

PROGRAMMABLE LOGIC CONTROLLERS

OUTCOME 6

THREE TERM (P. I. D) CONTROL

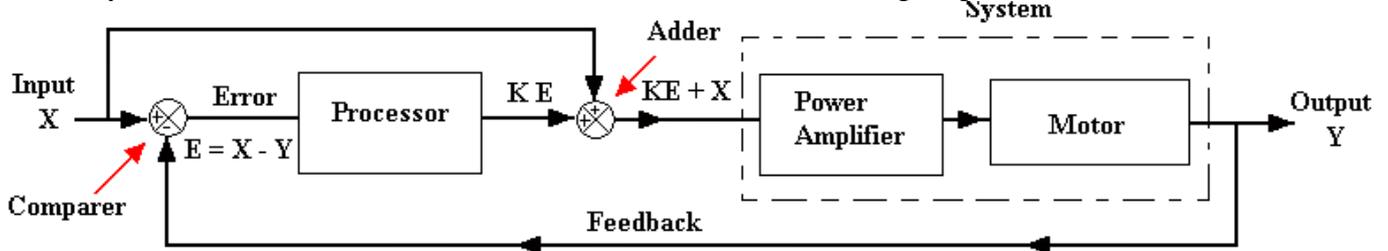
This is set at the British Edexcel National level NQF 4

Contents

1. *Introduction*
2. *Proportional Control*
3. *Integral Control*
4. *Derivative Control*
5. *3 Term Control*
6. *Digital P. I. D Control*
7. *Numerical Processing*
8. *Digital Proportional*
9. *Digital Integral*
10. *Digital Derivative*
11. *Digital 3 Terms*
12. *Note on Types of Control Systems*
13. *PLC Programming*

1. Introduction

The following is a stand alone tutorial to explain the meaning of P.I.D. or 3 Term control used in analogue and digital control systems. P.I.D. stands for Proportional, Integral and Derivative. It is difficult to understand this feature on a PLC unless you are familiar with control theory. Here is an attempt to explain it briefly using the control of the speed of an electric motor as an example. Here is a block diagram of a control system for an electric motor. The PID element will be in the signal processor.



The idea is to make the output speed Y the same as the input X at all times. The input would be from some kind of calibrated instrument producing the setting as a voltage. The output speed would be converted into a voltage by some form of transducer but this is not shown to keep it simple. If we change the value of X or some external disturbance causes Y to change, we get an error. The error is determined by the comparer or error detector. This works by subtracting Y from X so the error is:

$$E = X - Y.$$

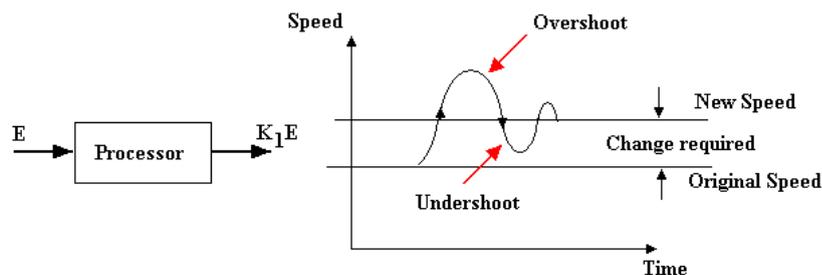
This is called *Negative Feedback*. The error is then processed by the processor. For the moment treat this as an amplifier with a gain of K . The signal from the processor is KE and this is added to the input with an adder. The input to the power amplifier is $X + KE$. The power amplifier converts this signal into electric power that drives the motor. The power amplifier should be powerful enough to cope with all requirements.

Suppose for some reason there is an error so that the motor is running slower than that selected. The input to the system will increase by an amount KE so the motor will speed up again until $E = 0$ and we have restored balance. If the motor is running too fast, E is negative and the input to the system is reduced and it slows down until balance is restored.

