

- Step 1:** Start the simulation in MAN with reset switched off and change K from 5.5 to 1.0 and set TINT 1.0.
- Step 2:** Switch to AUTO and switch on the Auto Load. You will see that the control is stable despite the noise because the P term is on low gain. The PV may not be very close to SP because there is no reset action.
- Step 3:** Switch on the reset action and observe how the instability quickly builds up as shown in Figure D7.6.



**Figure D7.6**  
Transition from P-control to I-only-control of a noisy process

## D7.9 Observation

Figures D7.5 and D7.6 show that integral action has a destabilizing effect, whether it is combined PI-Control or I-Control only. In both cases we observe an increase of magnitude in oscillation. In addition we observe, that I-Control ignores noise. We can see in Figure D7.6 that the change from P-Control to PI-Control has no effect on noise. Noise is not different in P-Control and PI-Control. Noise is almost totally suppressed when I-Control is used only.

## D7.10 Conclusion

This exercise brings us to the following conclusions:

D-Control has a stabilizing effect, whereas I-Control has a destabilizing effect. The only purpose of I-Control is to eliminate the offset of P-only Control.

The trade-off is the destabilizing effect. This may be compensated by adding D-Control. This will result in the use of PID-Control for control loops with stability problems.

D-Control has to be treated very carefully from a noise point of view. It is important to point out that D-Control should never be used without prior filtering of the PVE. Such a filter has to be chosen to reduce noise without adding a significant Lag to the loop. You may access the filter time constant under CONTROL/Measurement Options.

*Note: Never use D-Control without PV filtering (TD) if any process noise exists.*

---

# Exercise 8

---

## Further Exercises in Identification of Process Characteristics

### D8.1 Introduction and Objectives

The designers of the PC ControlLab simulation facility have kindly made available to IDC Technologies a comprehensive suite of training exercises designed to help the student to develop a good understanding of many of the basic principles of process control.

At the stage of the IDC workshop when you have completed Chapter 2 we would suggest that you work through the following exercise provided by Wade Associates which is designed to build up your knowledge of typical process characteristic responses. This series of exercises covers the recognition of the following features.

- First Order Lag plus Dead time process. ( As already seen in Exercise 1)
- Negative process gain
- Integrating process
- A simple frequency response test

Try the exercises before you look at the answers that are available at the end of this Appendix.

## LABORATORY EXERCISE 1A PROCESS DYNAMIC CHARACTERISTICS

**OBJECTIVE:** To become familiar with various forms of process dynamic characteristics, and to learn a method of constructing a simple process model from step test data. Optional: To become familiar with obtaining data from frequency response tests.

**PREREQUISITE:** Completion of PC-ControLAB tutorial (under Help | Tutorial ) or an equivalent amount of familiarity with the program operation.

**BACKGROUND:** All process have both steady state and dynamic characteristics. From a process control standpoint, the most important characteristic is the process gain. That is, how much does the process variable (PV) change for a change in controller output. If both the PV and the controller output are expressed as normalized variables (i.e., 0 - 100%), then the process gain is a dimensionless number.

The two most important dynamic characteristics of a process are the amount of dead time in the process and its time constant. Real processes rarely exhibit a response of a pure first order lag (time constant) and dead time, but can often be approximated as a first order lag and dead time.

This exercise tests for the process gain, dead time and time constant for both a “pure” process (can be exactly represented as first order lag plus dead time) and for a more realistic process which can only be approximated as a first order lag plus dead time.

### 1. RUNNING THE PROGRAM

Start **Windows**.

Run **PC-ControLAB**.

### 2. FIRST ORDER LAG PLUS DEAD TIME PROCESS

Click on **Process | Select Model**.

Highlight “Folpdt.mdl” (First Order Lag Plus Dead Time) and press **Open**.

Press **Zoom** and change the PV scale range to 50-75. (Note that the PV scale has already been converted to 0 – 100% of measurement span.)

With the controller in **MANUAL**, press **Out**. Note the initial values:

Present PV: \_\_\_\_\_

Present Controller Output: \_\_\_\_\_ Key in a new output value of 45.0.

*(Hint: After the numerical value is keyed in, wait until a vertical green line on the grid is just crossing the grid boundary before pressing ENTER. This will make it easier to estimate subsequent times.)*

After the PV has stabilized at a new value, press **PAUSE**.



Final value of PV: \_\_\_\_\_

What type of process response does this appear to be? \_\_\_\_\_

How much did the PV change? \_\_\_\_\_

How much did you change the controller output: \_\_\_\_\_

Process Gain:  $K_p = \frac{\Delta PV}{\Delta Output}$  \_\_\_\_\_

How long after the controller output change before the PV started changing?

Dead Time ( $T_d$ ) \_\_\_\_\_

Calculate 63.2% of PV change: \_\_\_\_\_

Actual value of PV at 63.2% of change: \_\_\_\_\_

How long after the PV started changing (i.e., at the end of the dead time) before the PV crossed the 63.2% point?

Time Constant ( $\tau$ ) \_\_\_\_\_

Suppose we had used the 2/3 point, rather than 63.2%, to estimate the time constant.

2/3 of PV change: \_\_\_\_\_

Actual value of PV at 2/3 of change: \_\_\_\_\_

How long after the end of the dead time before the PV crossed the 2/3 point?

Time Constant (approx) ( $\tau_{est}$ ) \_\_\_\_\_

What would be the percent error if we used the 2/3 point to estimate the time constant, rather than the true 63.2%?

Percent error:  $\frac{\tau_{est} - \tau}{\tau} \times 100$  \_\_\_\_\_

Select **Process | Change Parameters**. Observe the values listed for Dead Time and Time Constant. Do these parameter values agree with what you observed?

\_\_\_\_\_

## Changing process parameters

Select **Process | Initialize**.

Select **Process | Change Parameters**.

Select “Dead Time” and change its value to 2.0 (minutes).

Select “Process Gain” and change its value to 1.0.

Select “Time Constant” and change its value to 3.0 (minutes).

Press **CLEAR**.

Change the controller output from 35.0 to 45.0. Observe the response. Is this what you would expect? \_\_\_\_\_

*You have just observed the response of a pure first order lag and dead time process. Very few, if any, processes are this “clean.” We will now look at a process with unknown dynamics, but we will attempt to approximate it with a first order lag plus dead time model.*

### 3. UNKNOWN PROCESS

Click on **Process | Select Model**. Highlight “Generic” and press **Open**.

Notice that the PV scale is now in Engineering Units, rather than in percent. (If not, then select **View | Display Range | Engineering Units**)

Upper end of scale (corresponds to high end of transmitter range) \_\_\_\_\_

Low end of scale (corresponds to low end of transmitter range) \_\_\_\_\_

Span of PV \_\_\_\_\_

Initial value of PV (in engineering units) \_\_\_\_\_

Initial value of PV (in percent of span) \_\_\_\_\_

Controller Output: \_\_\_\_\_

Change the controller output from 35.0 to 45.0.

When the PV reaches (apparent) equilibrium, press **Pause**. Does this look like a true first order lag plus dead time process? \_\_\_\_\_

Does it look “approximately” like a first order lag plus dead time process? \_\_\_\_\_

What is the final value of the PV (to the nearest whole number)? \_\_\_\_\_

How much did the PV change, in engineering units? \_\_\_\_\_

How much did the PV change, in percent of span? \_\_\_\_\_

Estimate the process gain.  $K_p$  \_\_\_\_\_

To estimate the dead time, draw (or visualize) a tangent to the PV curve, drawn at the point of steepest rise. From the time of controller output change to the intersection of this tangent with the initial steady state value is the apparent dead time.

Apparent dead time:  $T_d$  \_\_\_\_\_

Different observers might estimate anywhere between 1½ to 2 minutes. For the purpose of calculating controller tuning parameters, it is better to take the longer value where there is any uncertainty, since that will produce more conservative controller tuning.

The apparent time constant is the time from the end of the dead time to 63.2% of the process rise (or approximately the time from the end of the dead time to 2/3 of the process rise.)

Apparent time constant:  $\tau$  \_\_\_\_\_

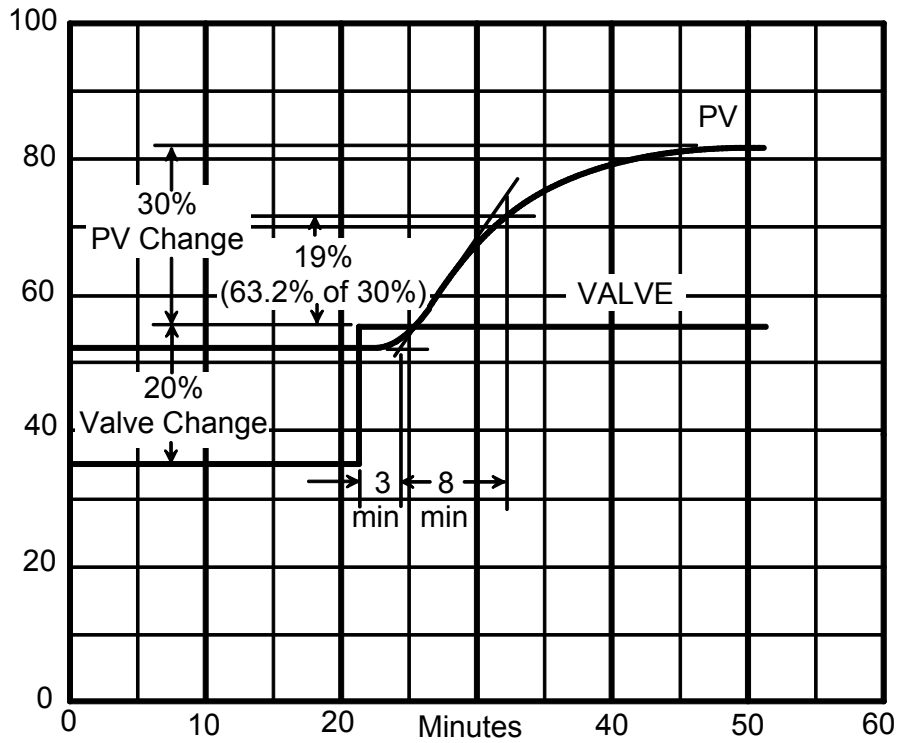
*Rather than estimating the time to 2/3 rise point, you can calculate the value of the PV at 63.2% of the ultimate PV change, then use the scroll bar to determine the time precisely between the (estimated) end of the dead time and the 63.2% rise point. This gives a more accurate estimate of the time constant, but takes longer.*

NOTE: One of the uses that can be made of the estimates of process gain, dead time and time constant is to calculate controller tuning parameters. (See Laboratory Exercise 9, PID Tuning from Open Loop Tests.) Since dead time is more difficult to control than a first order lag, then if you estimate dead time too short, you are estimating that the process is easier to control than it really is. This will result in controller tuning parameters that cause the loop to be overly aggressive.

Similarly, if you estimate the time constant too long, you are estimating that the process is easier to control than it really is, and again the resulting controller tuning parameters will cause the loop to be overly aggressive. On the premise that is one is to make an error, it is better to err in the conservative direction than in the aggressive direction, then the following pragmatic guidelines can be given:

*If there is any uncertainty in estimating the process parameters,*

*estimate the dead time on the long side, and  
estimate the time constant on the short side.*



**Figure 1**  
**EXAMPLE OF ESTIMATING PROCESS MODEL PARAMETERS**

**4.0 OTHER FORMS OF STEP RESPONSE**

This section will explore other forms of step response.

**4.1 Negative Process Gain**

Select **Process** | **Select Model**. Highlight “generic3.mdl” and press **Open**.

Record the initial values:

PV: \_\_\_\_\_ **DegF** \_\_\_\_\_ %

Controller output: \_\_\_\_\_ %

Increase the controller output by 10%. When the PV reaches equilibrium, record the following:

Final value of PV: \_\_\_\_\_ **DegF** \_\_\_\_\_ %

Final value of controller output: \_\_\_\_\_ %

Process Gain:

$$K_P = \frac{\Delta PV(\%)}{\Delta \text{Controller Output}(\%)}$$

$$= \frac{PV_{\text{final}}(\%) - PV_{\text{initial}}(\%)}{\text{Cont Output}_{\text{final}}(\%) - \text{Cont Output}_{\text{initial}}(\%)}$$

## 4.2 Integrating Process

*An **integrating process** is one which will not achieve a natural equilibrium. The control of liquid level, where either the vessel inflow or vessel outflow is fixed by an external set point, is a common example of an integrating process.*

Select **Process | Select Model**. Highlight “level2.mdl” and press **Open**.

*This process model simulates a level control application in which the tank outflow is an independent process disturbance (load), and the controller controls a valve in the inflow line.*

*This process model has some elements of "realism" that are not needed in this exercise; These features will be used in subsequent laboratory exercises. For the present, we will eliminate these "realism" elements.*

Select **Process | Change Parameters**.

Highlight the parameter labeled "Valve Pos: 0 = No; 1 =Yes." Change this parameter value to I.0.

Select the parameter labeled “Levl Noise: 0=No; I=Yes”. (Use the scroll bar at the left hand side of the parameter list, if this parameter is not visible initially.) Change this parameter value to 0.0.

Press **Clear**.

Change the controller output from 35.0 to 38.0. This simulates an increase in inflow to the tank.

What does the PV (tank level) do? \_\_\_\_\_

Does it appear that the PV is going to reach an equilibrium? \_\_\_\_\_

Before the PV reaches a limit (0 or I 00), press the button **StepIncr** above the controller faceplate. This simulates an increase in outflow from the tank. Note the increase in the trace labeled 'LOAD'.

*If the PV is now falling, then the rate of outflow must exceed the inflow. Slightly increase the controller output (to increase the tank inflow) to attempt to stabilize the level. If the PV is rising, decrease the controller output. Can you eventually (approximately) stabilize the level by manual control? \_\_\_\_\_*

## 4.3 Inverse Process Response

*Occasionally the process response to a step change in controller output, or to a load change, is initially in the opposite direction to that expected from "first principles." This is normally due to some underlying, second order effect. Once the second order effect disappears, then the process responds in the expected manner. A response such as this is called an "inverse response."*

*An example of inverse response is the “shrink-and-swell” effect of steam boiler drum level. If the steam draw-off and steam rates are in equilibrium, then the drum level will be constant. On an increase in steam draw without a corresponding increase in feedwater rate, a decrease in drum level would be expected. Initially, however, due to the reduction in pressure and consequent flashing of water into steam, the level rises; if*

*the excess steam draw is maintained, the level eventually begins to drop. This is qualitatively demonstrated in this exercise.*

Select **Process | Select Model** and recall the Level2 process.

Select **Process | Change Parameters**.

Select the parameter labeled "Levl Noise: 0=No: 1=Yes". Change this parameter value to 0.0.

Select the parameter labeled 'BLK 57 LEAD TIME". Change the value of this parameter from 1.5 to -3.0.

Press **Clear**.

Press **StepIncr**. This simulates an increase in steam flow from the boiler drum.

Observe the PV (drum level). What does it initially do? \_\_\_\_\_

*(This is the drum level "swell" effect. If the steam flow had have been decreased, rather than increased, we would have seen the opposite effect, or the "shrink".)*

What does the level eventually do? \_\_\_\_\_

Suppose a feedback level controller had been in Automatic and controlling the feedwater rate (drum input). Upon sensing the initial change In level, would the level controller Increase or decrease the feedwater rate?

Would this change be in the proper direction for long-term correction?

*Laboratory Exercise 23, (See IDC Practical No 15), Drum Level Control, will demonstrate a control technique widely used in steam generation applications for overcoming this problem.*

## 5.0 FREQUENCY RESPONSE (Optional)

*One means of characterizing a process is by its response to a sinusoidal (sine wave) input at various frequencies. At each frequency, the relevant data is:*

*the **ratio of the amplitudes** of the input and output sine waves;  
the **phase shift** between input and output signals.*

*After a number of data points are taken, a **Bode plot** of the data can be constructed.*

*This portion of this laboratory exercise will determine a few data points for a Bode plot.*

Through the **Process | Select Model** menu, read in "Generic1" process model. Be sure you are using the FEEDBACK control strategy (read the right hand side of the display title bar).

