MATRIX | PROCESS CONTROL

Process Control System

MATRIX

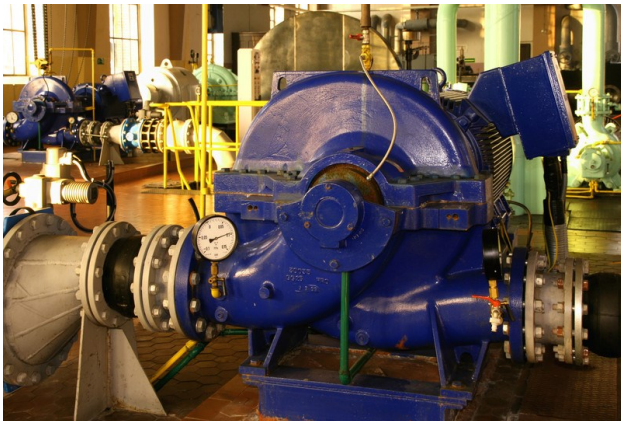**CP3720**

Contents

Worksheet 1

Understanding the drive

In Process Control pumps for water, oil, food-stuffs, air and other materials are at the heart of every system. Understanding how to control driving devices is an important part of understanding how to understand how to control processes.

This photograph shows a pump in a water pumping station.

	Flow	Level	Temperature	Pressure	Servo Pendulum
Drive system	Water pump	Water pump	Heater	Air pump	DC motor
Loss actuator	Variable valve	Variable valve	Baffle	Digital valve with manual adjuster	Bearings
Loss method	Restriction	Fluid leakage	Heat loss	Air leakage	Friction

Over to you:

1) Make sure that you have read the Process Control Reference Manual and understood how your system works, have installed drivers, software etc.

2) For the system you are using open the Manual application. E.g.: 'Open ManualControl.exe'.

3) Press the 'GO' button in the top left hand corner. If your unit is connected properly and the drivers are installed then the 'APPLICATION', 'COMMUNICATIONS' lights should illuminate.

4) If you are using the Flow or level apparatus you will need to calibrate and zero the sensor.

5) On all the systems there is a loss adjustment mechanism: e.g. a valve. Adjust this so that there is no LOAD.

6) Slide the DRIVE slider so that it is off - at position 0. Press the RUN button. Your system is now running but the drive is off.

7) Move the drive slider up to 50% and monitor what is happening in your system. Try different % drives.

8) Adjust the LOAD so that there is some load in the system and the system is stable. With Level and Flow load is regulated by a mechanical valve with many settings. With Temperature this is a fan and a baffle that you adjust manually. With Pressure this is an on/off valve with an adjustable air loss aperture.

9) With the LOAD at this given level, vary the DRIVE output upwards between 0 and 100% in 10% steps. Let the system settle and make a note of the values in the values in the table and graph on the next page.

Worksheet 1

Understanding the drive

Flow		Level		Temperature		Pressure	
LOAD %:		**LOAD %:**		**LOAD %:**		**LOAD %:**	
Drive output	**Manual reading**	**Drive output**	**Manual reading**	**Drive output**	**Manual reading**	**Drive output**	**Manual reading**
10		10		10		10	
20		20		20		20	
30		30		30		30	
40		40		40		40	
50		50		50		50	
60		60		60		60	
70		70		70		70	
80		80		80		80	
90		90		90		90	
100		100		100		100	

Flow

Drive (%) — 0 to 100
l/h — 0 to 250

Level

Drive (%) — 0 to 100
mm — 0 to 200

Temp.

Drive (%) — 0 to 100
C — 0 to 100

Pressure

Drive (%) — 0 to 100
PSI — 0 to 20

Worksheet 1

Understanding the drive

Servo pendulum - speed

LOAD: no weights

Drive output %	Manual reading rpm
5	
10	
15	
20	
25	

Servo speed
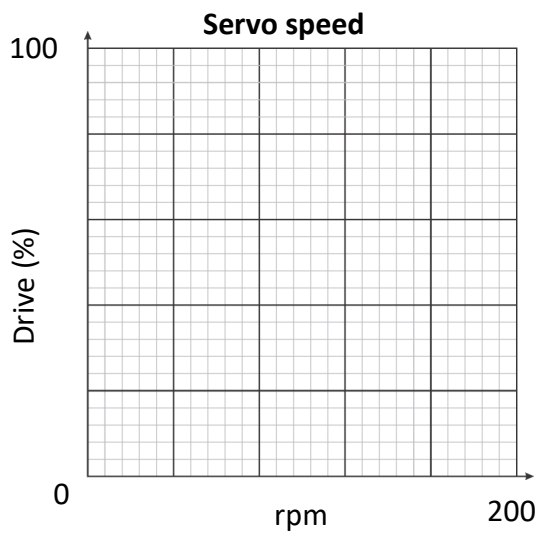
Worksheet 1

Understanding the drive

So what?

Range

These experiments show you that a system with a given load has a certain RANGE. The output does not go down to 0 or up to infinity. It has a minimum rate that can be achieved and a maximum rate that can be achieved.

The range is determined by several things:

- The capability of the drive: the power of the pump, or heater
- The natural load in the system: friction in pipes, heat load
- The purposeful load in a system - the leakage, the cooling effect of the fan. You choose this load.

Drive

The system will have a certain power. How much power is important for the overall performance of the system. However for the purposes of the system equations the thing that is really important is the percentage of drive in the system: from 0% power to 100% power. This is specified as a percentage.

Output scaling factor

Each system has a maximum system value. The maximum value of the drive is 100%. The Output scaling factor for each system is defined as the ratio of max drive to maximum system value. The maximum system value for Pressure is the point at which the pump can supply no more air in the system: around 20 psi. For temperature this is around 100C: the max we want the temperature to go up to. For level this is 200mm - the level at which the overflow pipe takes water out of the cylinder. For flow this is around 250l/h - the maximum achievable. For Servo speed the maximum speed is around 200rpm at 25% drive. For open loop systems the output scaling factor is important. For closed loop systems it becomes much less important.

Over to you:

For each system calculate the Output Scaling Factor.

Flow: Temperature:

OSF:_____   OSF:_____

Level: Pressure:

OSF:_____   OSF:_____

Servo pendulum - speed

OSF:_____   For each system experiment with the values of load so
   (use 25% drive value)   that you understand how each system works.

Worksheet 2

Understanding the sensors

Sensor outputs need to be converted by computers to values that make sense in our control equation. For some systems this is taken care of in the sensor itself. Other systems need calibration. All systems need checking to make sure that the sensor value is correct.

This photograph shows a typical flow rate sensor like the one in the Flow rig.

Over to you:

1) Make sure that you have read the Process Control Reference Manual section on Sensors and their operation. This tells you how each sensor is calibrated.

2) Open the Manual application by running 'Open ManualControl.exe'.

3) Press the GO button in the top left hand corner.

4) Adjust the LOAD so that there is some load in the system. With Pressure, Level and Flow this is a mechanical valve. With Temperature this is a fan and a baffle that you adjust manually. The load for the Servo pendulum is adding weights.

5) The procedure for zeroing and calibrating the sensor for each Process Control system us slightly different. Follow the relevant one to make sure that your system is calibrated.

Flow:

1) The Input Scaling factor can be calculated from the datasheet as shown in the Reference document.

2) For several values of drive make a note of the software reading and the Rotameter sensor reading. Making a graph and measuring the slope to determine the Calibration factor you need.

3) Adjust the Calibration factor property in the software and verify that the software now gives the same reading as the Rotameter.

Level:

1) Open the valve to empty the level tank.

2) Start with a low input scaling factor - say 0.001 and enter this in the Properties.

3) Open the valve until the gauge reads 0.

4) Click Zero on the software.

5) For any value of pressure read the level manually on the gauge and read the value from the software. Enter a calibration factor in the software properties and repeat the process to verify that this is accurate. The Sensor has a

Worksheet 2

Understanding the sensors

24bit (0 to 16 million) so the numbers here will be very small.

Temperature:

1) The input scaling factor here is 1. Enter this value in the software properties.
2) Use the manual thermocouple and meter to verify that this is accurate.

Pressure:

1) Start with a low input scaling factor - say 0.001.
2) Open the valves until the gauge reads 0.
3) Click Zero on the software.
4) For any value of pressure read the pressure manually on the gauge and read the value from the software. Adjust the calibration factor in the software properties by the ratio of gauge reading to software pressure reading. The Sensor has a 24bit (0 to 16 million) so the numbers here will be very small.
5) Repeat the process to verify that the pressure reading is accurate.