Let us consider the importance of the *Processor*. We have just treated it as an amplifier with a gain of K . In fact this is the P.I.D. unit and processes the signal in three ways, Proportional, Integral and Derivative. This is how it works.

2. Proportional Control

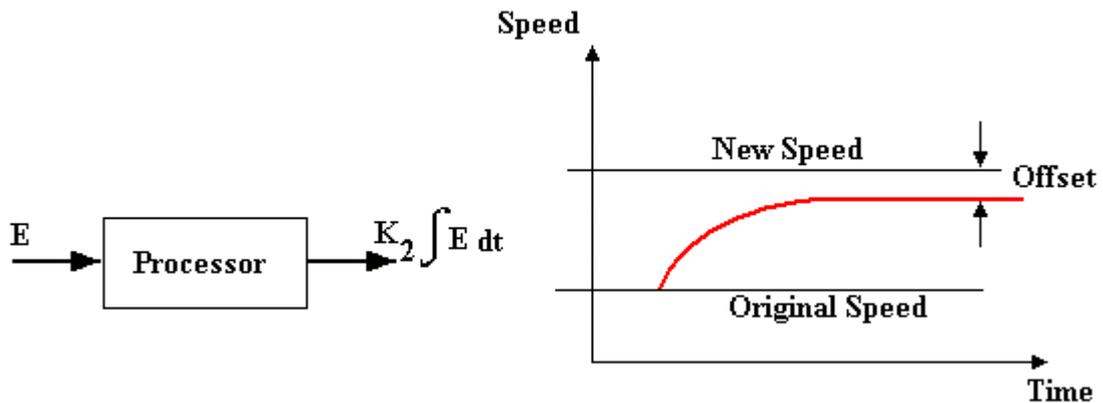
If the processor is a simple amplifier with proportional gain then the output signal is $K_1 E$



K_1 is the constant of proportionality but in fact it is the gain of the amplifier. The larger the value of this gain, the faster the motor will respond to any error. If this gain is too high, the motor can become unstable and speed up too quickly to stop in time at the right speed. This is called *Overshoot*. It can also work the other way when slowing down (*Undershoot*). In the worst case the speed continuously goes up and down and doesn't settle at the correct value. This is known as *Hunting*. We aim for the maximum gain without producing ill affects.

3. Integral Control

This is used to get rid of **Offset Error**. Typically this occurs when the output settles at the wrong speed. This could be due to stiction in the system, small errors in the transducers and most likely in the case of a motor, excessive load slowing it down. If this happens we need to make the input to the system even larger to produce more power to the motor.

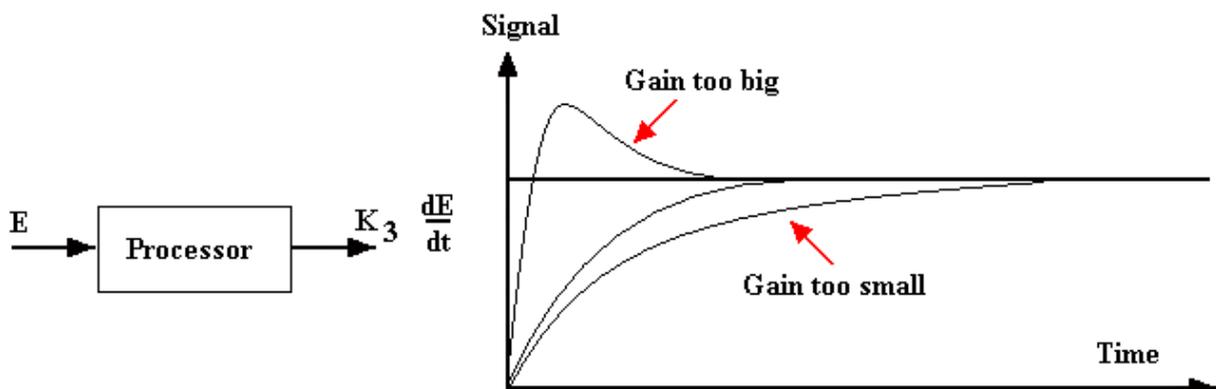


Integral control action increases the power with time such that the output is $k_2 \int E dt$

K_2 is the integral constant. So long as an error exists, the output will grow with time. In this way the power signal will build up until the motor speed is back to its normal level. The power signal will build up until it is big enough to overcome the stiction or load and so totally eliminate the error. Of course there is a limit to which anything can grow and the system should be designed to have a suitable limit.