Check to see that the controller is in Manual.

Select **Control | Control Options**.

Scroll down until you see "Enable Sinusoidal Output." Click on "YES."

Set the following parameter values:

Amplitude, Peak-to-Peak: 10  
 Period, minutes: 60

Press **Clear** then press **On** on the controller to initiate sinusoidal testing of the process.

Record the following:

Period, controller output and measurement: 60 minutes

Frequency, controller output and measurement, 0.0167 cycles/min

Controller Output Amplitude, Peak-to-Peak: 10%

Process Variable Amplitude, Peak-to-Peak (in % of full scale) \_\_\_\_\_

Amplitude Ratio, PV/Controller Output: \_\_\_\_\_

Time lag between controller output and PV (minutes): \_\_\_\_\_

Phase shift, degrees:

$$\text{Phase Shift} = \frac{\text{Time lag, minutes}}{\text{Period, minutes}} \times 360^\circ$$

\_\_\_\_\_

Repeat this test for periods of 30, 15, 7.5 and 3.75 minutes. (You may have to increase the amplitude of the input signal at the shorter periods. This is alright, since it is only the ratio of the output and input amplitudes that you are seeking.)

Period (Minutes)	Frequency (cyc/min)	Cont Output Amplitude	PV Amplitude	Amplitude Ratio	Time Lag	Phase Shift Degrees
60	0.0167	10.0				
30	0.0333	10.0				
15	0.0667	20.0				
7.5	0.1333	20.0				
3.75	0.2667	20.0				

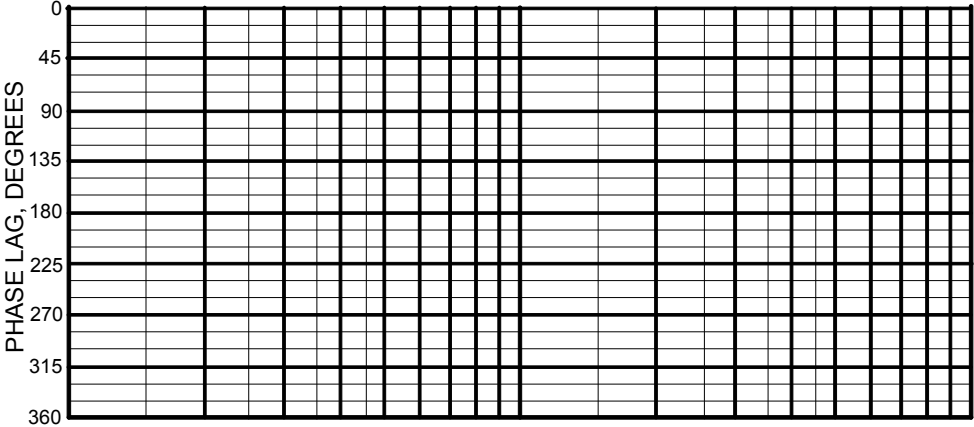
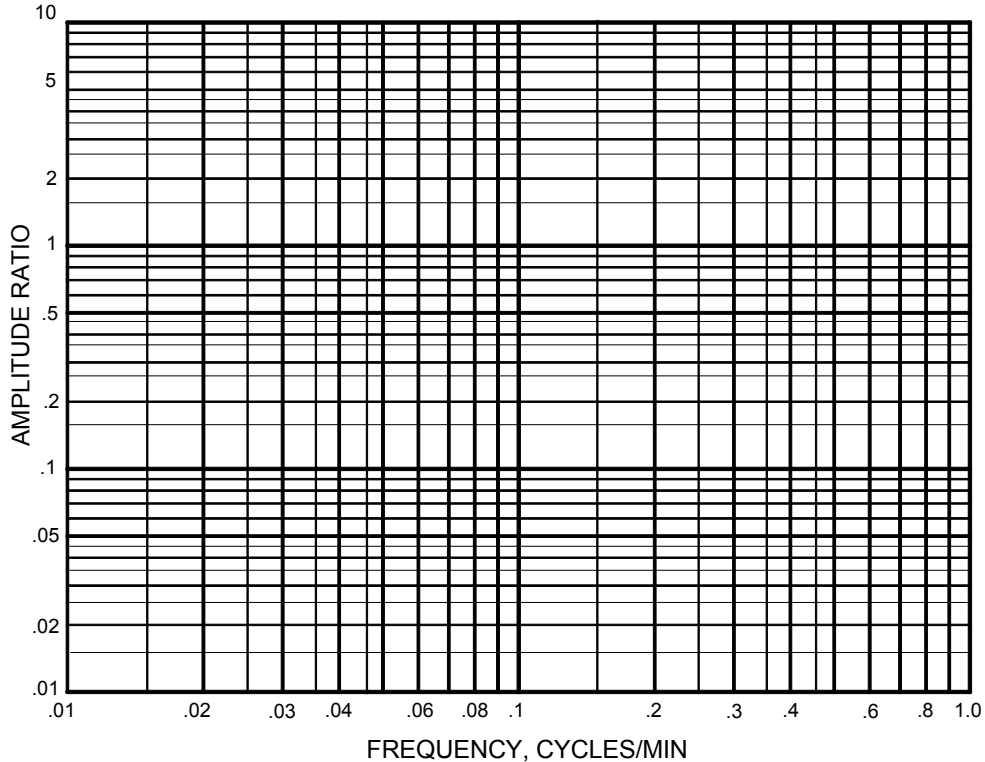
Plot these values on the graphs on the following page.

(Note: On many Bode plots, **decibels**, rather than amplitude ratio, is plotted on a linear scale. The conversion equation from amplitude ratio to decibels is:

$$\text{Decibels} = 20 \text{Log}_{10}(\text{Amplitude Ratio})$$

Also, frequency is usually plotted on a logarithmic scale reading in **radians per time unit**, rather than cycles per time unit. A complete Bode plot is beyond the scope of this laboratory exercise, however.)

For answers to this exercise please turn to the end of this Appendix.





---

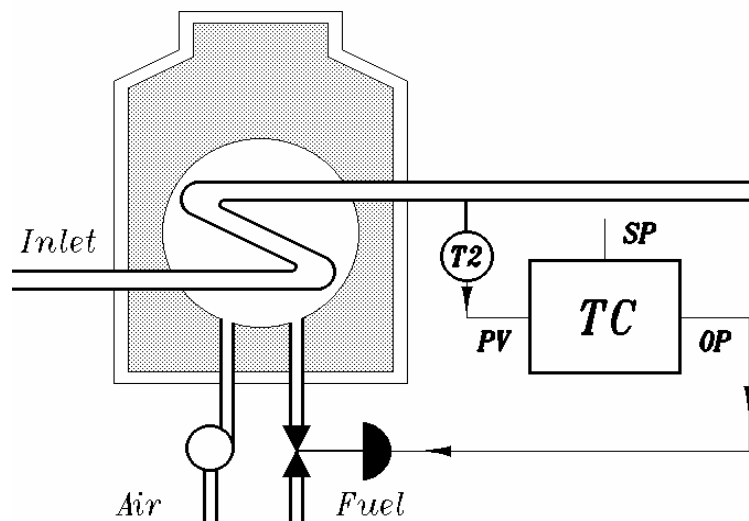
## Exercise 9

---

# Open Loop Method - Tuning Exercise

### D9.1 Objective

This exercise will give some practical experience in the Reaction Curve Method of tuning. Figure D9.1 shows the diagram of a typical control loop which is the feed heater we have used in the previous exercise. The steps for tuning are explained in Chapter 5.



**Figure D9.1**  
*Single loop control*

## D9.2 Operation

Open PC-ControlLAB and load the process model IDC Feed Htr EX-9.mdl. Load the control strategy file IDC Feed Htr Ex-9.stg.

It is practicable that the tests is done with a realistic level of noise in the process and that the measurement noise filter should be included in the tuning process. Therefore you should set up the Load signal on the simulator and we suggest the following settings be made under LOAD/Auto Load Change Random:

Max size: 0.3

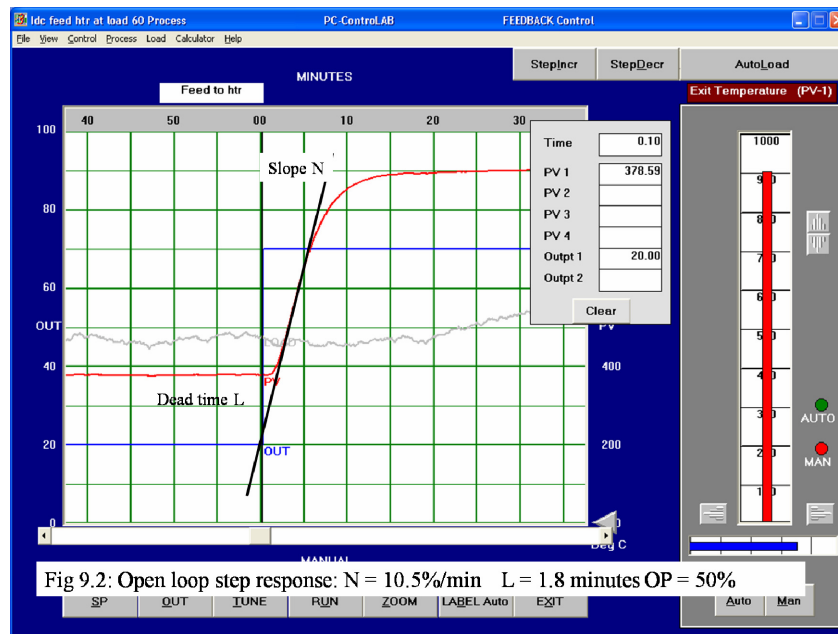
Correlation: 0.1

This produces a noise signal of typically 3% PV with a random walk effect. It is not possible to eliminate the random walk effect in this simulation.

Make a good judgment about the noise observed on the PV and set the digital noise filter (TD) accordingly using the command CONTROL/Measurement Options. A good value of TD will result in a still noticeable but very small magnitude of noise on PV. The noise filter has to be set up before tuning in order to have it included in the result of tuning. From a practical point of view, the noise filter becomes part of the process as far as loop tuning is concerned. In this exercise, a good value for TD will be between 0.05 and 0.1.

The open loop tuning method requires that we first generate a good step response to an OP step. In this case choose the practical range of PVE that you can achieve by starting with OP at about 20% and stepping it to about 70%. Use the StepIncr/Decr controls to bias the load signal into a suitable region for this test.

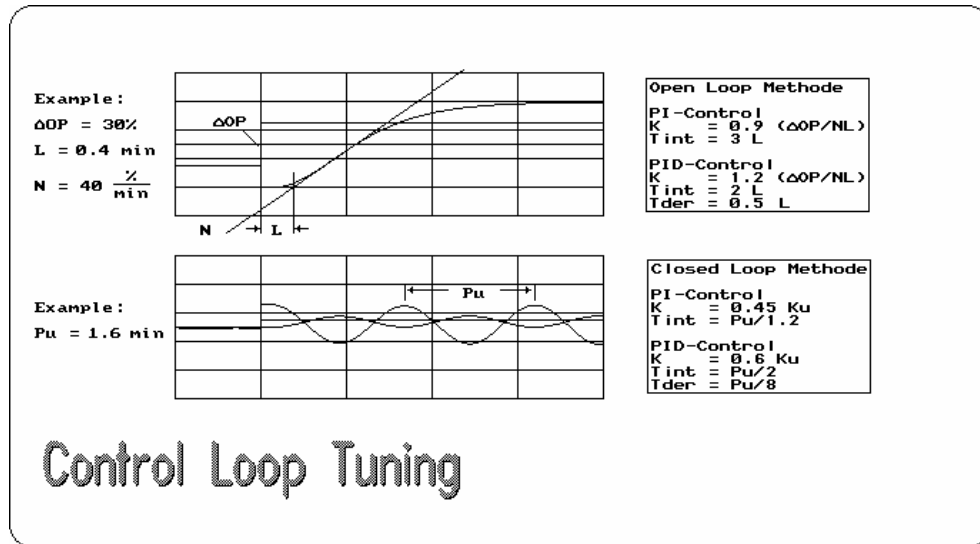
To obtain a Process Reaction Curve as shown in Figure D9.2, set MODE to MANUAL, then OP to 20 and wait until the PVE is steady at approximately 400 °C . Then make a change of the OP from 20% to 70%.



**Figure D9.2**  
Reaction curve trend display for open loop tuning

### D9.3 Observation

Observe the reaction curve and tune the controller as explained in Chapter 8. The summary of the major tuning formulas is shown in Figure D9.3.



**Figure D9.3**  
 Tuning formula display

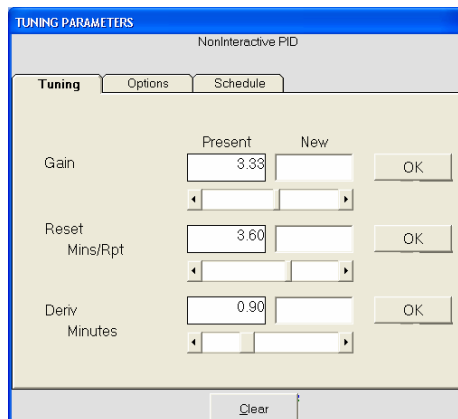
The values obtained from the reaction curve are:

- dOP = 50% (from 20% to 70%)
- N = 10.5 % / minute (PVE range from 0 to 1000C)
- L = 1.8 minutes

Using the formulas as described in Chapter 8 paragraph 8.4.3 we obtain the following tuning constants for PID-Control:

- K = 3.3
- TINT = 3.6 minutes/repeat
- TDER = 0.9 minutes

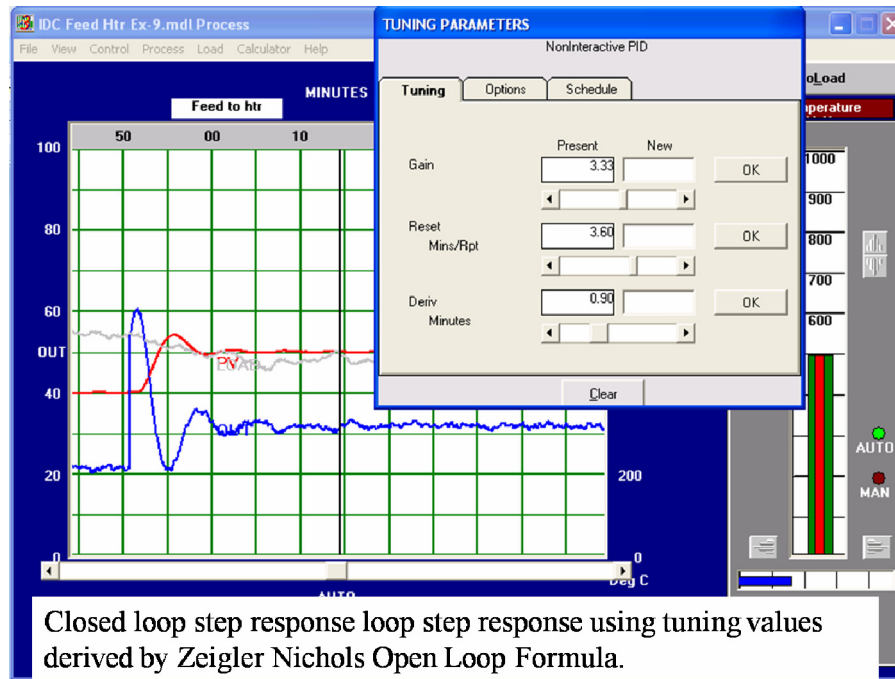
After setting the tuning constants to their calculated values, the tuning panel on the simulator should look as shown in Figure D9.4.