Servo Pendulum:

The encoder has scaling and calibration factors of 1:1 because it is a digital system that gives the reading in degrees.

Over to you:

Note the values of the sensor Scaling Factor and Calibration Factor if relevant.

Flow:

SF: _____

CF: _____

Level:

SF: _____

CF:: _____

Temperature

SF: _____

Pressure

SF: _____

CF: _____

For each system experiment with the values of load so that you understand how each system works.

Worksheet 3
ON/OFF control systems

Some control systems can make effective use of simple On/Off control techniques. Here we evaluate the technique for the Flow system.

The photograph shows valves and pumps in a gas distribution system.

Over to you:

1) Load the ON/OFF application. E.g.: 'Open OnOffControl.exe'. This is an application that allows you to see the effects of simple on/off feedback control on the system.
2) Press the GO button in the top left hand corner.
3) Set the LOAD to a certain value. Select a Set point value.
4) Press RUN and monitor the output.
5) Select other set points and losses and see how the output changes.
6) Make some notes and compare how successful it is to use this technique to operate the different systems.
7) Understand the Hysterisis property and how it changes the output.

So what?

You should have made some observations:

- It is possible to use this technique to roughly control the process.
- The output is not in any way 'smooth' or accurate.

There is an error between the Set Point and the Process Value. This error is positive

- for low values and negative for high values.
- The ON/OFF frequency can change with Set point.

Conclusion

To a point this works - it allows some adjustments of value. But it is not satisfactory and the operational life of drive systems may be reduced by operating them in this way.

Notes on the Servo pendulum

The motor here is powerful and the fact that the data is sampled at discrete intervals makes the true information hard to see. To see the ON/OFF effects properly you may need to turn the control frequency down to 20Hz.

Worksheet 4
System time constant

When designing a digital PID system we need to set some basic parameters around the timings of measurements in the system. This depends on the response time of the system itself.

Photograph shows a pipe-fed hydroelectric plant

Over to you:

1) For the system you are using open the Manual application. E.g.: 'Open ONOffControl.exe'.
2) Press the GO button in the top left hand corner.
3) Adjust the LOSS so that there is a no loss in the system.
4) Set the DRIVE to around 50% of the scale.
5) Run the application and stop it as soon as the output reaches the set point.
6) Zoom in on the graph and measure the time taken for the output to rise from its starting position to 63% of its set point.

So what?

The time constant of a system is the time taken to get from 0% to 63% of the final output step change.

For the Flow system this is very fast: the pump is powerful and water is reasonably incompressible.

For the temperature system this is quite slow.

For practical digital control systems the interval between samples should be around a tenth of the time constant.

Measure the time constant and calculate the ideal sampling frequencies for each system:

Flow: Level:

_____   _____

Temperature   Pressure

_____   _____

The sampling time in your system can make a big difference in how your system operates.

Worksheet 4
System time constant

Measuring time constant of Servo pendulum system

The motor on the system s very powerful and the inertial in the system is relatively small. This means that the time constant for this system is very small.

The time constant with 6 weights on the disc will be very different for no weights.

To measure the time constant here:

Over to you:

1) Load the ON/OFF application. E.g.: 'Open OnOffControl.exe'. This is an application that allows you to see the effects of simple on/off feedback control on the system.

2) Press the GO button in the top left hand corner.

3) Set the LOAD to a certain value. Select a Set point value.

4) Press RUN and monitor the output.

5) Select other set points and losses and see how the output changes.

6) Zoom in on the graph and measure the time taken for the output to rise from its starting position to 63% of its set point.

So what?

You will find that the system responds very fast. Faster than the software can time accurately. So you will need to make a guess at the time constant.
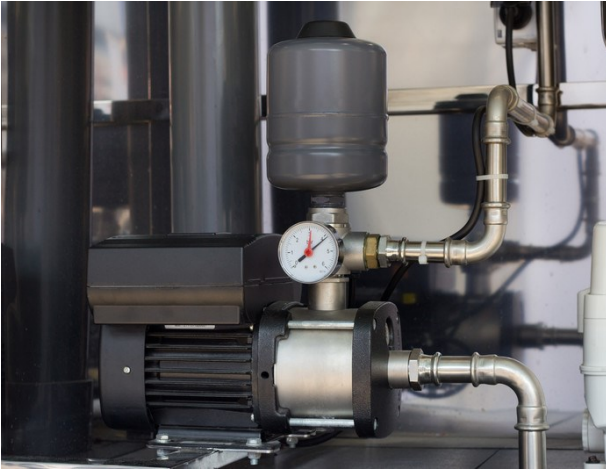
Over to you:

1) Add 6 weights and repeat the process.

2) Measure the rise time and calculate the time constant.

So what?

Even with 6 weights on the system responds fast. You will have to make an estimate as to what the time constant is. Remember that The system is digital and the rise time can be measured from the time the set point rises.

Worksheet 5
P controller

There is a much better way of controlling systems than simple on/off control: proportional control where we vary the power being supplied to the process depending on the difference between the desired output and the actual flow rate. In this worksheet you will explore the creation of a Proportional controller for flow based systems.

Photograph shows a water pump in a factory.

Over to you:

1) Load the PID application. E.g.: 'Open PIDControl.exe'.

2) Refer to the Reference document and make sure you are familiar with how the software works.

3) In the Properties section turn Integral wind up off. Turn Derivative filter off. You will use these later.

4) In the Properties section enter the values for Output scaling factor that you calculated earlier.

5) In the Properties section enter the values for Sensor scaling factor that you measured earlier.

6) In the properties section enter the Calibration factor you measured earlier - if it is needed.

7) In the Properties section enter a Control frequency that is appropriate based on the system time constant.

8) Press the GO button in the top left hand corner and check your computer is communicating with the unit with the Application and Communications LEDs green.

9) Set the load quantity so that there is a loss in the system.

10) This software is a full PID controller. For now we are going to concentrate on the P term. So firstly reduce Ki and Kd to 0 which takes them out of our equation and select a starting value of Kp as 1. On the screen you can see the control equation with the P, I and D terms.

11) Select RUN.

12) Move the set point to around half way up the scale.

13) When the value settles make a note of the Set point and the output value for each system.

So what?

You have just witnessed a simple Proportional controller in action. What the software does is take the Process Value and subtract the Set Point from it to give an Error value. It then applies the Proportional P multiplier to this value and then the Scaling Factor multiplier (and calibration if there is any) to this value and then sets the Pulse Width Modulation drive of the pump to this value. This can be expressed mathematically as:

Error = Set Point - Process Value

Worksheet 5
P controller

Flow		**Level**		**Temperature**		**Pressure**		**Servo speed**	
Set point:		Set point:		Set point:		Set point:		Set point:	
Final output		Final output		Final output		Final output		Final output	

PWM % = Scaling Factor x Proportional factor x Error

Very often you will see this expressed in a different way by engineers:

$$O(t) = K_f \times K_p \times e(t)$$

Where:

O(t) shows PWM as the letter 'O' and the '(t)' implies that this is varying in time.

K_f is the Scaling Factor for the PWM drive

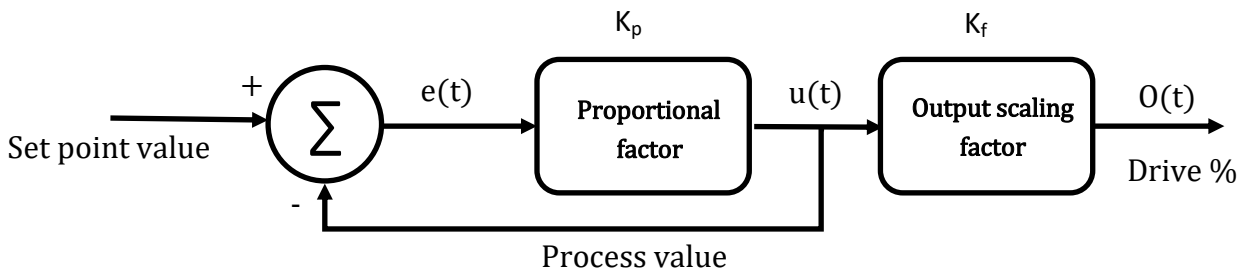K_p is the value of the Proportional multiplier

e(t) is the error and once again the '(t)' implies that this is varying in time.