The use of integral control brings in a new problem, an increase in the tendency to overshoot, especially when the system has a large proportional constant. This is solved by using derivative action also.

4. Derivative Control



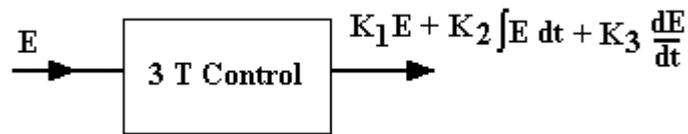
Ideally we wish an error to be corrected as quickly as possible with no overshoot. To do this we need to have maximum gain at the beginning of the correction and then reduce the gain as the error approaches zero. This is achieved with derivative control. With derivative control the output of the processor is:

$$K_3 \frac{dE}{dt}$$

The signal is proportional to the rate of change of the error and K_3 is the derivative constant. It follows that at the start of the change dE/dt is greatest. At this point the power will be at its greatest value. As the error is reduced, dE/dt gets smaller and the motor is decelerated. With ideal derivative action, the response is as shown.

5. 3 Term Control

The signal processor may incorporate all three control elements and it then becomes a Three Term Controller.



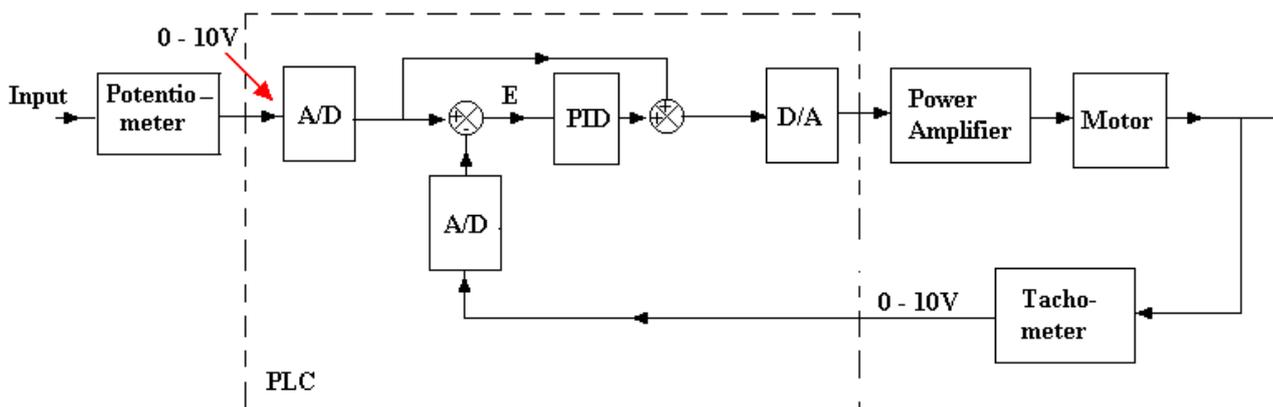
The three terms used by the controller are Proportional, Integral and Derivative. For this reason it is also called P.I.D. control. In order to obtain optimal response for a given system, all three control terms are used and the controller produces the following action.

$$\text{Output} = K_1 E + K_2 \int E dt + K_3 dE/dt$$

This is called 3 Term control or PID control. In analogue systems the control unit will have the facility to adjust K_1 , K_2 and K_3 in order to obtain optimal settings for the given system. This is often called *Tuning*.

6. Digital P. I. D Control

Consider the complete block diagram below. It shows how an electric motor might be controlled with a programmable Logic Controller.