**Figure D9.4**  
 Detail display with tuning constants

## D9.4 Confirmation of proper tuning

Change Mode to Automatic and make a change of the SP. Observe the process settling down (Observe PV). Use the trend display as shown in Figure D9.5 for observation. Most parameters can be operated from the trend display as well. As a rule of thumb, the settling down should take place with quarter damping (1/4 decay). That means two successive maximum values of a damped oscillation should have a magnitude ratio of 1 to 4.



**Figure D9.5**  
*Step response of preliminary tuned loop based on ZN formula*

## D9.5 Fine tuning

Based on process knowledge (Noise on PV etc), experience and precise knowledge of the control actions of each control mode (P-Control, I-Control and D-Control) fine tuning of the process should be performed now. Only minor variations to the tuning constants should be made.

### TD-fine tuning

Observe the noise of the control action (OP). Use your judgment as to how much to increase or decrease TD. The judgment has to be based mainly on the effect noise in the OP signal has on the manipulated variable (Wear and tear of a valve for instance).

An increase of TD may require an increase of T2 (Derivative Time Constant) to compensate for the lagging phase shift of the noise filter.

### K-fine tuning

Since K is the Gain for all control modes (PID), reducing gain reduces the effect of all control actions equally and vice versa. A reduction of K reduces speed of control and increases stability and vice versa.

### T2-fine tuning

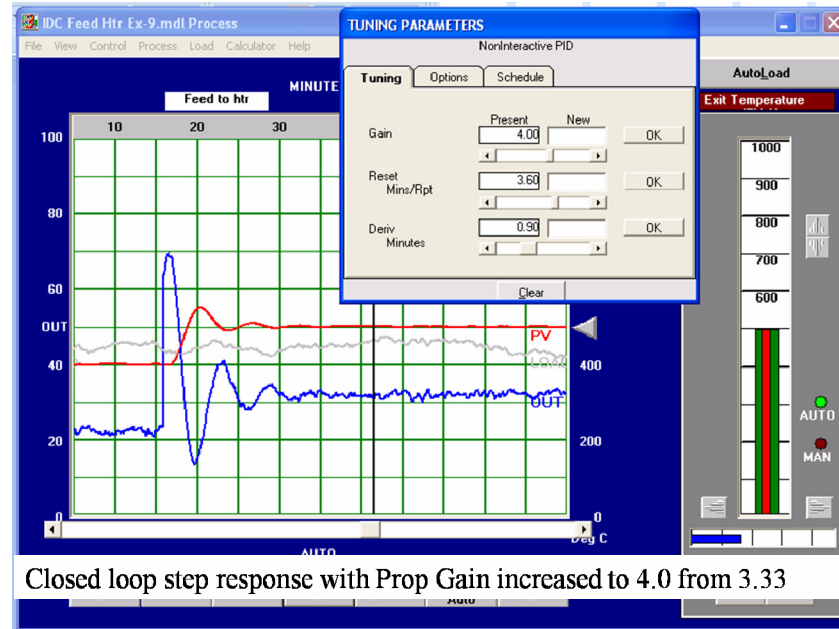
After tuning, (using the Ziegler Nichols formulas), the damped oscillation of the process after a change of SP takes place equally around the new SP. Increasing the Derivative Time Constant (T2)

will bend the baseline towards the old value of SP and therefore reduce the overshoot. In addition stability in general will be increased. It has to be noted, that the area of error increases as well with increased stability. The reason for this is the slower approach of PV towards the SP.

### T1-fine tuning

In most cases Integral tuning should be corrected only in relatively stable loops like flow loops. T1-Fine Tuning should be done with close observation using the trend display. Decreasing T1 increases instability and vice versa.

The example given in Figure D9.6 shows the quarter decay response obtained by increasing proportional gain above the value calculated from the Zeigler Nichols formula.



**Figure D9.6**  
*Fine Tuned version with increased proportional gain*

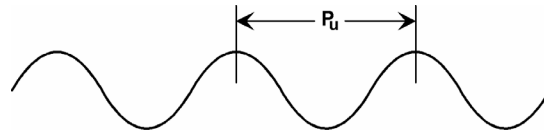


---

# Exercise 10

---

## Closed Loop Method - Tuning Exercise



### D10.1 Objective

This exercise will give some practical experience in the Closed Loop Tuning Method. It will show you how to recognize the characteristic responses shown by a close loop control system and how to use these features to guide you to find suitable PID tuning parameters.

We are going to use the exercise kindly provided by Wade Associates which involves an interactive step by step exercise in which you will record your answers on the question sheets.

**Background:** Tuning by closed loop process testing involves putting the controller in AUTOMATIC, removing all Reset and Derivative, and setting the Gain just high enough to cause a sustained process oscillation. From this test, the relevant parameters are the period of oscillation in minutes, and the Gain which ultimately caused the sustained oscillation. These are called the “ultimate Period” and “ultimate Gain”, respectively. From this data, the tuning parameters can be calculated.

**Note:** The open-loop (the subject of IDC Exercise 7) and the closed-loop methods (the closed-loop method is the subject of this exercise) may or may not produce similar tuning values, even when using the same process model. The “Generic” process model is used in these exercises; there is some difference in the results. For demonstration purposed, is you wish to obtain similar results from t he two methods, use the Generic2” model.

## 1. RUNNING THE PROGRAM

Start **WINDOWS**

Run **PC-ControLAB**

## 2. TUNING BY CLOSED LOOP PROCESS TESTS .

### 2.1 Setup

Confirm the following:

Process: **GENERIC** (see the top line, left hand side)

Select: Control Options/ Control Strategy: **FEEDBACK** (see the top line, right hand side)

Start PC-ControLAB

Select **Control | Select Strategy | Feedback**

Select **Process | Select Model/ PC Pracs/EX-10/Generic.mdl**

Select **Process | Initialize** to initialize the process model.

If you are more familiar with using Proportional Band, rather than Gain, for tuning controllers, or if you are more familiar with tuning the reset (integral) mode in Repeats per Minute, rather than Minutes per Repeat, then:

Press **Tune** then select the **Options** tab to set up the program to match the system you use:

Display GAIN as:

GAIN

PROP BAND

Display RESET as:

Mins/Repeat

Repeats/Min

Press **Tune** then select the **Options** tab. Select Reset Action OFF.

### 2.2 Process Testing

Set: Gain: 1.0 or PB:100%

Reset:

OFF (see above)

Deriv:

0.0 minutes

Put the controller in Auto.

Make a set point change of 5% of full scale. (Press **StepIncr** once.)

If there is no oscillation, or if the oscillation dies out, increase the Gain (or decrease the Proportional Band) and repeat the set point change. (The Gain can initially be changed approximately 50% of its present value, or the PB can be changed to one-half of its present value. As the response gets closer to sustained oscillation, smaller changes should be made.) You should not have to observe the response for more than three cycles to determine whether or not the oscillation is decaying or not.)



When sustained oscillation is ultimately achieved, record the following:

Gain or PB required for sustained oscillation       $K_{cu}$  = \_\_\_\_\_  
 or       $PB_u$  = \_\_\_\_\_  
 Period of sustained oscillation       $P_u$  = \_\_\_\_\_

Use the table for the closed-loop Ziegler-Nichols method (Table 1 at the back of this exercise) to calculate tuning parameters for a P, PI and PID controller. Enter these in the table below:

*(First calculate Gain ( $K_C$ ), Integral time ( $T_I$ ) and Derivative ( $T_D$ ) from the equations. Then, if your system uses PB rather than Gain, or Reset Rate rather than Reset Time, calculate those values.)*

	<b>P</b>	<b>PI</b>	<b>PID</b>
<b>Gain (<math>K_C</math>)</b>	<b>4.5</b>	<b>4.0</b>	<b>5.4</b>
<b>Prop Band (PB)</b>			
<b>Integ Time (<math>T_I</math>)</b> (min/rpt)	—	<b>5.8</b>	<b>3.5</b>
<b>Reset Rate</b> (rpt/min)	—	—	
<b>Deriv Time (<math>T_D</math>)</b>	—	—	<b>0.88</b>

Before testing for the closed loop response, go to **Tune | Options** tab and set Reset Action ON.

For each type of controller, enter the parameters, put the controller in Auto and test the loop for a 10% (of full scale) set point change.

Calculate or measure the decay ratio, period and (for PI controller only) the period-to-integral time ratio. (This will be used in a subsequent exercise.)

Also, for each type of controller, make a 5% load change. (Press **StepIncr** or **StepDecr**.) Mark which controller type has the best, and the worst, response to a load change.

	<b>P</b>	<b>PI</b>	<b>PID</b>
<b>Decay Ratio</b>			<b>0.16</b>
<b>Period, mins</b>			<b>9.2</b>
<b>Period</b> <b>Integral Time</b>	—		—
<b>Load change, best</b> <b>and worst response</b>		<b>Worst</b>	<b>Best</b>

**Table 1**  
Ziegler-Nichols Tuning Parameter Correlation For  
Closed Loop Process Data

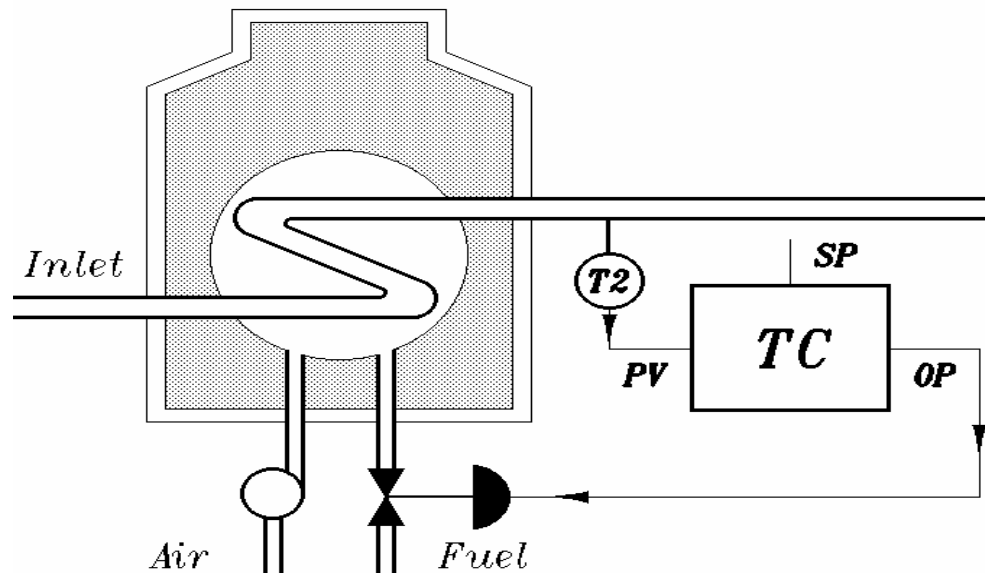
CONTROLLER	PROP ONLY	PI	PID
<b>GAIN</b> $K_C$	$0.5 K_{cu}$	$0.45 K_{cu}$	$0.6 K_{cu}$
<b>RESET</b> $T_I$ - Mins/Repeat	----	$\frac{P_u}{1.2}$	$\frac{P_u}{2}$
<b>DERIVATIVE</b> $T_D$ - Minutes	----	----	$\frac{P_u}{8}$

$K_{cu}$  = Controller Gain that causes sustained oscillation

$P_u$  = Period (in minutes) of sustained oscillation  
= Time between any two successive peaks

### Continuation from Exercise 9

Now that you have seen the closed loop tuning method it should be possible for you to return to the Feed Heater model in Exercise 9 and apply the closed loop tuning method in similar fashion to the open loop method



**Figure D10.1**  
Single loop control of feed heater

---

# Exercise 11

---

## Exercise in Improving ‘As Found’ Tuning

**This exercise has been kindly supplied to IDC Technologies by Wade Associates**

### Objective

To provide practice in improving “as found” tuning for proportional plus integral (PI) control of a self-regulating process.

### Background

Quite often a control systems engineer or instrumentation technician is called upon to improve the behavior of a loop that is currently in operation, but without resorting to either the open loop or closed loop testing methods. Assuming that the loop is not behaving acceptably at present, and that process and equipment problems (e.g. sticking valve) have been eliminated, then most persons resort to “trial and error” tuning. For novice tuners, this is often simply an exploratory procedure; “How about changing this knob in this direction and see what happens”.

This laboratory exercise presents a method for directed trial and error tuning, where each tuning parameter change is made for a deliberate reason. The objective is to go from the current unacceptable behavior to acceptable behavior as efficiently as possible; i.e., in the fewest number of tuning parameter changes.

This method is based upon the premise that if a PI controller, controlling a self-regulating process, is well tuned (that is, exhibiting a slightly underdamped oscillation with a quarter wave decay), then there will be a predictable relationship between the period of oscillation ( $P$ ) and the integral time ( $T_I$ ). This relationship (stated in three different ways) is:

$$1.5 \leq \frac{P}{T_I} \leq 2.0 \quad (1)$$

$$1.5T_I \leq P \leq 2.0T_I \quad (2)$$

$$0.5P \leq T_I \leq 0.67P \quad (3)$$

This premise leads to the following rule-based procedure:

1. If the loop is not oscillating, increase the gain, say by 25 to 50%.
2. If the loop is oscillating then:
  - 2.1 If the Period is between 1.5 and 2.0 times the integral time (or the period-to-reset ratio is between 1.5 and 2.0), then either increase or decrease the gain as required to obtain the desired decay ratio (such as quarter wave damping)
  - 2.2 If the Period is greater than 2.0 times the integral time (or the period-to-reset ratio is greater than 2.0), then choose a new integral time according to the criterion:
$$0.5P \leq T_i \leq 0.67P$$
  - 2.3 If the Period is less than 1.5 times the integral time (or the period-to-reset ratio is less than 1.5), then
    - 2.3.1 If the decay ratio is greater than 1/4, then decrease the gain, say by 25 to 50%, depending upon how much the decay ratio exceeds 1/4.
    - 2.3.4 If the decay ratio is less than, or approximately equal to, 1/4, then choose a new integral time, using the criterion given in 2.2.
3. After each adjustment, make a slight set point change to test the response to the latest combination of tuning parameters.

The essence of the rule-based procedure listed above is shown in flow chart form in a figure at the back of this laboratory exercise.

## 1 Running the program

Start **Windows**

Start **PC-ControLAB**

## 2 Loop Tuning

Select **Control | Retrieve Strategy, Model and Tuning/ PC Pracs/EX-11/** Highlight “Feedbck1.stg” (not “Feedback”) and press **Open**.

Observe from the top row that this opens the normal Feedback control strategy, as well as the Generic process model which you have worked with in previous laboratory exercises. The thing that is different here is that the loop has already been tuned – for better or for worse. Press **Tune** and note the existing tuning parameters.

If the PV scale is not in engineering units, select **View | Display Range | Engineering Units**.

Put the loop in **AUTO**.

Change the set point to 300 DegF.

*NOTE: The following two procedures will NOT product the same response:  
Make the set point change first, then put the loop into AUTO.  
Put the loop in AUTO first, then make the set point change.*

*For the purpose of the procedure described in this laboratory exercise, it is important for you to see the set point response with the loop already in AUTO. Therefore, the correct procedure is to put the loop into AUTO then make the set point change.*

Observe the response. Suppose that this is the behavior of the loop when you are asked to make tuning parameter changes. In other words, this is your “as found” condition.

Does the loop need to be retuned? \_\_\_\_\_

If so, list the “as found” conditions in the top line of the table below. Then use the procedure listed in “BACKGROUND”, or use the flowchart, to make tuning parameter changes. Keep track below of each tuning change you make. (Suggestion: Use set point values of 275 °F and 300 °F.)