Worksheet 5
P controller

$$O(t) = K_f \times K_p \times e(t)$$

Control Function

Block Diagram

So what?

The system we have created can be expressed as a mathematical equation. This is sometimes referred to as the Control Function.

This is also sometimes expressed in diagrammatic form. This diagram is shown above and it is known as a Block Diagram.

The Block Diagram shows the flow of signals in a system and it is used by engineers as a shorthand way of explaining the signal flow and the mathematical processes in the system.

In the block diagram the Greek letter Epsilon ($\sum$) in the circle is used to mean that the incoming signals are summed. The '+' and '-' signs denote whether the value or the negative equivalent of the value are summed. The rectangular boxes denote multiplications to the signals.

When discussing control systems we are often as much concerned with the mathematical part of the problem as opposed to the mechanics of

converting the mathematical result into a driving signal and this part of the problem is left out of the mathematics. This means that the system above is most often described mathematically like this:

$$u(t) = K_p \times e(t)$$

The key problem with Proportional only controllers is that the error is never zero: if it was then the output drive would be 0 and the process would stop.

The pseudocode needed to set up a system like this is shown below:

```
PWM_out = Scale_factor x Kp x error
```

Worksheet 5
P controller

Over to you:

You need to get a feel for this system and the various values of Kp and how the value chosen affects the system performance.

1) Adjust the system so that the set point is 50% of full scale.

2) Once the system has settled adjust the set point to 75% of full scale.

3) Then adjust it for 50% again.

4) Flow, Level, Temperature, Presssure: repeat this for values of K_p (Proportional value) of 1, 2, 3, 4 and sketch the results below for each of Temperature, Pressure, Level, Flow and Servo speed.

4) For each of the high set points make a note of the final settled value.
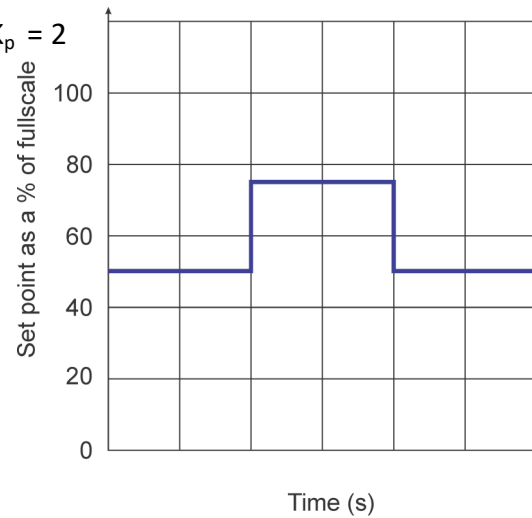
So what?

You should see that there is always an offset in a P only controller. The offset is determined by the value of K_p : the higher the value of K_p, the smaller the error.

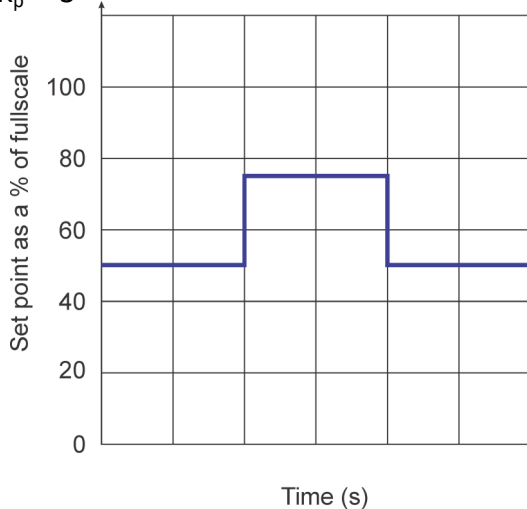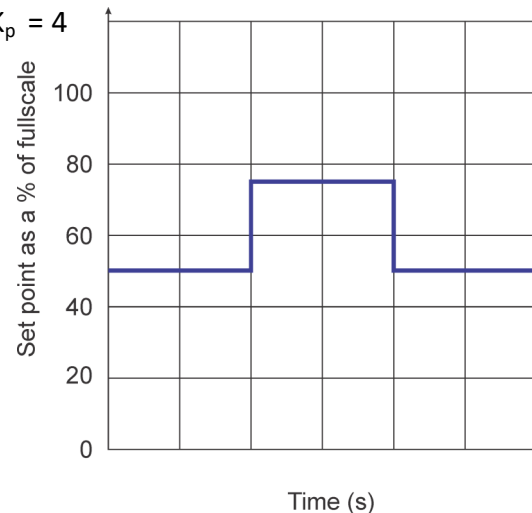Higher values of K_p also give more instability.

$K_p = 1$
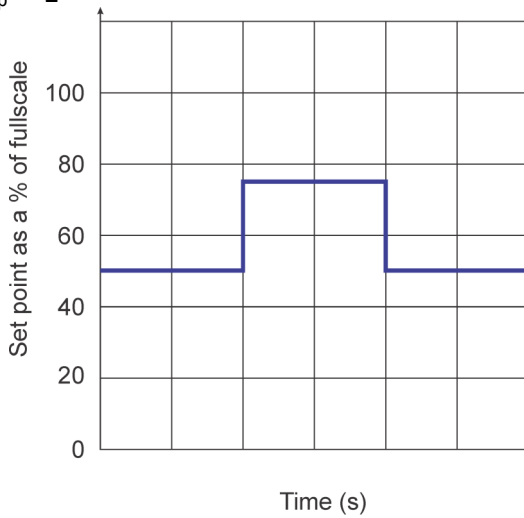
$K_p = 2$

$K_p = 3$

$K_p = 4$

Worksheet 6
PI controller

The problem with the P only controller is that there is always an offset required. This problem can be solved by adding an Integral term that corrects the long term error in output.

Photo shows pumps in a boiler room.

Over to you:

1) Load the PID application: 'Open PIDControl.exe'.
2) Refer to the Reference document and make sure you are familiar with how the software works.
3) In the Properties section turn Integral wind up off. Turn Derivative filter off. You will use these later.
4) In the Properties section enter the values for Output scaling factor that you calculated earlier.
5) In the Properties section enter the values for Sensor scaling factor that you calculated earlier.
6) In the properties section enter the Calibration factor you calculated earlier - if it is needed.
7) In the Properties section enter a Control frequency that is appropriate based on the system time constant.
8) Press the GO button in the top left hand corner and check your computer is communicating with the unit with the Application and Communications LEDs green.
9) Flow, Level, Temperature, Presssure: Alter Kp to 1 and Ki to around 10.
9) Servo speed: Alter Kp to 1 and Ki to around 0.1.
10) Select RUN.
11) As before move the Set point between 50% of full scale and 75% of full scale.
12) For varying values of Kp and Ki make a note of the performance of the system on the small graphs on the next page.
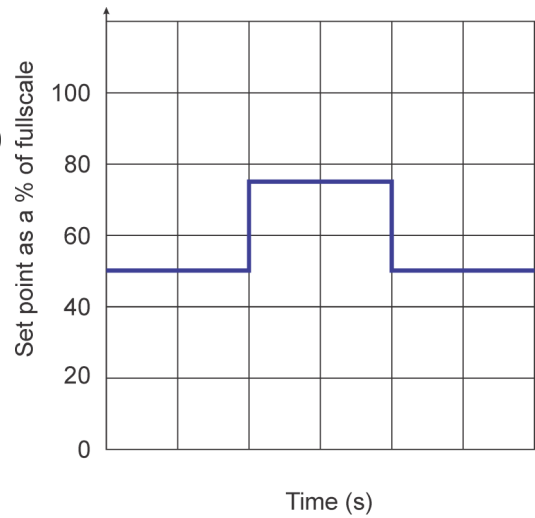
Worksheet 6
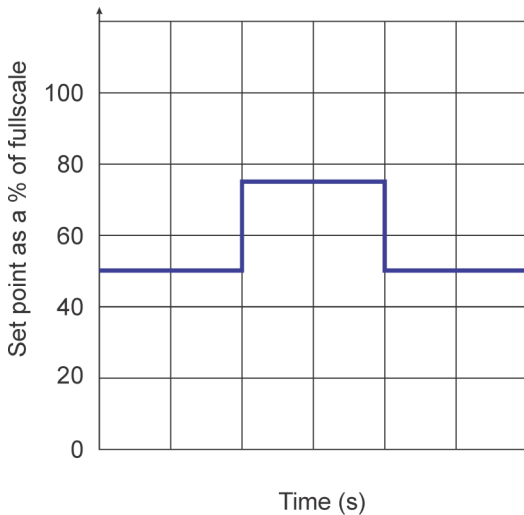## PI controller

FLTP:

K_p = 1

Ki = 20
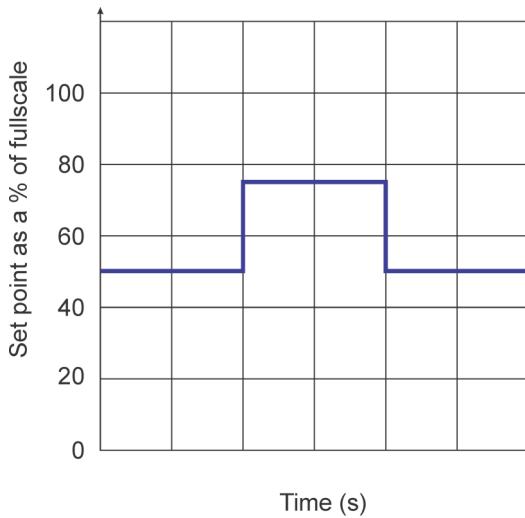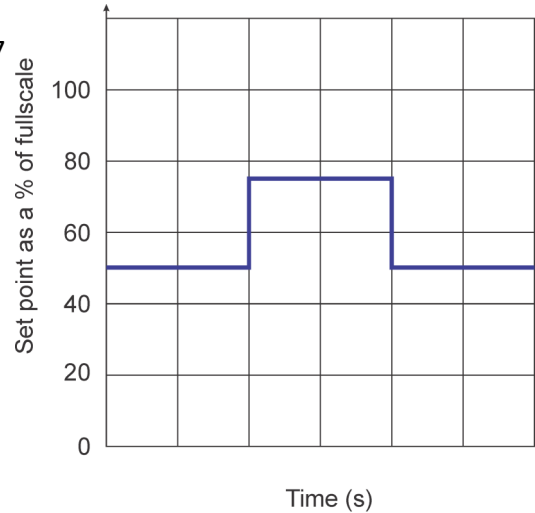
Servo:

K_p = 1

Ki = 0.1

FLTP:

K_p = 1

Ki = 20

Servo:

K_p = 1

Ki = 0.06

FLTP:

K_p = 1

Ki = 30

Servo:

K_p = 1

Ki = 0.04

FLTP:

K_p = 0.7

Ki = 40

Servo:

K_p = 1

Ki = 0.02

So what?

You have just witnessed a simple Proportional Integral (PI) controller in action. This takes the P only controller in the previous worksheet and adds an additional term to the system. The great thing about the PI controller is that it takes the error in the system down to close to zero and hence solves the major problem with the Proportional only controller. The PI controller also takes into account the past history of the system in terms of boosting drive to the pump if the pump is not having the desired effect.

The Integral term applies a correction to the system that is not only proportional to the error but also proportional to the time for which the error has existed: the longer the error has persisted the greater the effect.

In practice this is achieved by an integration: for a given time we add the error multiplied by the time the error has persisted for and multiply this by a new factor: K_i
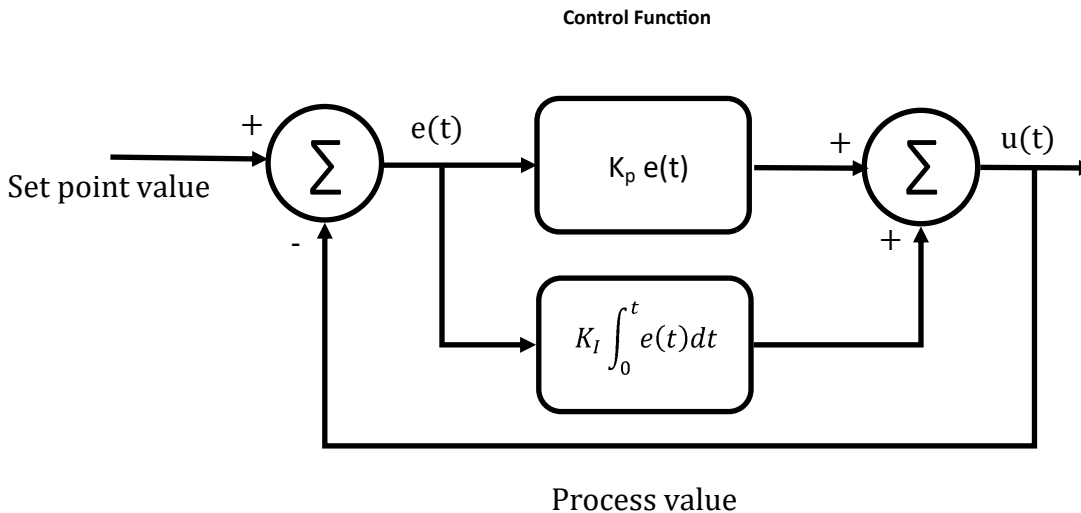
Mathematically this is expressed as:

$$K_I \int_0^t e(t)dt$$

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt$$

Control Function

Set point value

e(t)

K_p e(t)

$K_I \int_0^t e(t)dt$

u(t)

Process value

Block Diagram

Mathematically the overall Control function has changed as you can see above - it now has an Integral term in it.

The block diagram has also changed: the error is fed into two processes - one for the Proportional term and one for the Integral term. They are summed to give an output function which is fed to the PWM drive process. Note that for clarity we are leaving the PWM process out of the block diagram.

Programmatically using pseudocode we can simply develop a program to do this as shown below.

The problem with the PI controller is that there is an overshoot - the output value is more than the setpoint for a certain time. For some systems that's fine - for others it is not.

```
Previous error = Error
Error = Process_value – Set_value
Integral = Integral + error
PWM_out = Scale_factor x (Kp x error + Ki x Integral)
```

Worksheet 7
PID controller

There are situations where we need to control a pump to deliver a required amount of liquid quickly without any overshoot. For example with a prepaid petrol pump. The PI system falls short of requirements here but adding a differential term can help.

Over to you:

1) Load the PID application: 'Open PIDControl.exe'.

2) Refer to the Reference document and make sure you are familiar with how the software works.

3) In the Properties section turn Integral wind up off. Turn Derivative filter off. You will use these later.

4) In the Properties section enter the values for Output scaling factor that you calculated earlier.

5) In the Properties section enter the values for Sensor scaling factor that you calculated earlier.

6) In the Properties section enter a Control frequency that is appropriate based on the system time constant.

7) Press the GO button in the top left hand corner and check your computer is communicating with the unit with the Application and Communications LEDs green.

8) Alter the value of Kp to 1, Ki to 20 and Kd to 0.02.

9) Select RUN.

As before move the Set point between 50%

1) of full scale and 75% of full scale.

2) For varying values of Kp, Ki and Kd make a note of the performance of the system on the small graphs on the next page.

FLTP:

$K_p = 1$

Ki = 20

Kd = 0.02

Servo:

$K_p = 1$

Ki = 0.5

Kd = 1
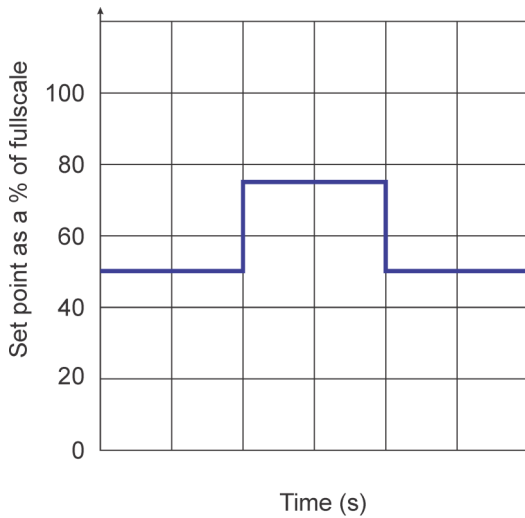
FLTP:

$K_p = 1$

Ki = 20

Kd = 0.04

Servo:

$K_p = 1$

Ki = 0.5

Kd = 2
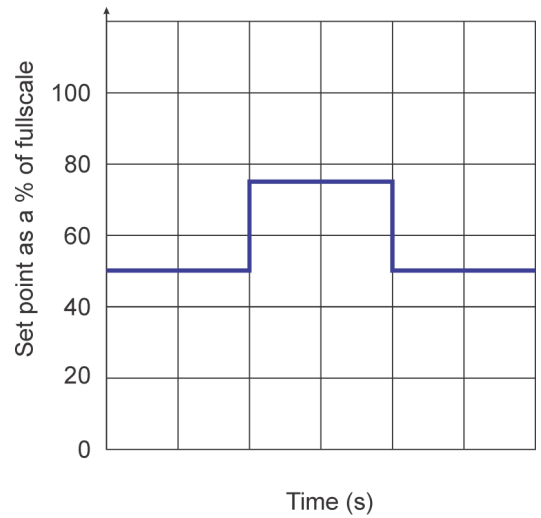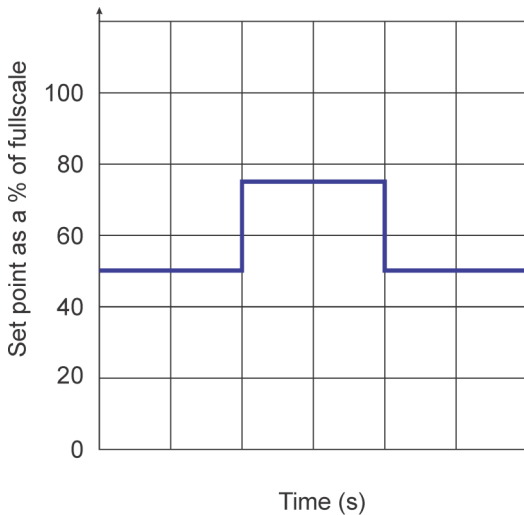

FLTP:

$K_p = 1$

Ki = 20

Kd = 0.06

Servo:

$K_p = 1$

Ki = 0.5

Kd = 3

FLTP:

$K_p = 1$

Ki = 20

Kd = 0.08

Servo:

$K_p = 1$

Ki = 0.5

Kd = 4
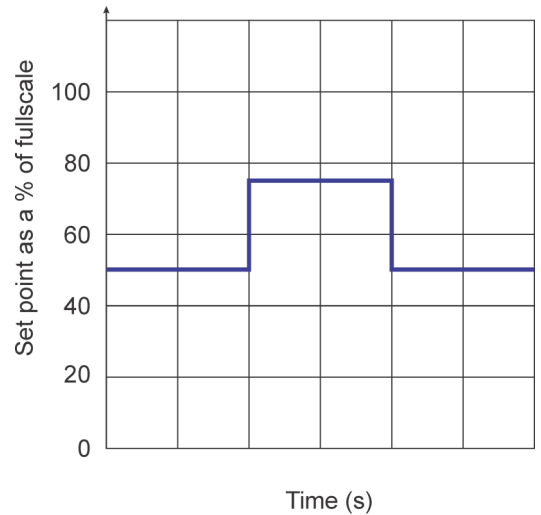
So what?

You have just witnessed a full Proportional Integral Differential (PID) controller in action. This takes the PI controller in the previous worksheet and adds an additional term to the system.

The problem with a PI only controller is that the system does not take into account the rate of change of the system error. If the error is falling fast then we might want to slow the pump up a bit. The Derivative term applies a correction to the system that proportional to the rate of

change of error.

In practice this is achieved by an differentiation term in the equation.

Mathematically this is expressed as:

$$K_D \frac{de(t)}{dt}$$

$$u(t) = K_P e(t) + K_I \int_0^t e(t)dt + K_D \frac{de(t)}{dt}$$

Control Function

Block Diagram

Process value
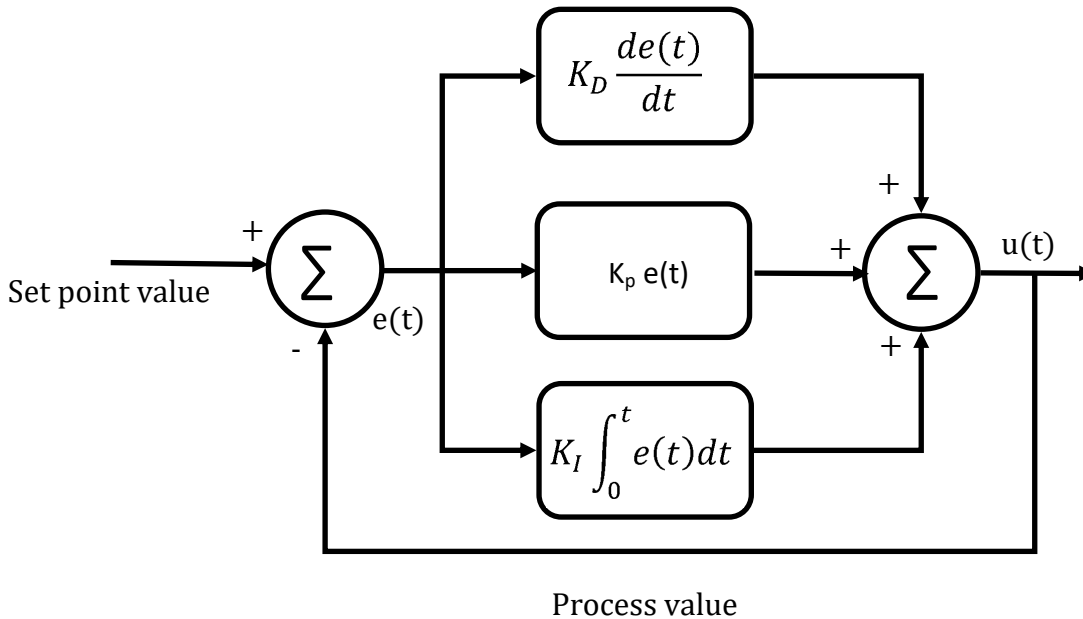
Mathematically the Control function has changed as you can see above - it now has an Derivative term in it.

The block diagram has also changed: the error is fed into two processes - one for the Proportional term and one for the Integral term. They are summed to give an output function which is fed to the PWM drive process.

Programmatically using pseudocode we can simply develop a program to do this as shown below.

So what?
You have now got a feel for the effects of each of the terms in P, I and D. But how do we work out the values we need?

```
Previous error = Error
Error = Process_value – Set_value
Integral = Integral + error
Differential = error - last error

PWM_out = Scale_factor x (Kp x error + Ki x Integral + Kd x Differential)
```

$$U(s) = \left(Kp + \frac{Ki}{s} + Kd.s \right) E(s)$$

The control function can be rewritten as shown above to form a Transfer Function. In the transfer function the integral is written as 1/s and the differential term is written as s. The explanation of this is beyond the scope of these worksheets but there are many references to this on the internet.

In a digital system the terms in s are mapped to digital sampled terms usually represented by a 'Z'. Here successive digital values of the error are written as:

E^Z current error
$E^{(Z-1)}$ previous error 1 sample time ago
$E^{(Z-2)}$ error 2 sample time periods ago
$E^{(Z-3)}$ error 3 sample time periods ago
Etc.

The differential term is rewritten as
   Kd.(current error - previous error)

Or more mathematically as:
   $Kd. (E^Z - E^{(Z-1)})$

The integral term is rewritten as:
   Integral = integral + error

Or more mathematically as:
   $Ki . \sum (E^Z + E^{(Z-1)} + E^{(Z-2)} + E^{(Z-3)} + ...E^{(Z-n)})$

Where the terms are limited to n samples - or the sum of all the errors since switch on.

So that the transfer function can be rewritten as:

$$U(Z) = Kp.E^Z + Ki . \sum (E^Z + E^{(Z-1)} + E^{(Z-2)} + E^{(Z-3)} + ...E^{(Z-n)}) + Kd. (E^Z - E^{(Z-1)})$$

Over to you:
- Check that the coding examples match these new definitions of the integral and differential terms
- Discuss what effect sample time has on the values in a system.

Worksheet 8
Zeigler Nichols algorithm

MATRIX | **PROCESS CONTROL**

Control type	K_P	K_I	K_D
PID classic	0.6 Ku	1.2Ku/Tu	0.075 KuTu
P	0.5Ku		
PI	0.45Ku	0.54 Ku/Tu	
PD	0.8Ku		0.1 KuTu
Some overshoot	Ku/3	(2/3) Ku/Tu	(1/9)Ku/Tu
No over-shoot	0.2Ku	(2/5)Ku/Tu	(1/15) Ku/Tu

So You should now have some idea of how Kp, Ki, and Kd and their values affect a system's performance. But what are the right values? The Zeigler Nichols algorithm helps you define vales for different control systems.