Trial No.	Gain (See Note 1)	T <sub>i</sub> (See Note 2)	Period	Decay Ratio	$\frac{\text{Period}}{T_i}$
As found					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

**NOTE 1:** If you are working in Proportional Band, enter the value of Proportional Band, rather than Gain.

**NOTE 2:** If you are working in Reset Rate (repeats/min), rather than Reset Time (minutes/repeat), take the reciprocal of the reset rate to obtain Reset Time (T<sub>i</sub>).

### 3. A comparison

Many (novice) loop tuners, faced with the “as found” condition, would simply reduce the controller Gain until acceptable damping (e.g., quarter-wave decay) was achieved. We will demonstrate why that may not be a good idea.

#### 3.1 Re enter the original tuning parameters.

With the loop in Automatic, set the set point at 275 DegF and let the loop come to equilibrium.

Working between set point values of 275 and 325 DegF, adjust the Gain until quarter-wave damping is achieved. Do not change the reset.

Gain: \_\_\_\_\_

Reset (Min/repeat): \_\_\_\_\_

Suppose the product specifications require that the PV be within a certain tolerance above and below SP to be “within specs” then one of two criteria can be used to evaluate the response of the loop to a disturbance:

(1) Maximum deviation from set point, for a step disturbance of specified size;

(2) The time required for the loop to regain “on spec” production; that is, to get within the tolerance band with no further deviation outside the band.

Press [Zoom] and select a display range from 275 to 300.

Change the set point to 287.5 (This is midway of the chosen display range.). When the loop comes to equilibrium, press **StepIncr** to make a 5% disturbance (load change). **StepIncr**.

Record:

Maximum deviation of PV from set point: \_\_\_\_\_

How long before loop “settles down” to within 1/2% ( $\pm 2.5$  Deg) of SP \_\_\_\_\_

(At this Zoom range, the horizontal green lines on the display are 2.5 Deg apart. So the tolerance band is within one green line above and below SP.)

**3.2** Enter your final tuning parameters from part 2.0. (These should have produced a quarter wave decay response following a set point change, and a period-to-reset ratio which meets the criterion stated in “BACKGROUND”.

With the loop in Automatic, set the set point at 287.5 DegF and let the loop come to equilibrium.

Make a 5% disturbance (load change) by pressing **StepIncr**.

Record:

Maximum deviation of PV from set point: \_\_\_\_\_

How long before loop settles down to within 1/2% of SP \_\_\_\_\_

**3.3** Which tuning combination (Section 3.1 or 3.2) produced the best response to a load change?

---



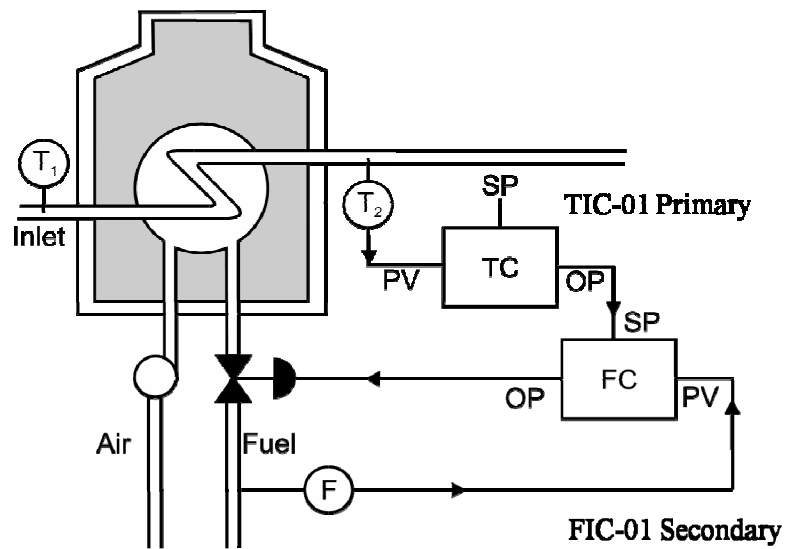
---

The tuning in used in Section 3.1 might represent the results of attempting to achieve acceptable damping by adjusting gain only, followed by an attempt to return to set point faster by making the reset faster. (Not wise choices!) The tuning in Section 3.2 is the result of “intelligent trial and error tuning.” The comparison shows less deviation from set point, as well as a faster return to “on spec” production.

# Exercise 12

## Cascade control operations

### D12.1 Objective

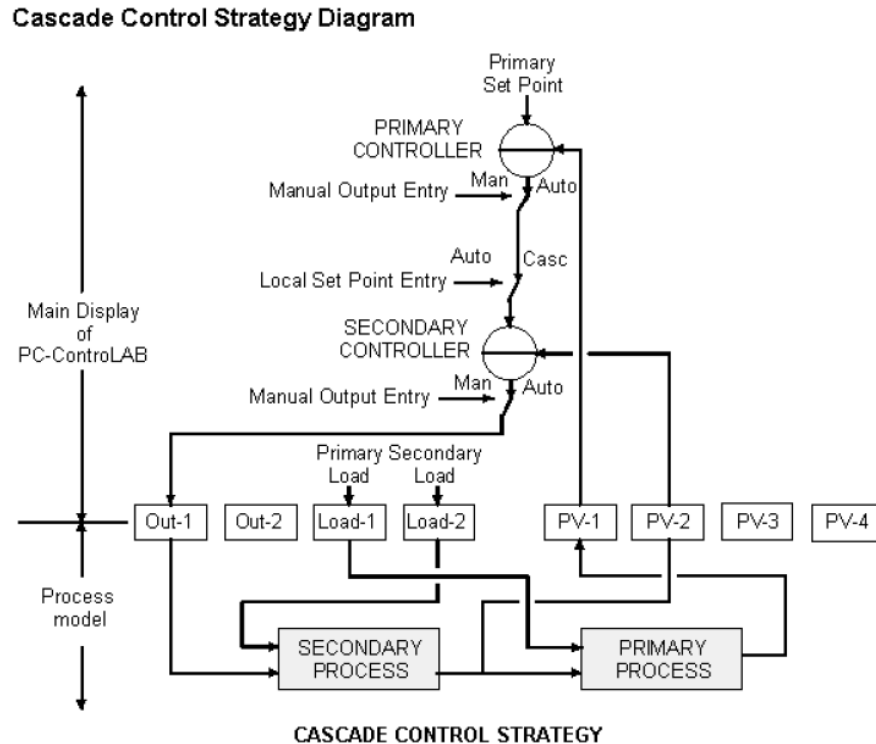


**Figure D12.1**  
*Cascade control arrangement for the feed heater*

The objective of this exercise is to gain some experience with the setting up and operation of cascade control. For this we will take the example of the feed heater we have been working with for single loop control and change it into a cascade control loop where the primary or outer loop is for the product temperature control with OP connected to SP of the secondary or inner loop controlling the flow of fuel gas to the furnace. This configuration is described in Chapter 9 and the arrangement within Exercise 12 can be seen in Figure D12.1. If you have studied chapter 9 the principles of cascade control should already be clear and all that remains is to set up the simulation and carry out some trial operations.

## D 12.2 Configuration of PC-ControLAB

Firstly it is important to be clear on how ControLAB implements and presents cascade control. This can be seen by using the help facility to display cascade control. Fig D12.2 shows the diagram that explains the controls.



**Figure D12.2**  
Cascade Control arrangement in ControLAB.

From Figure D12.2 you can see that the primary process will be the temperature control of the feed stream from the heater and this will remain as PV-1 as per the single loop version. The secondary process will be the flow control loop for fuel to the heater and PV-2 represents the flow. PV-1 has the range 0 to 1000 °C, PV-2 has the range 0 to 200 kg/hr of fuel flow.

### D12.3 Operation

Step 1: Open PC-ControLAB and load the process model PC Pracs/EX-12/IDC Feed Htr casc.mdl. Load the control strategy file PC Pracs/EX-12/ IDC Feed Htr PID Cascade.stg. Observe that this produces a cascade control display with two faceplates for PV-1 and PV-2.

Step 2: Keep the AutoLoad switch off at this stage to avoid any noise in the process. This model incorporates a load variable (LOAD 2) to allow flowmeter noise to be simulated. The secondary load trace (Green Trace) shows the load value and you should adjust this to approximately 50%.

Step 3: Begin by checking the open loop response of the flow loop and by checking the range of flow available is approximately 0 to 160 kg/hr. Now adjust its tuning to give you a tightly controlled flow using PD control. Remember to set up the PV noise filter under CONTROL/Measurement Options. Set the value to 1.00.

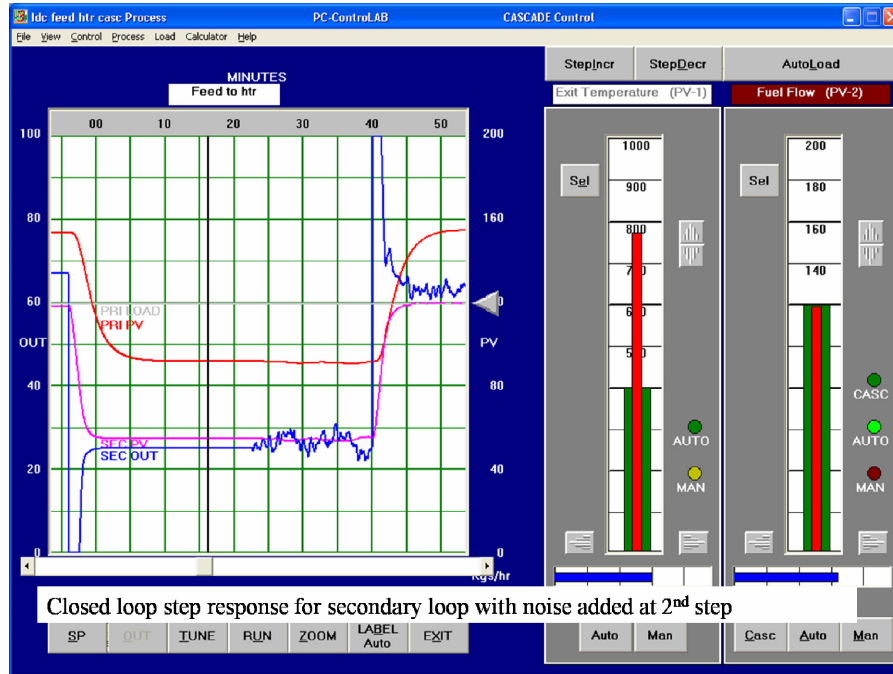
Note 1: You should avoid using integral action in the primary loop since this will introduce unwanted lag time or phase shift into the secondary loop process characteristics. Essentially the



secondary loop becomes part of the process under control as far as the outer loop is concerned. Therefore **switch off reset action** under **Control Options**.

Note 2: The control equation should be selected to ensure that the D action operates only on the PV (i.e. on measured value) and not on the error; otherwise any noise and fast changes applied to SP-1 from the outer loop will be amplified by the D term acting on the error SP-PV. Under Control Options select **PID non-interacting** and “**Derivative on measurement only**”.

Step 4: Carry out a step change to SP in Auto set to confirm you have established tight flow control. Switch on the noise load to confirm the stability of control. The result should look something like Figure D12. 3.



**Figure D12.3**  
*Secondary loop (fuel flow) closed loop step response with noise.*

Step 5: Now check the range of temperature in PV-1 that you can achieve by adjusting the flow set point over its range. Our model has its load set at 60% feed flow but if you change LOAD 2 you will be able to see the effect of product load changes. In cascade control applications you will not want the secondary loop set point SP2 to be driven to unrealistic extremes so it is necessary now to set limits on the output of the primary controller which is going to drive the SP of fuel flow control. Set the limits for OP-1 at 10% and 80% by selecting the primary control (Exit Temperature) and using the Control Options to set the limits on the output.

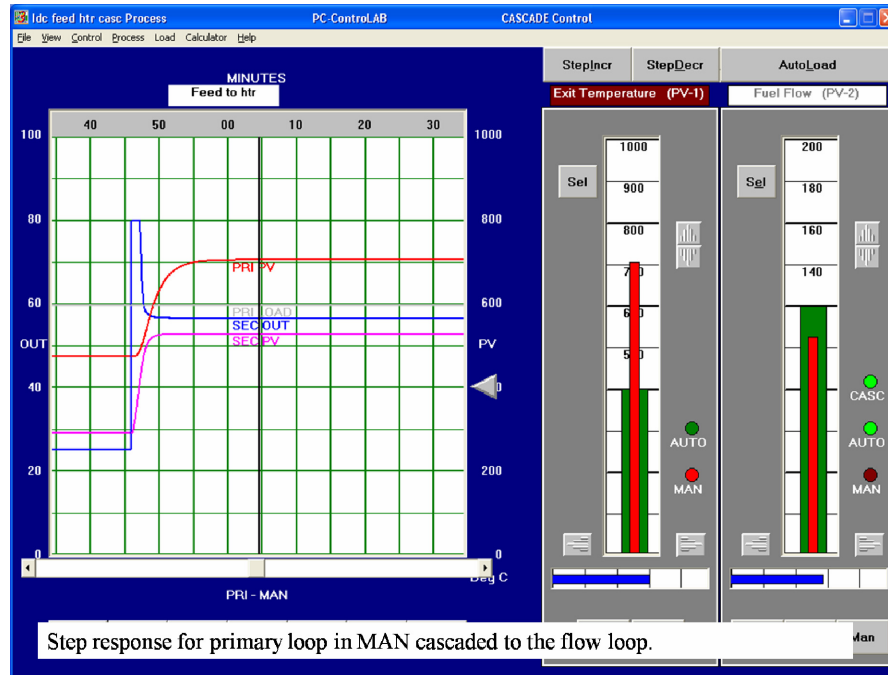
Step 6: Now, keeping both controllers in MAN set the secondary controller OP = 20% and notice that the primary controller OP is tracking the PV of the secondary controller. This PV tracking action ensures that when you switch the secondary to cascade mode the set point now coming from the primary controller OP will be matched to the secondary. This is known as “bumpless transfer”.

Step 7: Now you may try changing the secondary controller to CASCADE and you will be able to drive the fuel flow SP up and down by operating the primary controller’s output in MAN.

Step 8: Now it’s time to place the primary loop into AUTO. The tuning constants should first be set to the default values of  $K_p = 1.0$  and  $T_{int} = 100$  minutes. (This will give a very slow and mild response.) Remember to match the SP to the PV (Warning do not use SP tracking in this example).

Check it is turned of under Control/Options). Try some simple step responses to see that the controller is commanding the secondary flow control SP.

Step 9: Try tuning the primary loop to achieve a good response by the methods we have already tried before. Increase proportional gain until you see overshoots and then reduce Tint to eliminate offset in reasonable time. Finally add a small amount of D time to dampen the slow overshoots. Your results should look something like Figure D12.5. even with noise added. The tuning in this example was prop Gain = 1.5, Reset 2.00 minutes/repeat, Deriv: 0.05 Minutes.



**Figure D12.4**  
*Step response to SP change at primary loop with cascade control to the flow loop.*

## D12.4 Observation

Note how the SP of the secondary loop is driven as a slave to the primary loop. The secondary loop responds strongly to small changes and can often become noise prone due to its high gain.

In cascade control the secondary loop helps to provide a stable and fast response for the manipulated variable under all conditions. It is tuned for fast response and can greatly reduce non-linearity (effectively gain changes) due to the characteristics of the valve and its action on the process. This allows for easier manipulation of the slower primary loop which is tuned for accurate and stable response. The absence of integral action in the secondary loop avoids phase shifting and allows scope in the primary loop for strong integral action.

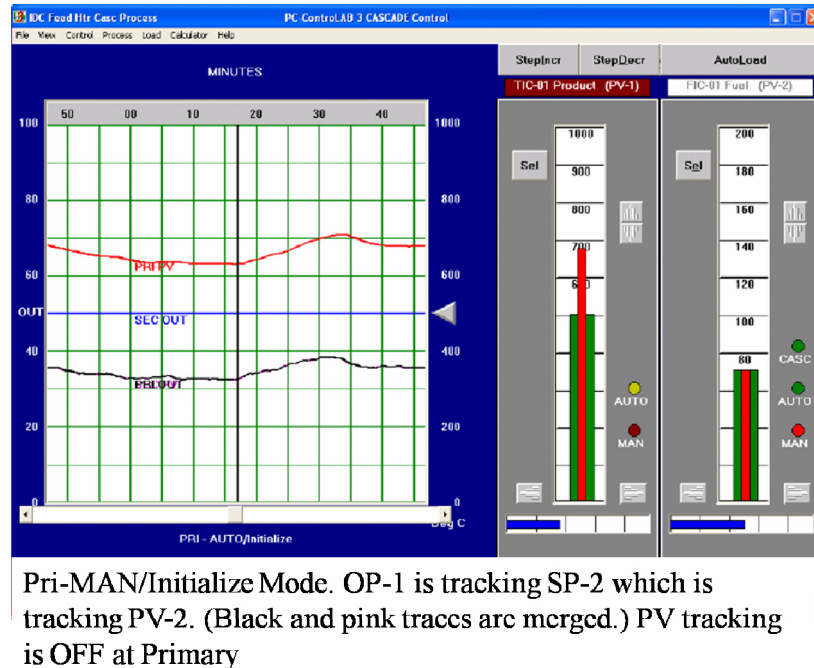
## D12.5 Initialization in cascade control

Here we consider the effects of PV tracking, Initialization and Mode changes in Cascade Control. This is easiest to learn simply by experimenting with the simulated feed heater cascade controls you have already set up and commissioned in this exercise

Start with the simulation running with the temperature loop (Primary) in AUTO and the flow loop (Secondary) in CASCADE. Check or adjust the secondary load (green trace) to 50%. Set the load random noise signal strength for both loops under LOAD/AUTO LOAD to Max Size 0.3% and Correlation 0.3.

Switch on the AutoLoad for both the temperature and the flow variables. Make the SPE = 500 degC and allow the process to settle down. Now change the flow loop mode to MANUAL and OP2 to 50% of range.

You will see that this causes the temperature will rise to around 700 degrees as you increase fuel flow. The temperature set point remains at 500 degrees but the OP of the primary control is following the SP of the flow control and the Auto action of the temperature controller has been suspended because it has no valid controller to feed to. This prevents the reset action from “winding up”. The display indicates “PRI-AUTO/initialize”. See Figure D12.5.



**Figure D12.5**  
*Example of Pri-MAN/Initialize Mode*

Now change the primary to MAN and note that the display reads PRI-MAN/Initialize but the OP continues to track the SP of the flow control.

The flow controller has set point tracking in force so its SP is moving with the flow PV. The result of this action benefits the moment when you want to switch back to Cascade mode because the SP of the flow control does not have to “bump” to a new remote SP value. Now place the flow control into CASCADE and see that this is so.

Now because there is no PV tracking on the temperature loop the SP is far away from the present PV. If you change to AUTO on this loop the process will be sharply disturbed as the controller action kicks in. PV tracking is optional for this loop. Try setting it on and repeat the exercise. Observe that the temperature SP now tracks the PV as you ramp up the fuel flow in MAN mode.

## D12.6 Observation

When we change from CASC control to AUTO or MANUAL some variables are unpredictable as discussed earlier. When changing back to CASC control, these unpredictable variables might cause a bump within our control system, when they have to change back to real and defined values.

## D12.7 Conclusion

Based on the observation that unpredictable values of OP1, SPE1 and SPE2 cause a bump in control when we change to CASC control, the following requirements have to be met to avoid this.

The secondary controller's SPE2 has to follow PVE2 as long as the controller is in MANUAL mode (PV Tracking). The same requirement for PV Tracking should also be made for the primary controller although this is less critical. In addition, we cannot permit OP1 to assume any unwanted value when the secondary controller is either in MANUAL or AUTO mode. For this reason OP1 has to assume the same value in % as SPE2 has, as long as the secondary controller is not in CASC mode. This is called Initialization. In our example, the flow controller (Secondary controller) is the initializing controller and has to be configured as such. The temperature controller is the initialized controller.

We come to the conclusion, that it is advisable to make use of PV Tracking and Initialization whenever possible. Special care is necessary when deciding on the use of PV Tracking in a primary controller. The incorrect use of PV-Tracking may cause some operational problems.

---

## Exercise 13

---

### Exercise in Ratio Control

**This exercise has been kindly supplied to IDC Technologies by Wade Associates**

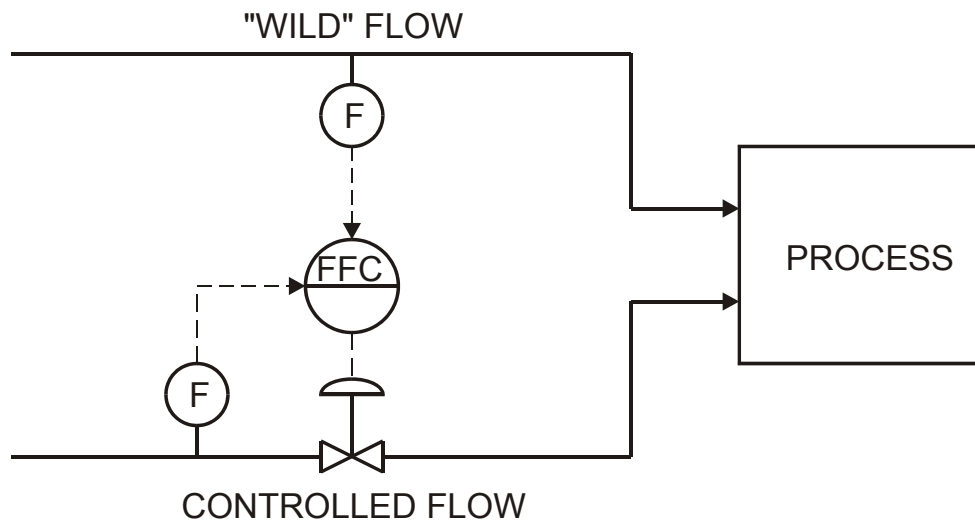
**Objective:** To demonstrate the behavior of ratio control.

**Prerequisite:** Completion of PC-ControLAB tutorial (under **Help | Tutorial** ) or an equivalent amount of familiarity with the program operation.

**Background:** Ratio control is used when it is necessary to maintain a certain ratio between the flow rate of two streams. One stream, called the “wild” flow, is measured only. It is the pacing stream. The other stream, the controlled flow, is controlled so as to maintain a specified ratio between the two.

There are two general types of ratio control. In one type, illustrated in this laboratory exercise, the ratio is manually set. In the other type, the ratio is automatically set by the output of a Primary feedback controller.

A typical configuration of ratio control is shown in the following figure. For illustrative purpose, this laboratory exercise assumes that the controlled flow is Steam Flow, with a measured range of 0–1000 P/hr. The “wild” flow is a process stream, with a measured range of 0–400 gpm.



1. **RUNNING THE PROGRAM**

Start **Windows**. Run **PC-ControLAB**.

Click on **Control / Select Strategy / Ratio**.

Click on **Process / Select Model/ PC Pracs/EX-13/ highlight “RatioFlo.mdl”** and press **Open**

2. **CONTROLLER SET UP**

Press **Tune** and enter the following tuning parameters (representative of a flow loop).

Gain:	0.5
Reset:	0.15 min/rpt
Deriv:	0.00 min

Put the controller in **Auto**.

Observe:

Present value of process variable, in engineering units \_\_\_\_\_

Present value of process variable, in percent of measurement span: \_\_\_\_\_

Present value of “wild” variable, in engineering units  
(gray trace, labeled LOAD) \_\_\_\_\_

(Click on the label “GPM” to change the vertical scale from the PV to the wild flow.)

Present value of “wild” flow, in percent of measurement span: \_\_\_\_\_

Ratio,  $\frac{\text{PV, \% of span}}{\text{Wild flow, \% of span}}$  : \_\_\_\_\_

Click on the label “Steam Flow (PV-2)” above the faceplate.

Click on the button labeled **R** on the controller faceplate to access the present calculated ratio, as well as the set ratio.    Calculated ratio: \_\_\_\_\_

Set ratio: \_\_\_\_\_

*In concept, the set ratio back could be calculated from the present ratio of measured variables; this would provide bumpless transfer from the AUTO mode to the RATIO mode. This would probably not be wise, however, since the required ratio is normally set by process conditions or product specifications. The operator, not the system, should be the one to enter the required ratio. Therefore, the set ratio is not backcalculated in this program or in most commercial systems.*

*When the ratio is set by the output of a feedback controller (for instance, in the Multiplicative Feedforward control strategy – see Exercise 17), the ratio is back calculated, thus producing the output for the Primary controller. This provides for bumpless transfer from Manual to Auto for the Primary.*

If you change the controller to the Ratio mode now, where do you think the controller set point will go? \_\_\_\_\_

Try it! Put the controller in Ratio. What is the SP? \_\_\_\_\_

*You probably observed that that made quite a “jolt” to the flow loop. A wise operator probably would not change from Auto to Ratio when there was that much difference between the existing ratio and the ratio setting. Instead, he/she would probably set the ratio setting to whatever the application or product specifications required, then adjust flow set point until actual ratio met the required ratio. Then the switch to Ratio mode would be made.*

Press **R** and enter a set ratio of 0.5.

After the loop stabilizes, observe:

Present value of PV, in engineering units \_\_\_\_\_

Present value of PV, in percent of measurement span: \_\_\_\_\_

Present value of “wild” variable, in engineering units \_\_\_\_\_

Present value of “wild” flow, in percent of measurement span: \_\_\_\_\_

Ratio,  $\frac{\text{PV, \% of span}}{\text{Wild flow, \% of span}}$  : \_\_\_\_\_

Press **StepIncr** twice to simulate an increase in “wild” flow.

Did the controlled flow (PV) follow it at the set ratio? \_\_\_\_\_

Press **StepDecr** twice, then press **AutoLoad**. This simulates a randomly varying wild flow.

Does the controlled flow (PV) follow it at the set ratio? \_\_\_\_\_





---

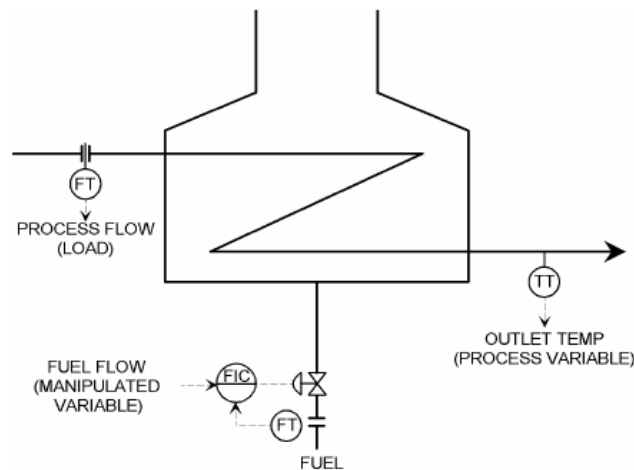
# Exercise 14

---

## Combined feedback and feed forward control

### D14.1 Objective

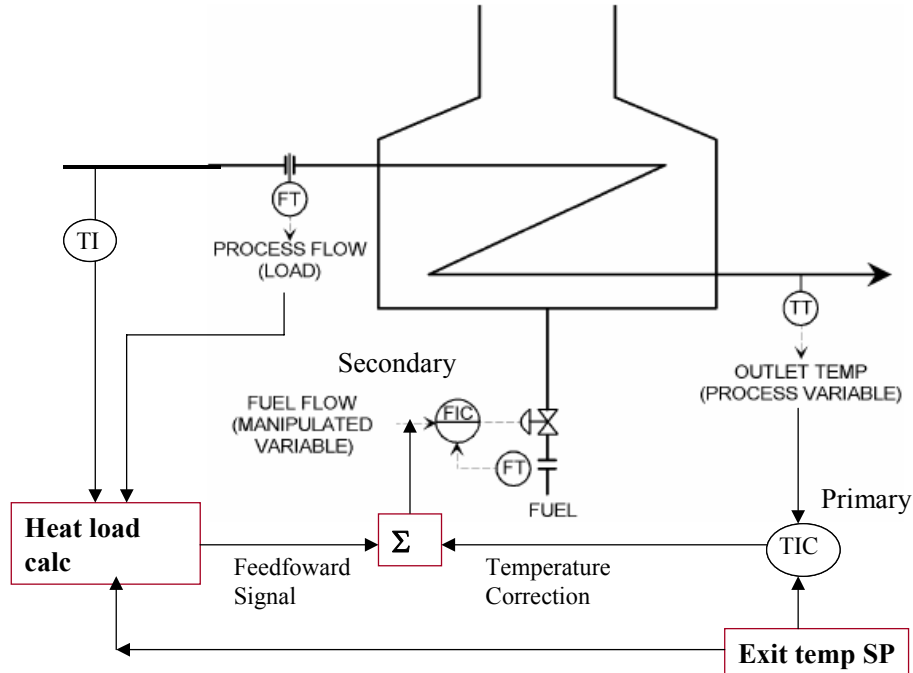
This exercise is provided to explore combined Feedback and Feed Forward control. The intention of this section is to introduce this control strategy. Although it is not possible to go into all aspects of this topic in this exercise, the control of the training applications have not been simplified and is realistic. The control example provided here is "Feedheater Control" (Figure 14.1). Later in this exercise we will look at a similar design for boiler drum level control using the exercise provided by Wade Associates



**Figure D14.1**  
*Basic feed heater process*

## D14.2 Process and control model

In exercise 12 we used a feed heater example such as this to practice with cascade control. However in that model changes to feed flow were not measured directly and we had to wait for their effect on the exit temperature before the control system could make any corrections. In the feed forward configuration as shown in Figure D14.2 the heat load presented by the incoming feed and the required temperature rise is calculated as a fuel flow requirement and added directly to the output of the primary controller. When the secondary loop is in Cascade mode the primary loop output then creates a more accurate and responsive set point for the heating fuel flow.



**Figure D14.2**  
Feed heater with feedforward signal based on heat load and feedback based on exit temperature

In this example the feedforward controller uses the changes in feed flow (and feed temperature) as a heating load change to predict the corrections to fuel flow set point.

### *A few points to remember:*

The main control is Feed Forward control, whenever all major disturbances are used to calculate the corrective action. In most cases, Feedback control serves as a long term correction. In essence this means that it should act against the deviations of the PV from the set point which feed forward cannot be expected to correct. Therefore, tuning should be done in the following order.

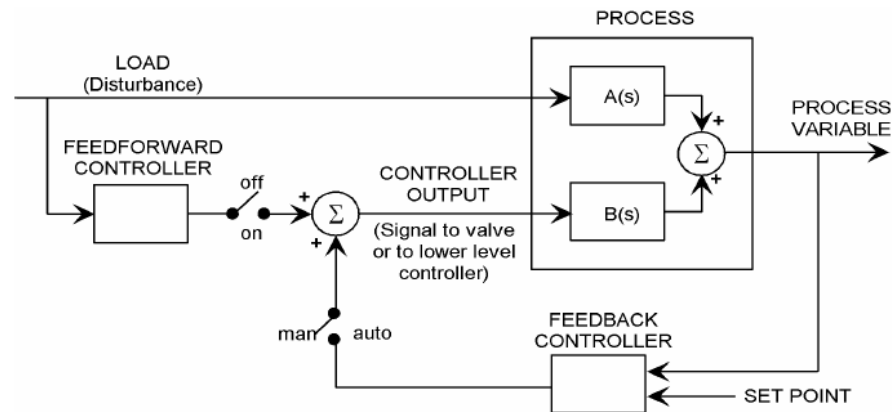
- Firstly, the flow controller which is common for Feedback and Feed Forward control has to be tuned.
- Secondly, Feed Forward control and finally Feedback control has to be tuned. The Tuning of Feedback control should aim towards minimum Feedback control action; just enough to eliminate the process drift. Anything more adds to the wear and tear of equipment without significantly improving control results.