Over to you:

First you need to gather some information about your system. To do this:

1) With the same settings as in the previous experiment set Kp to 1, Ki to 0, Kd to 0.

2) Press the RUN button.

3) Slowly increase P to a point of 'neutral stability': this is defined as a value when the output starts oscillating in a predictable and stable way. This is the critical or ultimate gain and we define this as 'K_U'.

4) Stop the system by clicking on the stop button when you have gathered a number of cycles of 'stable instability'.

5) Click on the graph area. Use the scroll wheel on the mouse to zoom into the graph. Measure the time between the instability peaks. This is the neutral stability period or T_U.

So what?

Ziegler and Nichols were famous engineers in the 1940's who did a lot of research on control systems. They came up with a method of approximating the solution to the problem of selecting values of P, I and D for control systems based on their behaviour in terms of K_U and T_U.

This table is shown below.

Over to you:

Using a spreadsheet and the values of K_U and T_U you measured calculate the values of Kp, Ki, and Kd for the different types of control system shown in the table.

As this is a discrete system you will need to divide the value of Ki by the sample frequency and you will need to multiply the value of Kd by the sample frequency to get the values you need.

Very the Set point between 50% and 75% (lower for the temperature apparatus) of full scale as before and verify that this works for the Flow system for PID classic.

For the servo pendulum you will find that the system is very noisy: it will not work without a derivative filter. Use 6 weights and calculate Zeigler Nichols as above.
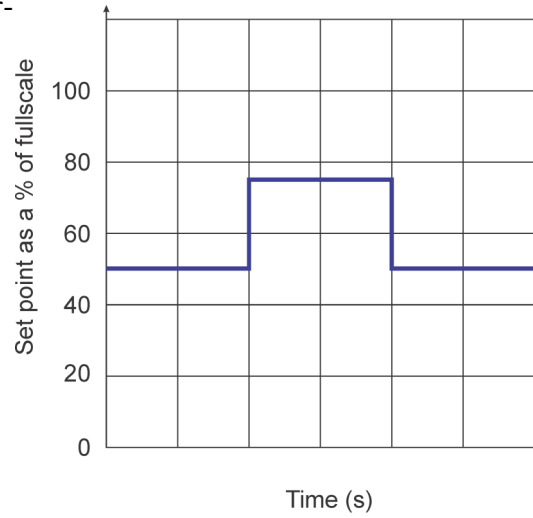
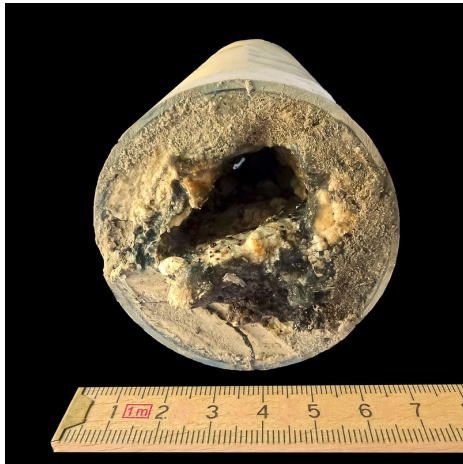# Worksheet 8
Zeigler Nichols algorithm

PID classic
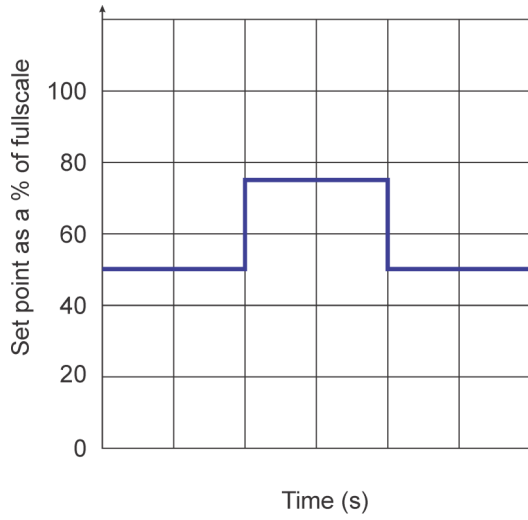
No over-shoot

Worksheet 9
Integral wind up

As you have seen a properly tuned PID control system provides a great solution. If everything in your system runs perfectly then this works really well. In practice there are some 'real world' problems that we sometimes need to take care of before the system will be fit for purpose.

Photograph shows a partially blocked pipe.

Over to you:

1) Set the system up with the PID classic values for Kp, Ki, and Kd that you deduced in the Zeigler Nichols calculations in the previous worksheet.

2) Choose a suitable DRIVE and LOAD for your system.

3) Run the system and notice how the integral term appears on your graph: the yellow line.

4) Stop the system from reaching set point and see how the integral value builds up. To do this:

- Pressure - use the manual valve to leak air at a higher rate.

- Temperature: use a hair dryer to cool the system

- Flow: clamp one of the tubes with your fingers

- Level: clamp the tube with your fingers

- Servo pendulum: stop the movement with your fingers.

So what?

For some systems the Integral term builds up to a point where it significantly increases the time a system takes to settle down. A fault in the system like a blocked pipe or loose wire, can result in a very large accumulated value for the Integral component that can take some time to clear. This is called Integral wind up. Because of this many systems introduce a variation to the controller that limits the value the of the Integral component. This is called Integral clamping as the Integral is clamped to a value we can set.

Over to you:

1) The software has a facility in the properties table to allow you to clamp the value of the Integral. Select 'Integral clamp' to YES .

2) Repeat steps 1 to 3 and compare the results when the system was unclamped. Take screen grabs or export the data to Excel and compare the two sets of information.

3) For the system you are using does the integral clamp produce a better control system?

So what?

The Integral clamp allows system designers to allow for problems in the control process that might affect the performance of the system. If there is a problem in the system and the integral value is allowed to rise to any value then when the problem with the system is fixed the system will take a long time to return to its operational state. By limiting the value the integral term can

assume the system is made more robust.

This necessitates an additional term to our block diagram which you can see below

You can also see how this would be implemented in code.
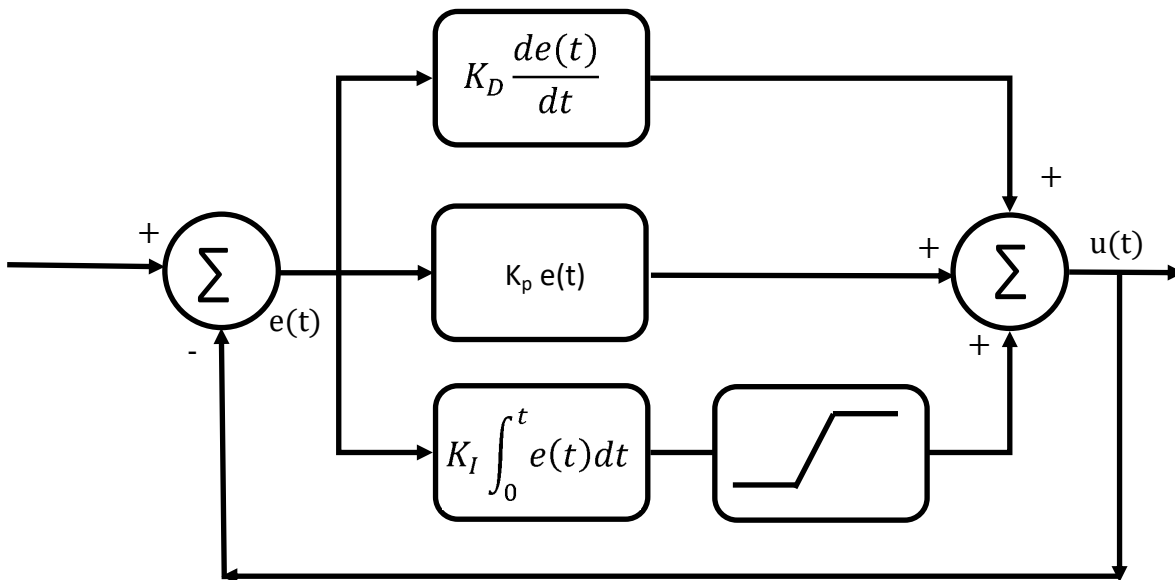

```
Previous error = Error
Error = Process_value – Set_value
Integral = Integral + error
If Integral = Max_Clamp_Value
     Integral = Max_Clamp_Value
EndIf
If Integral Min_Clamp_Value
     Integral = Min_Clamp_Value
EndIf
```

Worksheet 10
Derivative filter

A key problem with some control systems is noise. If there is some noise in the system path then the Differential component can introduce large amounts of feedback which amplifies the noise terms. This is so bad in some systems that PID control is not possible: we need a mechanism to limit the effects of noise.