### D14.3 Tuning Method

For this exercise we shall use the heater process model made available by Wade Associates with the ControlLAB simulation package. The following exercise steps are based closely on the details kindly supplied to IDC Technologies.

#### Description of the feed forward configuration

After starting the ControlLab program select Control/Retrieve Strategy Model and Tuning and then open the file “IDC FEEDFWD HTR.stg”. This action incorporates the process model file “IDC FEEDFWD HTR.mdl”. The configuration of this cascade + feedforward system is shown in figure D14.4



**Figure D14.3**  
General arrangement of the feedforward controller

The strategy for setting up the feed forward controller is to measure the open loop response from the feed flow to the exit temperature to find the process transfer function  $A(s)$  and to find the transfer function of the fuel flow to temperature  $B(s)$ .

From this the gain, lead and lag time constants of the feedforward controller (i.e. the transfer function  $FF(s)$ ) can be estimated using the relationship shown in Figure D14.4.

The following boxes summarize the procedure to be followed to arrive at the settings for the feedforward controller. You should be familiar with obtaining the parameters from exercise 2 and you may use those notes to revise the method.

Setting Up Notes: 1

Find parameters for process response through path A from load change to exit temperature  
 Load change  $\times A(s)$  = Temperature change

$A(s) = \text{Gain (Kpa)} \times \text{Lag (Tlga)} + \text{deadtime (Tdta)}$

$$A(s) = \frac{K_{pr} \cdot \exp(-T_{dta} \cdot s)}{(T_{lga} \cdot s + 1)}$$

Find Kpa, Tdta and Tlga from step reponse.

**Figure D14.4**  
 Finding the process parameters for  $A(s)$

Setting Up Notes: 2

Find parameters for process response through path B from flow set point change to exit temperature  
 Flow SP change  $\times B(s)$  = Temperature change

$B(s) = \text{Gain (Kpb)} \times \text{Lag (Tlgb)} + \text{deadtime (Tdtb)}$

$$B(s) = \frac{K_{pb} \cdot \exp(-T_{dtb} \cdot s)}{(T_{lgb} \cdot s + 1)}$$

Find Kpb, Tdtb and Tlgb from step reponse

**Figure D14.5**  
 Finding the process parameters for  $B(s)$

Setting Up Notes: 3

Find parameters for FF (s)

General FF (s) =  $K_f \frac{(T_{ld} \cdot s + 1)}{(T_{lga} \cdot s + 1)} \exp(-T_{dt} \cdot s)$

$$FF(s) = \frac{-A(s)}{B(s)}$$

$$FF(s) = \frac{-K_{pa} \cdot \exp(-T_{dta} \cdot s)}{(T_{lga} \cdot s + 1)} \times \frac{(T_{lgb} \cdot s + 1)}{K_{pb} \cdot \exp(-T_{dtb} \cdot s)}$$

From above deduce gain  $K_f$ , lead  $T_{ld}$ , lag  $T_{lg}$  and dead time  $T_{dt}$   
 Insert these values in the FF controller tuning panel  
 Place into FFwd mode and test 5% disturbances in load.  
 Fine tune values for minimal disturbance.

**Figure D14.6**  
 Finding the process parameters for  $FF(s)$

Once we have found the transfer function parameters for  $FF(s)$  these are to be loaded into the FF tuning page on the controller display after which response tests can proceed. What we are trying to achieve is a response from the Ffwd stage that will very closely compensate the flow set point for changes in the load.

## D14.4 Operations

Step 1: Start PC –ControLAB and select the Control Strategy PC Pracs/Ex-14/IDC FEEDFWD HTR.stg

Step 2: Confirm that you have the process model “ IDC FEEDFWD HTR ”.(See PCControLAB title bar) Note the process temperature range is 0-500°F and fuel flow range is 0-100 GPM

Step 3: Set up the secondary controller tuning with Gain = 05. and Reset = 0.15 minutes/repeat. Now test the step response in AUTO (press ON when Fuel Flow Control is selected) and verify that it is fast and stable. Leave the loop in AUTO (ON) with SPE-2 = 30

Step 4: Switch the display to the primary controller and make sure Ffwd is OFF and the mode is MAN. (indicates yellow). Changing feed load will now only affect PV-1. Set the PRI LOAD to 60%. The temperature should be settled around 360°F.

Step 5: Make step changes of 5% in the load by using the Step buttons and zoom the display so that you can measure the step response of the temperature PV-1. from this curve and using the methods described in Exercise 1 or 8 obtain the parameters for the transfer function  $A(s)$  as described in Figure D14.4 above.

Gain  $K_{pa}$ .....  
Lag time constant  $T_{lga}$ .....  
Dead time  $T_{da}$ .....

Step 6: Obtain the  $B(s)$  transfer function by testing a step response of the secondary controller to a step change in SPE-2 by placing the primary controller in MAN and secondary controller in CASC. Make a D14% downward step change in the OP-1 (30% to 20% ) to create the step response and obtain the parameters for the transfer function  $B(s)$  as described in Figure D14.5 above.

Gain  $K_{pb}$ .....  
Lag time constant  $T_{lgb}$ .....  
Dead time  $T_{db}$ .....

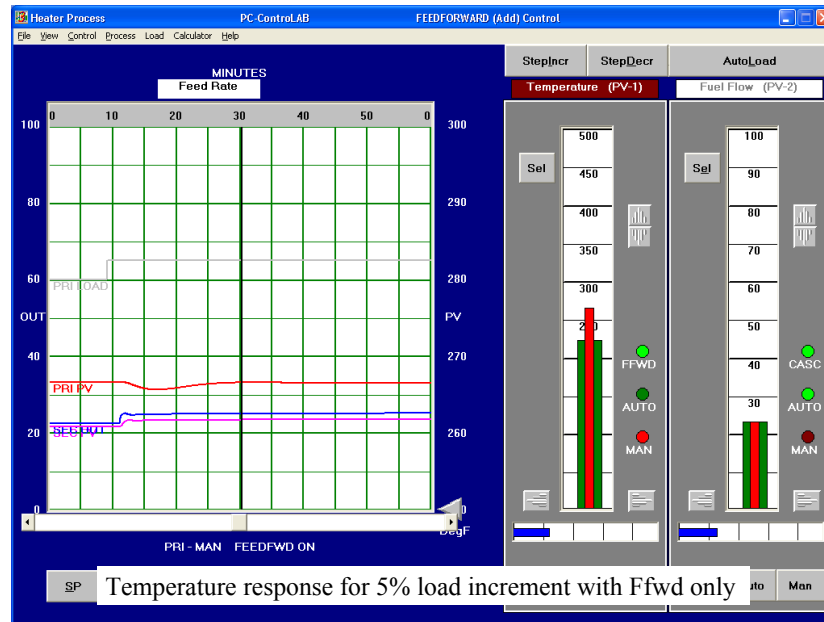
Step 7: Now arrange the two transfer functions as shown in figure D14.6 above to extract the parameters for  $FF(s)$  and load these into the feedforward controller settings under TUNE and under the tab “ Feedfwd”.

Step 8. You should now be able to test the response of the feedforward controller. Keep the secondary controller in CASC and leave the primary control in MAN so that there is no corrective action from the temperature feedback controller. Press the Ffwd button to connect the feedforward signal to the set point of the fuel flow control. The temperature (PV-1) should remain steady and if you now step the load upwards by 5% you should see that the fuel flow steps upwards approximately 2 minutes later as the dead time allowance in the Ffwd controller expires.

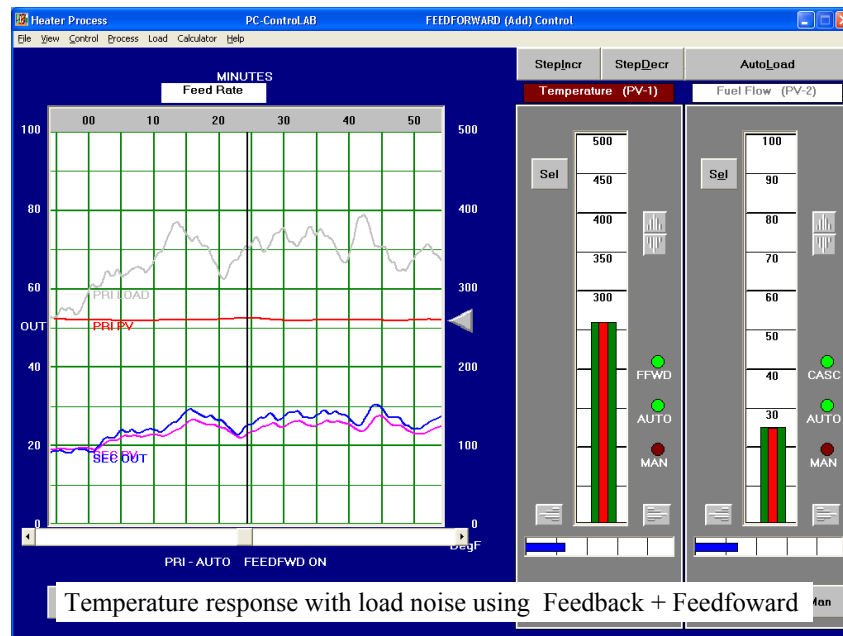
If the tuning is good PV-1 should show only a small disturbance and ideally it should not move more than a few degrees. Some adjustment of gain may be needed to improve the matching. See Figure D14.7 for a typical result.

Step 9: Now that you have a good predictive control function, the temperature feed back control can now be introduced to provide the slower acting correction to fuel feed to bring the exit temperature into alignment with the required exit temperature set point . Using the step response data from the  $B(s)$  test you may use the Zeigler-Nicholas formula to arrive at settings for gain and

reset actions or you may keep the gain low and use moderate integral and derivative action to give you a stable corrective action. Now you will see that the process operates close to set point even when the AutoLoad noise is switched on. See Figure D14.8. You can also switch the feedforward in and out to see how the FF signal helps to reduce the variability in PV in the presence of disturbances and noise in the load.



**Figure D14.7**  
Temperature response in feedforward only control, 5% load change.



**Figure D14.8**  
Temperature response with load noise using feedback + feedforward control.

## D14.5 Observation

We have seen how feedforward signals can be added to feedback signals to compensate the effect of disturbances in the loading process. It should be noted that this method depends on the relationship between loading and the corrective action being constant in gain and in dynamic lags. In the heater model the gain is not constant over the full temperature range and it is highest at low loads and high temperatures. However the differences are relatively small and the controller remains stable over the whole range. Where this is not the case the sensitivity of the feedforward controller may have to be changed as the operating point changes. This fairly easy to do but it is important to aware of the likelihood of this condition in practical applications.

The success of feedforward control also depends on the ability to measure the disturbances affecting the process. If we cannot detect the disturbances with a sensor we cannot build in the compensation.

## D14.6 Conclusions

Feedforward control with feedback can be a very effective control strategy especially where the process is subject to large disturbances in the process and where the load parameters can be measured. Processes with long dead times and long lags will benefit particularly.

Non-linearity in the process will make the matching of feed forward signals change with the conditions and this may lead to instability problems unless variable gains are employed.

The arrangement of controllers in Cascade with a Feedforward switch allows users to easily operate with or without feedforward as required and this in turn assist the tuning and setting up procedures.

## D14.7 References and Tuning Answers

This exercise is based on the exercise produced by Wade Associates which can be accessed in the IDC Workshop as filename: 16\_AddFf.pdf. The tuning results obtained in the Wade exercise are presented in the file 16\_AddFf.pdf with some additional notes. The tuning values found from tests and used in this version of the exercise are as follows:

Feedforward Gain  $K_f = 0.36$   
 Lead Time (mins) = 8.00  
 Lag Time (mins) = D14.00  
 Dead Time (mins) = 2.00

## D14.8 Further applications

Two very widely seen applications of feedforward additive control are to be found in typical large industrial boilers. These are :

Balanced furnace draft control: Furnaces in boilers and other large combustion systems often have to be controlled at very slightly negative pressures relevant to atmospheric pressure (i.e. small negative gauge pressure). Feedforward control senses the air flow to the furnace and adds a value to output of the furnace draft pressure controller to change the suction fan (ID Fan ) speed or dampers in anticipation of a pressure change.

Boiler drum level control uses steam flow output to drive the feedwater inlet flow to balance the boiler drum water inventory whilst the level controller is used to trim the level in the presence of large transients. This example is the subject of Exercise 15 overleaf.





---

## EXERCISE 15

---

# Application of feedforward to control of boiler drum level

This exercise has been kindly provided to IDC Technologies by Wade Associates for application using their PC-ControlLAB simulation package.

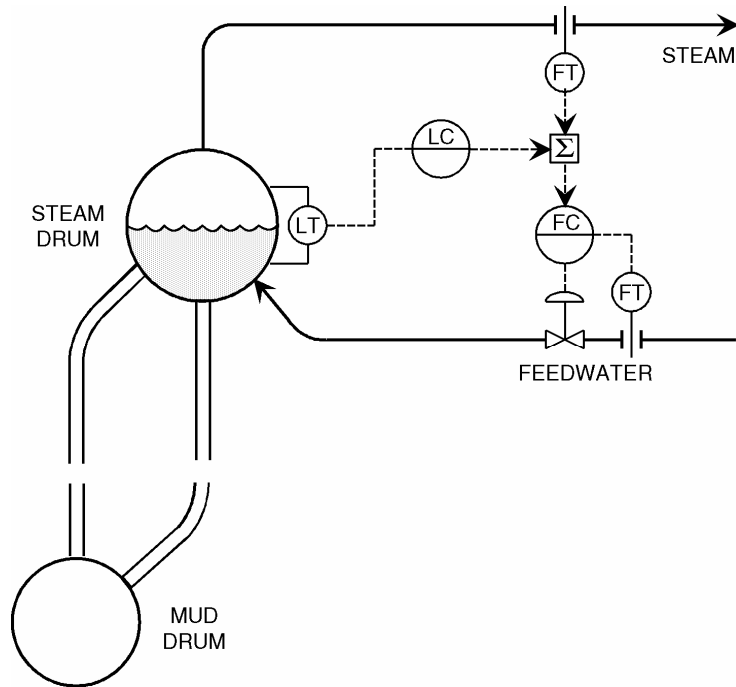
**OBJECTIVE:** To illustrate a liquid level control phenomena, known as “shrink and swell”, that is very common with steam boilers.

**PREREQUISITE:** Completion of the following exercises:

- Tuning Liquid Level Control Loops
- Characteristics of Cascade Control
- Characteristics of Additive Feedforward Control

**BACKGROUND:**

This section will demonstrate the control of a boiler drum level in the presence of the phenomena known as "shrink and swell." The first type of control will be standard level control, cascaded to feedwater flow control. The second type will be three-element drum level control in which the steam outflow rate will be measured and will adjust the feedwater flow controller by feedforward control. The level controller output will be used as feedback trim. See Figure D15.3.



**Figure D15.3  
THREE ELEMENT CONTROL  
DRUM LEVEL CONTROL**

## RUNNING THE PROGRAM

Start **Windows**.

Start **PC-ControLAB**.

### 1. DRUM LEVEL CONTROL

#### 1.1 Control with Normal Cascade

Select and Open: **Control / Select Strategy / PC PRACS/ EX-15/ DRUM LEVEL FFwd.stg**

Check that the Process Model has opened as “level2” Process.

Select **Process / Change Parameters**. Set or confirm the following:

<u>Parameter Label</u>	<u>Value</u>
Valve Pos: 0=No; 1=Yes	1.0
Cont Inflo=1; Cont Outflo= -1	1.0

Enter the following tuning values:

#### Secondary Controller

Gain	0.5
Reset	0.15 minutes/repeat

Primary Controller

Gain	0.65
Reset	2.0 minutes/repeat

Put the Secondary controller in Cascade. Put the Primary controller in Automatic. For the moment, leave **Ffwd OFF**.

Change the Primary controller set point by 10%. The response should be essentially the same as in Section 3; that is, quarter cycle decay with a period of about 30 minutes. (In an actual application, the drum level response would be much faster than this.) When the loop has come to equilibrium, put the Primary controller in Manual. With the Primary controller selected, press StepIncr to simulate an increase in outflow. After the Primary PV has decreased about 15%, press StepDecr.

Did the Primary PV (level) immediately begin to decrease when **StepIncr** was pressed?

\_\_\_\_\_

Put the Primary controller in Automatic. When the PV comes to equilibrium at the set point, then put the controller back in Manual.

*So far we have not added the simulation feature that creates the drum level “shrink and swell” effect. We will add this feature now and repeat the above test. Through the **Process | Change Parameters** table, select the parameter labeled “BLK 57 LEAD TIME.” (Be sure NOT to select “BLK 57 LAG TIME.”) Change the parameter value from **+1.5 to Minus 3.0**. Press **[OK]**.*

With the Primary controller selected and in Manual, press **StepDecr** to simulate a decrease in outflow. When the Primary PV has increased about 15%, press **StepIncr**.

Did the Primary PV (level) immediately begin to increase when **StepDecr** was pressed?

\_\_\_\_\_

Did the Primary PV (level) immediately stop increasing when **StepIncr** was pressed?

\_\_\_\_\_

You should have observed that when StepDecr was pressed, the level initially decreased, then began to increase. This is known as drum level “shrink”. Also that when StepIncr was pressed, then drum level increased further before settling to an equilibrium value. This is known as drum level “swell”. With straight level-to-flow Cascade control, when the steam flow rate is decreased, due to the “shrink” effect, the Primary (level) controller will initially sense a decrease in level, consequently it will increase the set point of the Secondary (feedwater flow) controller. The correct response should have been to decrease the feedwater flow rate.

*Thus, due to the shrink-and-swell phenomena, pure Cascade control for this application will produce poor or unacceptable performance, particularly if the same tuning parameters are maintained.*

Adjust the Primary controller SP to match the current PV, then put the Primary controller in Automatic. (Leave Feedforward Off.) Change the set point to 5% above the current PV.

Is the loop behavior acceptable? \_\_\_\_\_

If the answer is NO, then we will significantly loosen the Primary controller reset tuning parameter. Enter the following for the Primary:

Gain: 0.65  
Reset: 8 minutes/repeat

Is the loop now more stable? \_\_\_\_\_

*Your answer should be "YES", but you should also observe that the loop is very sluggish, taking an extremely long time to stabilize at set point.*

Period of oscillation: \_\_\_\_\_

When the loop is at (approximate) equilibrium, press **AutoLoad** to activate automatic, random load changes. Observe the loop for at least 60 (simulated) minutes.

Does the level stay near to the set point? \_\_\_\_\_

*We will now implement the three element drum level control scheme, using the steam flow measurement as a feedforward signal.*

## 2.2 Three Element Drum Level Control

Select the Primary controller, press **Tune**, then select the **Feedfwd** tab. Enter 1.0 for Feedforward Gain. Leave the other parameters at 0.0. Press **Clear**.

At the Primary controller, put Feedforward control ON (Press **FFWD**.) Observe the loop for at least 60 (simulated) minutes, or longer.

Does the level eventually come to the set point? \_\_\_\_\_

Once near the set point, does the level remain near to the set point?

See overleaf for answers to the above points:

IDC Note: The feed forward technique applied in the exercise greatly improves the response of the level to disturbances in steam load flow as intended for boiler level control. Note however that the tuning of the level control primary loop remains sluggish in response to level changes. This is normally acceptable for boilers since there is usually no requirement to change the set point of the drum level.

## Answers to the exercise in Drum Level Control as supplied by Wade Associates

2. DRUM LEVEL CONTROL
- 2.1 Control with Normal Cascade
- Did the Primary PV (level) immediately begin to decrease when StepIncr was pressed? Yes
- Did the Primary PV (level) immediately begin to increase when StepDecr was pressed? No
- Did the Primary PV (level) immediately stop increasing when StepIncr was pressed? No
- Is the loop behavior acceptable? No
- Period of oscillation 50 minutes
- Does the level stay near the set point? No
- 2.2 Three Element Drum Level Control
- ... eventually come to the set point? Yes
- ... remain near to the set point? Yes



## Exercise 16

# Dead-time compensation in feedback control

### D16.1 Objectives

This exercise will give some practical experience in the Closed Loop Control strategy necessary if a long Dead-Time is part of the process to be controlled. Figure D15.1 shows the block diagram of a control loop with added process simulation as a means of overcoming process Dead-Time.

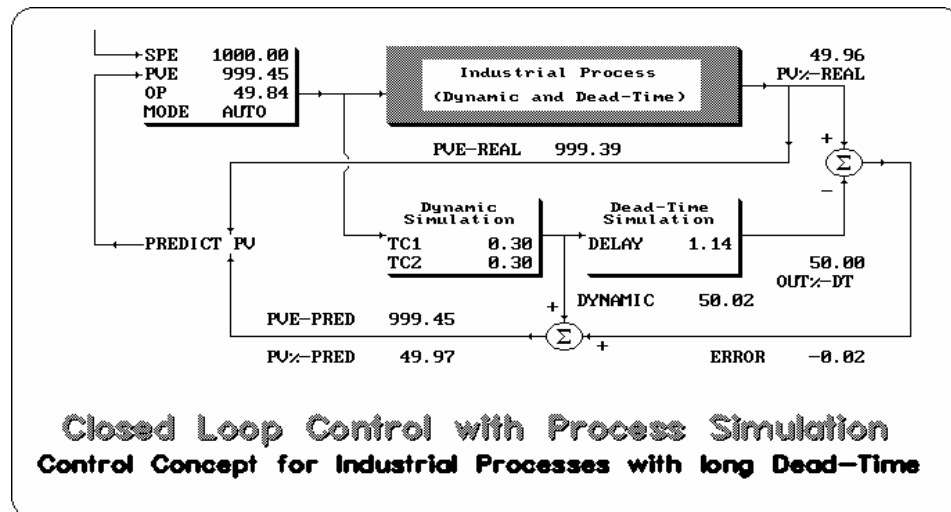


Figure D16.1  
Block diagram

#### Note concerning terminology

Within this exercise, we have a conflict of terminology. As we have no physical industrial plant to control, we have to simulate the behavior of such an industrial plant. If one of the tools of controlling an industrial plant requires a process simulation, we would not know what simulation is meant if we do not make a clear distinction between the process simulation acting as a stand-in on behalf of the real process, and the simulation acting as a tool of control.

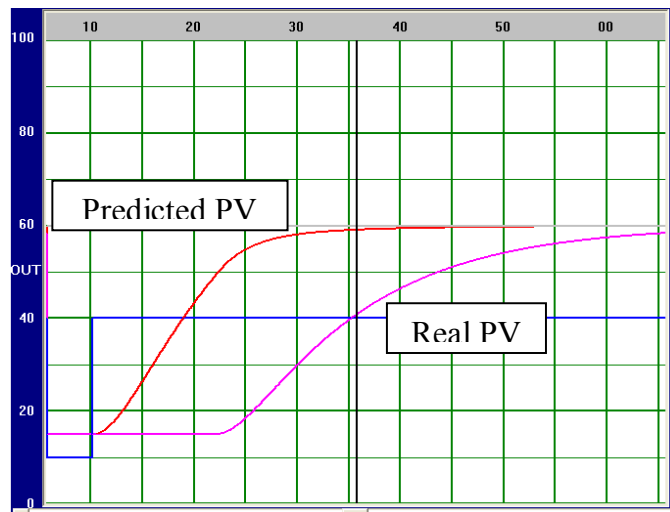
Therefore, in the description of this exercise, the term simulation is never used if referring to the real process; the terms for the real industrial process will be used here instead. In this exercise, the term simulation is used only for a process simulation which exists in addition to the industrial process and would exist in a real industrial plant for control purposes as well.

## D16.2 Operation

Step 1: Start the PC-ControlLAB programme and open the process model “ application **”IDC Ex 16.mdl”**. The tuning parameters for the process simulation are in the PROCESS/ “Change Process Parameters” tool bar. They are K-SIM, Simlag 1, Simlag 2 and SIM Dead Time. The values have been initialized to the correct values required to match the real process as close as possible. The process has two lags in series of 0.2 and 0.1 minutes and a dead time of 0.2 minutes.

Step 2: Set the Horizontal Grid Scale (under View) to “seconds” and set the variables display to show PV- and PV-2. PV-2 is PV-Real but the control loop is connected to PV-1 which is the predicted PV-PRED. Thus the feedback control operates on predicted PV value but in the steady state this value becomes the same as PV-Real.

Step 3: Then, on the Controller Display set MODE to MANUAL. Set the OP to 10% and wait for the process to settle down. Make a change of OP from 10% to 40% of range. See Figure D16.2.



**Figure D16.2**  
*Real PV and predicted PV reaction curves*

## D16.3 Observation

Observe the different reaction curves of the process variable coming from the real process (PVE-REAL) and simulated process variable (PVE-PRED). It can be seen that the reaction of the predicted process variable is similar, but if compared to the real process variable there is no dead-time. This is a simple form of model predictive control in which the predicted PV may be used as the feedback signal for a PID control loop instead of the actual PV. Since both forms of PV reach the same steady state value the predicted PV allows for some level of conventional tuning to be applied whilst reducing the effects on stability of the long dead time.



## D16.4 Operation

The principle of the dead-time compensator model is that the dynamic response of the model has been split into the part without the dead time and a separate part including the dead-time. The pure lag signal may then be used in feedback for a period of time until the real PV signal change develops. A small proportion of the real PV is added to the predicted PV to produce a combined PV that resembles a first order lag response. If there are errors between the dead-times of process and model these are calculated in the dead-time stage of the model and added to the output as a correction signal that will fall away as the disturbance decays.

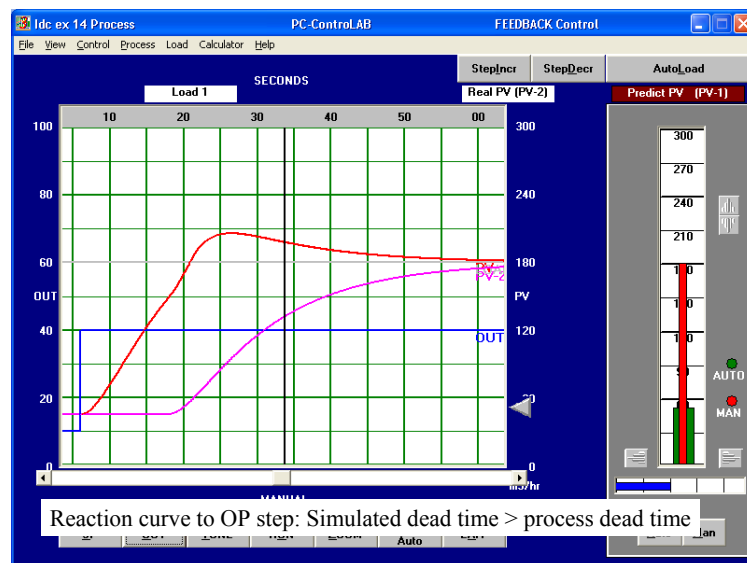
In order to explore the purpose and impact of the model parameters, do the following for each parameter (for K-SIM, Simlag 1, Simlag 2 and Sim Dead time):

- Make sure, the controller is in MANUAL mode.
- Change the tuning constant.
- Make a step change to the OP.
- Carefully observe the reaction curves of PVE-REAL and PVE-RED.
- Return the tuning constant to its correct value.
- Repeat the above operations with different values for each tuning constant. Avoid experimenting with more than one tuning constant at any one time in order not to confuse the results.

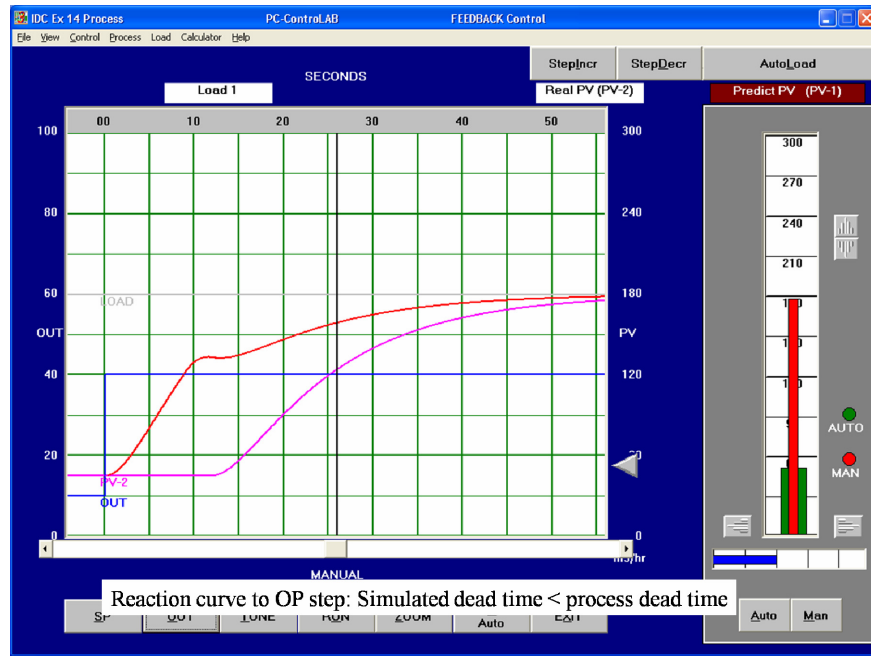
## D16.5 Observation

The reaction curve of PVE-PRED changes as follows when the tuning constants are changed:

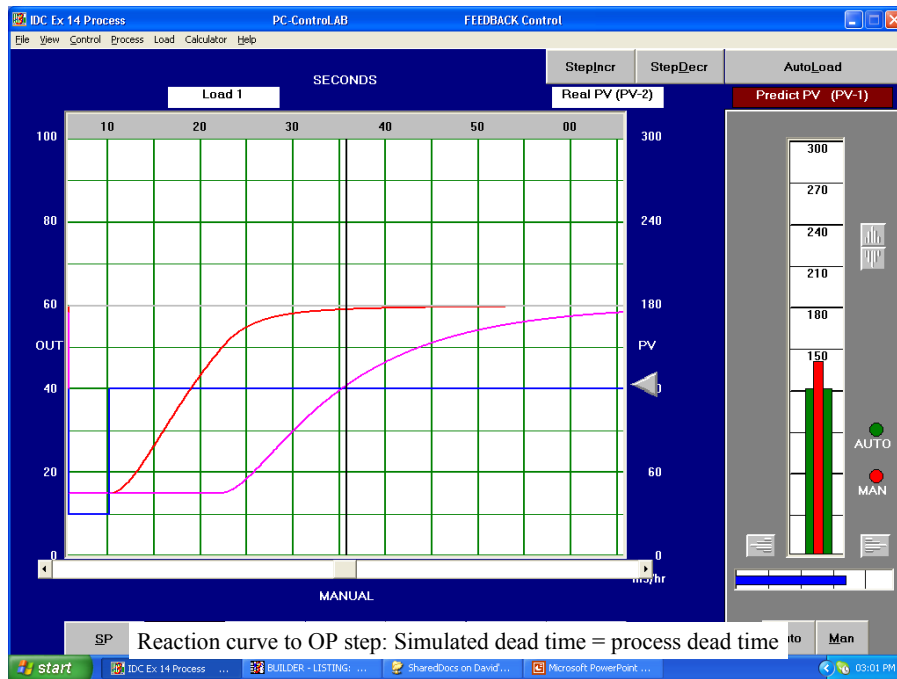
- The change of K-SIM changes the magnitude of the simulated response.
- The change of Simlag 1 and Simlag 2 and change the dynamic of the simulation (form of reaction curve).
- The change of Simdead time causes significant deformations of the reaction curve of PVE-PRED. The characteristic of these deformations indicate whether the dead-time parameter is too long or too short. Compare the reaction curves you find with the ones shown in Figures D16.3, D16.4 and D16.5



**Figure D16.3**  
Dead-time tuning: Model dead time > process dead time



**Figure D16.4**  
 Dead-time tuning: Model dead time < process dead time



**Figure D16.5**  
 Dead-time tuning: Model dead time = process dead time

All of the response shown in the figures were obtained with the simulation gain K-SIM set to 1.6. If you increase or decrease the gain value you will see that overshoots and undershoots easily occur. The difficulty with this type of model is that the effects are highly interactive and some time and experience are needed to separate the variables and find the best fit.