Photograph shows remote pumping station in Canada.

Over to you:

1) Set the system up with the PID classic values for Kp, Ki, and Kd that you deduced in the Zeigler Nichols calculations in the previous worksheet.

2) Choose a suitable DRIVE and LOAD for your system.

3) Run the system and look at the derivative term on your graph - the blue line.

So what?

Some systems are noisy and some are not. The derivative term is problematic in control systems because it amplifies any noise that is present in the system. Often that noise is down to the nature of sampling in a digital system. Differential terms are usually the most problematic terms in PID systems and this is why a great many systems are simply PI controllers leaving out the differential term altogether.

However with a little adjustment it is possible to make some improvements to the differential.

Over to you:

1) In the properties section adjust the Derivative filter term to YES.

2) Adjust the Derivative Filter term to 10.

2) For the system you are using does the derivative filter produce a better control system?

So what?

A small low pass filter added to the system alters the frequency of the signals that are fed back to the error signal. This makes the system much more stable and cuts down the noise in the system.

You can see how this affects the bock diagram below.

You can see how this is implemented in code below.

```
//DFC is the Derivative Filter Coefficient

output = last_output + DFC (output - last_output)
Differentialterm = Kd * output
Last_output = output
```

Worksheet 11
Manual tuning

Most of the time the Zeigler Nichols 'rule of thumb' calculations can give us good approximations to getting a stable system using P, I and D control. ZN also includes values for 'no overshoot'. Inpractice these do not always work for control systems.

Over to you:

Calculate the Ziegler Nichols values for 'no overshoot' and enter them into the system.

On some systems the ZN values for 'no overshoot' don't produce the results we want.
Manually adjust the values of Kp, Ki, and Kd to give a typical no overshoot result.

Make a note of your results. Describe how increasing and decreasing values of Kp, Ki, and Kd affect the system performance.

Interfacing with MATLAB/LabView

The worksheets above make use of off the shelf software that is controls the system. This is fine for simple systems. In practice if you are a design engineer you may need to design control software from scratch understanding the transfer function of the control system.

Photograph shows a design engineer in a lab..
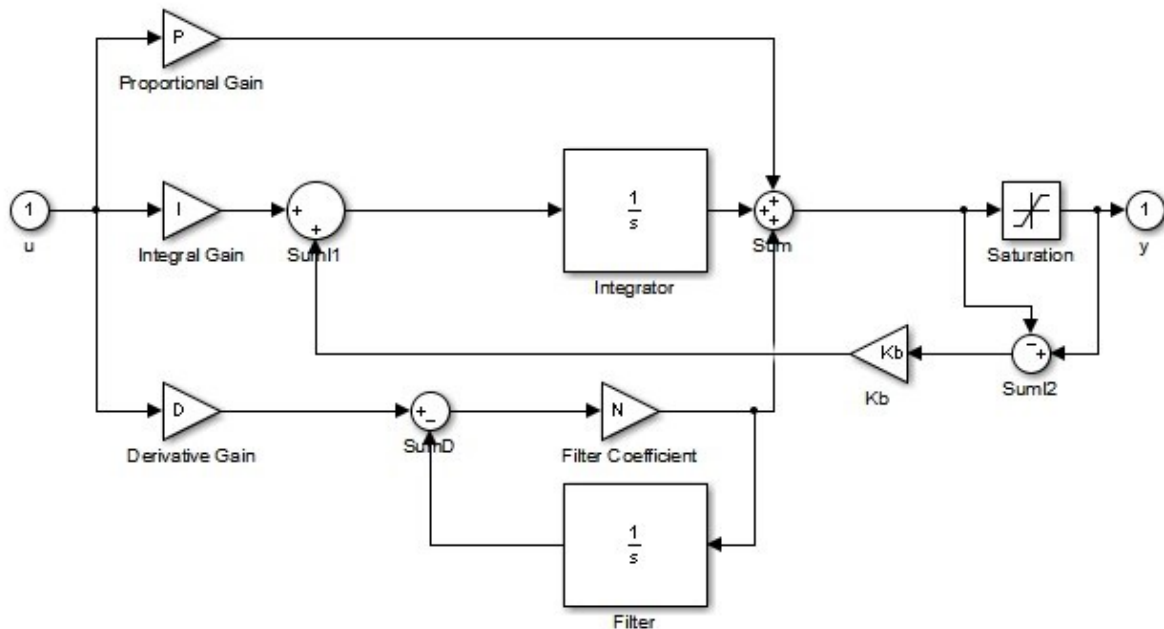
1) Familiarise yourself with the contents of the Process control reference guidebook.
2) Familiarise yourself with the list of API commands in the reference guidebook.
3) Construct a Matlab/Labview program that simply turns the drive on and reads the feedback value from the sensor.
4) Read worksheets 1 to 6. Replicate these functions in Matlab / Labview so that you understand the scaling factor of the pump, the calibration of the flow sensor and the operation of the Valve.
5) Using your knowledge of Laplace construct an S plane model for the PID system described by the block diagram in Worksheet 9.
6) Modify your program to implement integral clamping and derivative filtering.

Worksheet 13
Servo pendulum - angle control

With a standard H drive it is possible to use 4 FET transistors as switches to control the direction of current in a motor. This configuration is very different to the temperature, pressure, Flow and Level systems which have a motor or pump that only drives one way. In this case Ziegler Nichols does not apply and you will need to manually tune the system based on your understanding of the effects of Kp, Ki, and Kd.

By adding a single weight to the rotor disc you can create a pendulum. In this case we are interested in controlling the angle of the rotor: the problem is similar to a self balancing personal transporter like the one in the photograph above.

Over to you:

1) Adjust the system so that there is only 1 weight at 0 degrees.

2) Load the PID application: 'Open PIDControl.exe'.

3) In the Properties section turn Integral wind up off. Turn Derivative filter off.

4) In the Properties section enter the values for Output scaling factor that you calculated earlier.

5) Set Kp = 1, Ki = 0, Kd = 0.

6) In the Properties section enter a Control frequency of 1000Hz. Press the GO button in the top left hand corner.

7) Hold the disc so that the weight is at top dead centre and click on the ZERO button.

8) Press RUN

9) Experiment with the values of Kp, Ki, and Kd to see how they affect the system. For a typical personal transporter as you can see in the photograph what settings would you choose and why?

10) Is the term in Ki needed?

Instructor Guide

Instructor guide

About this course

Introduction

The course is essentially a practical one. Where possible, practical implications and applications of the theory are highlighted to make the course more relevant to the students.

The equipment offers tremendous scope for additional research on the topic and instructors may wish to encourage and support this research.

Aim

The course introduces students to concepts and devices used in a range of Process Control systems. It is ideal for the teaching of Unit 45 of the BTEC Higher National: Industrial systems and is usable in any course that is concerned with industrial control of processes and systems.

Prior Knowledge

Students will need general engineering knowledge only.

About these worksheets

These worksheets provide a practical environment for the study of Process Control. This is not the only resource a student will need. Students will also benefit from lectures, tutorials, text books and other
provided by the instructor. The experiments in this course should be integrated with teaching to introduce the theory behind it, and reinforced with written examples, assignments and calculations. The internet is a rich source of information on this topic and the instructor should give students research assignments accordingly.

We have set out the learning resource as a series worksheets. A Reference document is available separately and students will need to refer to this for instructions on how to use the hardware and software provided.
It is expected that the students will be given their own copy of the worksheets, either one by one, or as a pack but that they will make notes on separate sheets of paper. The instructor should monitor that students' understanding keeps pace with their progress through the worksheets. One way to do so is to 'sign off' each worksheet, as a student completes it, and in doing so have a brief chat with the student to assess grasp of the ideas involved in the exercises it contains. 'Answers' are not included. They are easy to calculate and are often subjective.