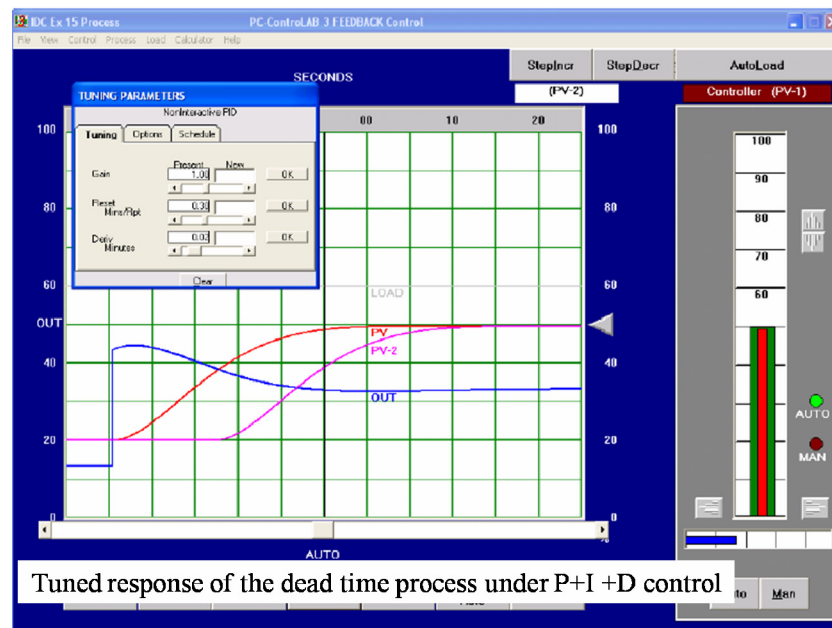
## D16.6 Conclusion

K-SIM, Simlag 1 and Simlag 2 shape the simulated reaction curve of PVE-PRED. Simdead time serves to match the simulated dead-time with the process dead-time. Interactions between the effects make it tricky to find the best fit but usually the dead-time is easy to measure so this parameter can be set first.

## D16.7 Operation of controller tuning

Once you have established a reasonably smooth first or second order lag response from the predicted PV this variable can be used to attempt the tuning of a basic PID feedback control loop. In practice this procedure may use the conventional Zeigler –Nichols approach since you now have a typical step response. However the loop gain values will need to be kept low to ensure stability so start with a very low loop gain and a very slow reset time.

The values you should obtain will be sufficient to provide good loop stability and you will find that integral action can be raised in small steps until the onset of oscillation. The results of your tuning efforts should be something similar to the results shown in Figure D16.6 below.



**Figure D15.6**  
*Dead-time tuning: Response to step change in SP under P+I control*

**Note of caution:** The tuned response shown in Figure D15.6 was obtained when using the PC-Control Lab programme running in scroll rate : Seconds. Please note that due to the short time constants involved in the model the simulation will not be stable in the “Minutes” scroll mode.

## D16.8 Conclusion

The basic idea of Dead-time compensation is straight forward. Nevertheless, it is important to realize that it is a strategy of intermeshed loops. The control loop using the predicted PVE-PRED is intermeshed with remnants of the real PVE-REAL via the error calculation. In addition, one has to realize that the reaction of control actions is predicted only. No prediction whatsoever can be made about the impact of disturbances. Derivative control is most effective in counteracting integral and lagging behavior, but has its limitations if applied to dead-time problems. Therefore, one should take care when adding derivative action that noise does not present added problems.



---

# Exercise 17

---

## Exercise in gain scheduling for control of non-linear processes

**This exercise has been kindly supplied to IDC Technologies by Wade Associates**

**OBJECTIVE:** To demonstrate the benefits of scheduled tuning for non-linear processes.

**PREREQUISITES:** Completion of Exercises

- 9 PID Tuning from Open Loop Tests
- 11 Improving “As Found” Tuning

**BACKGROUND:** If a process is highly non-linear and is operated over a wide operating region, then either the tuning parameters must be changed for different operating conditions, or else the controller will be tuned for the “worst case” condition, resulting in less than optimum tuning at other conditions. One means of coping with this is to issue instructions to the operator to change tuning parameters for different operating conditions. Another possibility is to develop a set of tuning parameters for each operating condition and employ some mechanism which applies the correct tuning parameter set for each condition. This is known as “scheduled tuning” (also sometimes called “adaptive gain”). A basic premise is that the operating condition can be indexed by a single variable, such as the measured process variable, the set point, controller output, or a disturbance variable to the loop. This variable can then become the key to finite regions of the operating zone, each with its own set of tuning parameters.

A slight variation of the above is to assign nominal parameters, then apply a unique multiplying factor to each of the parameters for each defined region. This is the approach used by PC-ControlLAB.

## 1. RUNNING THE PROGRAM

Start **Windows** and Start **PC-ControLAB**

### 2.1 Preparatory

Confirm that you are running the Feedback control strategy.

Select **Process** | **Select Model/ PC Pracs/EX-17/** Highlight “Temp.mdl” and press **Open**.

### 2.2 Tuning for Base Conditions

Notice that the PV scale is 0 to 100 DegC, that the set point is at 60 DegC, and that the load variable (process flow rate) is 0 to 100 m<sup>3</sup>/hr. For this exercise, we will assume that the base conditions (normal operating point) is close to the current set point value of 60 DegC with a load variable of 75% of full scale. The load variable can vary, however, between 50 and 100% of its full range.

We’re going to be making fairly small set point changes, so press **Zoom** and change the vertical scale of the PV from 0 – 100 to 55 – 65 DegC. (*Why small SP changes? To avoid having the controller output hit a limit when we are working at high loads.*)

With the controller in Manual, change the controller output from 48% to 53% (a 5% change). Use the open loop testing method (Laboratory Exercise 9) to determine an initial set of tuning parameters for a PI controller.

Enter these parameters. Put the controller in AUTO. Fine tune these parameters if necessary, using the “Improving As Found Tuning” method (Laboratory Exercise 11). Make your set point changes between 60 and 62 DegC.

When you have determined satisfactory tuning parameters for this condition (load variable = 75m<sup>3</sup>/hr), record them below:

Process Flow: 75 m<sup>3</sup>/hr.      Gain: \_\_\_\_\_      Reset: \_\_\_\_\_ min/rpt.

Now make load changes of 5% (Press **StepDecr** once. After the PV settles out, press **StepIncr** once.) and observe the response.

Return the Set Point to 60 DegC.

## 4. Other Operating Points

Change the process load to 95% of full scale. (Press **StepIncr** 4 times).

Change the SP to 62 DegC, then to 60 DegC. Observe the responses.

Also make load changes of 5% (press **StepDecr** once. After the PV settles out, press **StepIncr** once.) Observe the responses.

How do these responses (to SP and load changes) compare with what you observed when the load was at 75%?

---

Change the process load to 55% of full scale. (Press **StepDecr** 8 times).

Change the SP to 62 DegC, then to 60 DegC. Observe the responses.

Also make load changes of 5% (Press **StepDecr** once. After the PV settles out, press **StepIncr** once.) Observe the responses.

How do these responses (to SP and load changes) compare with what you observed when the load was at 75%?

*What you should have observed so far: Starting with the controller tuned acceptably when the load is 75%, a significant load increase with the same tuning parameters causes the control loop to be slightly more sluggish. A significant load decrease with the same tuning parameters causes the loop to be too aggressive.*

### Tuning for Other Operating Points

While we are at a load of 55%, adjust either the Gain or the Reset, or both, until you get approximately the same type of response for a SP change from 60 to 62 as you did when the load was 75%. Record your tuning parameters.

Process Flow: 55 m<sup>3</sup>/hr.      Gain: \_\_\_\_\_      Reset: \_\_\_\_\_ min/rpt.

Now go to 95% load and adjust the tuning parameters to get approximately the same type of response as you did when the load was 75%.

Process Flow: 95 m<sup>3</sup>/hr.      Gain: \_\_\_\_\_      Reset: \_\_\_\_\_ min/rpt.

## 5. Calculation of Tuning Schedule

Suppose we divide the operating zone into 3 regions, based upon the load value.

- Region 1      Load < 65%. (Actually, we never use a load less than 50%.)
- Region 2      Load between 65% and 85%.
- Region 3      Load ≥ 85%.

Since we said that in our base case (See Section 2.2) the load was 75%, then Region 2 is our base operating region. We will enter tuning parameters for that region as our nominal parameters, and adjust the parameters for the other regions by multiplicative factor. (The multiplicative factor for Region 2 will be 1.00.)

Calculate the Gain multiplier and Reset (Min/Rpt) multiplier for Region 1 and Region 3.

$$\text{Region 1 Gain Multiplier} = \frac{\text{Gain determined at 55\% load (Section 2.4)}}{\text{Gain determined at 75\% load (Section 2.2)}} = \underline{\hspace{2cm}}$$

$$\text{Region 1 Reset Multiplier} = \frac{\text{Reset determined at 55\% load (Section 2.4)}}{\text{Reset determined at 75\% load (Section 2.2)}} = \underline{\hspace{2cm}}$$

$$\text{Region 3 Gain Multiplier} = \frac{\text{Gain determined at 95\% load (Section 2.4)}}{\text{Gain determined at 75\% load (Section 2.2)}} = \underline{\hspace{2cm}}$$

$$\text{Region 3 Reset Multiplier} = \frac{\text{Reset determined at 95\% load (Section 2.4)}}{\text{Reset determined at 75\% load (Section 2.2)}} = \underline{\hspace{2cm}}$$

Enter these values into the table below. (Note that the Deriv multipliers can be left at the default value of 1.0, since the base Deriv value is 0.0. Also note that Region 2 represents our base case, so those multipliers are 1.0.)

Region	Low	High	Gain Multiplier	Min/Rpt Multiplier	Deriv Multiplier
1	0	65			1.0
2	65	85	1.0	1.0	1.0
3	85	100			1.0

Press **Tune** | **Schedule** tab.

Enter, or confirm, that the Number of Regions is set at 3.

Select PV-4 for the Key variable. (This process model has PV-4 connected directly to Load-1, which represents the process flow disturbance.)

Fix up the table to resemble columns 2 - 6 listed above.

Select Scheduled Tuning **ON**.

Return to the **Tuning** tab. Re-enter the tuning values you determined for the base case (Section 2.2).

Press **Clear** to remove the Tuning dialog box.

## 6. Testing of Scheduled Tuning

Change the load to 55%.

For each load value between 55% and 95%, make set point changes to 62, then to 60. Observe the response. Then increment (press **StepIncr**) the load to the next level and repeat.

Ideally, you should see a fairly satisfactory response at each load level. If not, consider what changes you might make to the schedule. Consider changing one or more of the following:

- Number of regions
- Break points between regions
- Multiplying factors.

Record your final tuning schedule in the table below:

Number of Regions: \_\_\_\_\_

Region	Low	High	Gain Multiplier	Min/Rpt Multiplier	Deriv Multiplier
1	0				1.0
2			1.0	1.0	1.0
3					1.0
4					1.0
5					1.0





*Answers to practical No 8*

**ANSWER KEY - LABORATORY EXERCISE 1  
PROCESS DYNAMIC CHARACTERISTICS**

2. FIRST ORDER LAG PLUS DEAD TIME PROCESS

Present PV:	52.5
Present Controller Output:	35.0
Key in a new output value of 45.0.	
Final value of PV:	67.5
What type of process response does this appear to be?	FOLPDT
How much did the PV change?	15%
How much did you change the controller output?	10%
Process Gain:	1.5
Dead Time (Td)	8 minutes
Calculate 63.2% of PV change	9.48
Actual value of PV at 63.2% of change:	61.98
Time Constant	5 minutes
Suppose we had used the 2/3 point, rather than 63.2%, to estimate the time constant.	
2/3 of PV change	10
Actual value of PV at 2/3 of change:	62.5
Time Constant (approx)	5.5 minutes
Percent error:	10%

Click on Process | Change Parameters. Observe the values listed for Dead Time and Time Constant.

Do these parameter values agree with what you observed? Yes

Select "Dead Time" and change its value to 2.0 (minutes).

Select "Process Gain" and change its value to 1.0.

Select "Time Constant" and change its value to 3.0 minutes.

Press CLEAR

Change the controller output from 45.0 to 35.0. Observe the response.

Is this what you would expect? Yes

3. UNKNOWN PROCESS

Click on Process | Select Model.

Highlight "Generic" and press Open.

Upper end of scale (corresponds to high end of transmitter range) 500 DegF

Low end of scale (corresponds to low end of transmitter range) 0 DegF

Span of PV 500 DegF

Initial value of PV (in engineering units) 275 DegF

Initial value of PV (in percent of span) 55%

Controller Output: 35%

Change the controller output from 35.0 to 45.0.

When the PV reaches (apparent) equilibrium, [PAUSE] the simulation. Does this look like a true first order lag plus dead time process? No

Does it look "approximately" like a first order lag plus dead time process? Yes

What is the final value of the PV? 350 DegF

Answer Key 1	Page 3	Process Dynamic Characteristics
How much did the PV change, in engineering units		75 DegF
How much did the PV change, in percent of span		15%
Estimate the process gain.		1.5
Apparent dead time:		2.0 minutes
Apparent time constant:		7.5 minutes
4. OTHER FORMS OF STEP RESPONSE		
4.1 Negative Process Gain		
Initial PV:		46 DegF 46%
Initial Controller Output:		70%
Final PV:		40.2 DegF 40.2%
Final Controller Output		80%
Process Gain:		-0.58
4.2 Integrating Process		
What does the tank level do?		Goes up
Reach equilibrium?		No
Can you eventually stabilize?		Almost
4.3 Inverse Response		
What does the drum level initially do?		Goes up
What does the drum level eventually do?		Goes down

5.0 FREQUENCY RESPONSE

PV Amplitude, Peak-to-Peak:	12.7%
Amplitude Ratio	1.27
Time Lag	8.0 minutes
Phase Shift	48° Lag

Period (Minutes)	Freq (cyc/min)	Cont Output Ampli	PV Ampli	Ampli Ratio	Time Lag (mins)	Phase Shift (Deg)
60	0.0167	10.0	12.7%	1.27	8.0	48
30	0.0333	10.0	8.65%	0.865	7.5	90
15	0.0667	20.0	7.3%	0.365	5.5	132
7.5	0.1333	20.0	2.44%	0.122	3.75	180
3.75	0.2667	20.0	0.54%	0.027	2.25	216