Timings

The Instructor's notes give the timings for the first pass with the first piece of equipment students use. Around 8 - 10 hours in total. Of course it depends on the ability of the students and the number of pieces of Process Control equipment you use. For the second piece of equipment students might half the time and for the third piece it might take a third of the time. Students should definitely be given the opportunity to work with more than one piece of equipment as the problems and characteristics are so different.

Learning Objectives

When used with software provided:

- **E**lectronically controlled systems
- Transducers and controllers, input and output devices
- Computer based control
- Set point, process value
- Block diagrams of control systems
- ON/Off control systems
- Open loop control systems
- Closed loop control systems
- Time constants in control systems
- Errors and Hysteresis
- Calibration of sensors and actuators and scaling factors
- Control equation and Kp, Ki, Kd terms
- P only controllers
- PI controllers
- PID controllers
- Digital control systems
- Effects of sampling frequency in digital systems
- Ziegler Nichols approximations for control systems
- Computer coding of control systems
- Integral wind up and clamping
- Derivative filtering
- Manual tuning of control systems
- Transfer functions and the s plane
- Digital sampled systems and Z terms

When used with Matlab many more learning objectives are available including:

- Transfer functions
- The use of Poles and Zeroes in a Smith chart for stability and performance

Note that these worksheets do not cover this work but the equipment is fitted with an API so that programs can be written in Matlab to facilitate this study.

Instructor guide

MATRIX | PROCESS CONTROL

Work-sheet	Notes for the Instructor	Time
General	At the time of writing there are 5 Process Control systems: Pressure, Temperature, Flow, Level and Pendulum/Servo. The reason for having 5 different systems is that Processes are not the same. The time constants are different and different types of control work with different types of processes. Students need practical experience of more than one process. A personal recommendation is that you would start with Pressure: it is relatively well behaved and reacts in a good time. Using just temperature will frustrate students as the time constants are long and each change students make will take 5 minutes to see toe effects. Temperature should be the last rig students use - and they need to experience it as it is probably the most important process in industry. One set of worksheets works with all 5 systems. Our vision is that students would complete the worksheets for one system and then go on to the next. Of course the second one would be much quicker to complete than the first as the terms, procedures and software would be familiar. In general it might be assumed that the second pass would be half that of the first time taken. Student should be given a digital copy of the Reference document which details how the hardware and software works. They will need to refer to this as they work through the worksheets. The students or a technician will need to set up the software on the computers that will be used - including USB drivers. Answers We don't generally put answers in worksheet - in Process Control often measurements are subjective so there is no one answer. We do have a document that shows results which comes to much the same thing. The results document is product code CT9228 and is available from Matrix or dealers. It shows many Process Control settings, graphs and Zeigler Nichols calculations and will save you some time in making demonstrations to your class.	

Work-sheet	Notes for the Instructor	Time
General	Using the equipment over the internet The equipment is fitted with Wi-fi and LAN connections so that it can be used over the internet. We have managed this successfully here in the UK and connection bandwidth is an important factor in the experience that students have. There is a Data Rate Divider setting in the properties section of the PID software that limits the data rate used for graphing and you may need to experiment with this to get good results. The students experience using this equipment over the internet is not as good as it is when next to the equipment and it can be enhanced by the use of cameras. The Level rig works particularly well here as a camera gives a good feel as to what is going on.	

Work-sheet	Notes for the Instructor	Time
1 Understanding drive	In the first worksheet we introduce the students to the equipment and some Process control terms: Drive, load etc. The software application used here is 'Open ManualControl.exe' which gives students access to the electronics controller without any allowance for scaling factors - so they can understand how digital systems use PWM for driving processes. PWM drive always varies between 0 and 100%. Drive affects different processes in different ways. Students derive a Scaling Factor from their work. For PID this is a starting value - the PID equation evens out any errors.	40 mins
2 Understanding the sensors	In control systems sensors often need zeroing, calibrating and scaling. Students use 'Open ManualControl.exe' which allows them to come up with a scaling factor for the sensor and in some systems can verify the value of the digital sensor against a known analogue equivalent. This is a good skill to understand even though more and more sensors are digital and have no need of initial calibration.	40 mins
3 On/Off control systems	Students can explore On/Off control systems using the program 'Open OnOffControl.exe'. They need to enter the Scaling Factors deduced in the previous worksheet into the Properties of the software . If you have more than one type of Process Control rig then you may get students to compare notes at this point: On/Off control can be acceptable for many control systems but it's a disaster for others. Seeing On/Off control for different systems at the same time would be a useful comparison. Students can understand the effects of hysteresis and set the turn on/ turn off values in the software Properties to understand this better.	40 mins
4 System time constant	Students are asked to measure the time constants of the equipment which are vastly different. Time constants differ with different loads and system parameters like weights on the Servo pendulum. The main effects of time constants are in setting the control frequency and in calculating Ziegler Nichols.	40 mins

MATRIX | PROCESS CONTROL

Work-sheet	Notes for the Instructor	Time
5 P controller	'Open PIDControl.exe' is the software that students will use for the bulk of their work with the equipment and is the software for the rest of the worksheets. This is where the real learning on PID systems starts. Students learn that there is always an offset in P only controllers, and that the offset is proportional to the Kp gain. Students are exposed to the control function and its block diagram representation. They also need to start to understand how to characterise the performance of a system using a step	60 mins
6 PI controller	Students can build on the P only controller with an integral function. They learn that the offset disappears and they will also see some systems becoming unstable with values of Kp and Ki. Gaining a qualitative feeling for the effects of different values of Kp and Ki in different systems is important for manual tuning later on and the graphs of the different values and the dynamic Control Equation on the screen helps here.	60 mins
7 PID control-ler	Students can build on the PI only controller with an differential function. The value of the d term varies between the equipment and where response time is short d terms can often just add noise and using the d term without derivative filtering can be ineffective in some systems. Students can find this frustrating as there is often no 'answer' to the problem. The Pressure system works well without derivative filter. The servo pendulum is very difficult to get working without a derivative filter as the DC motor is powerful and the disc is directly coupled - it's a very short time constant. In this worksheet we introduce the student to s terms, Z terms and how the Control Equation can be related to the Transfer Function. Transfer functions are well explained elsewhere - most often for analogue systems. Finding good explanations on the internet for transfer functions in digital systems is hard (let us know). Of course we are not going to do that well in half a page of text but we have highlighted something of it for you to build on. The frustrating thing here is that the code for digital Process Control systems is so easy: given that coding is so ubiquitous we felt it necessary to show students how to code PID systems at each step. Understanding wind up and derivative filtering is made easy when looking at the coding.	40 mins

Instructor guide

Work- sheet	Notes for the Instructor	Time
8 Zeigler Nichols algorithm	Zeigler Nichols is lovely shortcut to control systems that - for well behaved systems - means that analysis of the system in the s plane with poles and zeroes is not needed. Zeigler Nichols is well documented for analogue systems but digital ones need to be handled slightly differently - the Control frequency is part of the calculation - you will need to divide the value of Ki by the sample frequency and you will need to multiply the value of Kd by the sample frequency to get the values you need. Students should set up a simple spreadsheet with these calculations in as they will use it many times. Defining the point of 'neutral stability' is difficult in some systems: some behave beautifully (Pressure) and in others (servo pendulum) its hard to see where the point of 'neutral stability' starts.	40 mins
9 Integral windup	The graph on the software is superb at showing the problem with integral windup and the solution that integral clamping provides. Students can alter the system with a 'fault' like a blocked tube or a stuck bearing and see how windup occurs.	40 mins
10 Derivative filter	As with the integral wind up the graph on the software is superb at showing the difference that derivative filtering can make. Some systems will not work well with derivative filtering turned off.	40 mins
11 Manual tun- ing	Students need time to play with the systems here to get a feel for how the terms in Kp, Ki, Kd affect each system and get a feel as to how to alter the values manually. This is a bit of a black art and students will need some support and explanation.	60 mins
MATLAB / Labview	All manner of things are possible in advanced packages like MATLAB and Labview. We have not catered for these users in the worksheets but the equipment has a full API which these users can tap into and use for more mathematical operations. The API is detailed in the Reference document CP5956.	

Version control

19 11 20 first release

27 07 21 Additional material added for Servo

pendulum apparatus.

23 08 21 Tutor's notes added

28 07 23 Reformatted to new style

18 10 24 .bat replaced with .exe in all cases