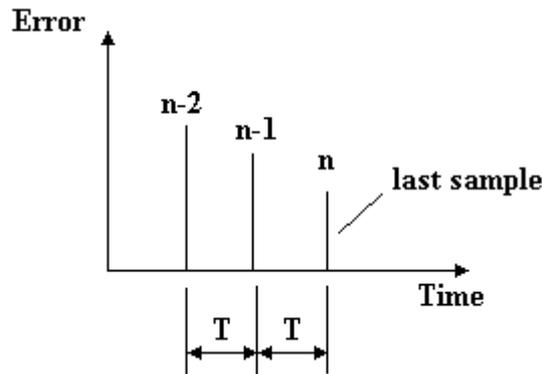
The input to the system is a voltage in the range 0 – 10 V generated by a potentiometer. The output speed of the motor is also in the range 0 – 10 V generated by a tachometer. Both signals are converted into digital form with A/D converters and read into the PLC registers. The negative feed back is then achieved with a subtraction routine and the error resulting is digitally processed with a 3 term controller. The result is added to the original input and converted back into an analogue signal and fed out to the power amplifier. For those studying the PLC, we need to know how the P.I.D. action is achieved. Not many PLCs have the facility to conduct P. I. D. control but those that do have special commands to invoke the action. In its simplest form, it involves increasing or decreasing the value of the number stored in the register. A programming command such as PIDINC causes the processing to take place on the number stored. The following description is of an *Incremental P. I. D. Controller* which is a numerical method.

7. Numerical Processing

The processing is based on digitally sampling the error at regular time intervals of T seconds. From this we can deduce the following.

- The incremental change in the error for proportional control
- The incremental change in the time integral for integral control
- The incremental change in the rate of change of error for derivative control

Let the most recent sample be the n^{th} sample. The one before that was $n - 1$ and before that $n - 2$.



8. Digital Proportional

Let the error be x (it was E previously) and this is multiplied by the proportional constant k_1 to give the output of the processor Q_n .

$Q_n = k_1 x_n$. The output of the previous sample would be $Q_{n-1} = k_1 x_{n-1}$

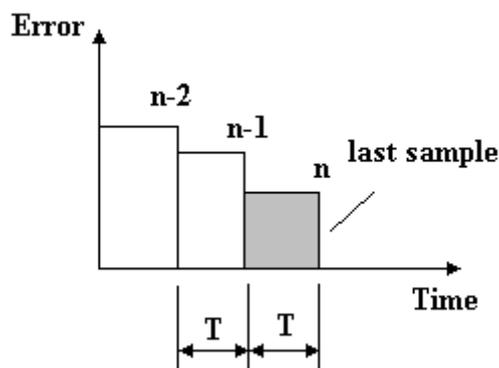
The *incremental change in the processor output* is then the difference between the two values.

$$\Delta Q = k_1 x_n - k_1 x_{n-1} = k_1(x_n - x_{n-1})$$

9. Digital Integral

Consider the time – error graph as made up of thin strips of width T . The time integral of the error over time T seconds is the shaded part of the graph. The incremental output is obtained by multiplying this by the integral constant k_2 so:

$$\Delta Q = k_2 T x_n$$



10. Digital Derivative

In this we need to determine the incremental change in rate of change of the error so we need three samples to give us two successive rates of change. We calculate the rate of change between two adjacent samples by taking the change and dividing by the time taken.

The rate of change between the sample n and sample $n - 1$ is

$$\frac{x_n - x_{n-1}}{T}$$

The rate of change between the samples $n - 1$ and $n - 2$ is

$$\frac{x_{n-1} - x_{n-2}}{T}$$

The difference between these two results is the incremental change

$$\frac{x_n - x_{n-1}}{T} - \frac{x_{n-1} - x_{n-2}}{T}$$

To find the incremental change ΔQ we must multiply this by derivative constant k_3 .

$$\Delta Q = k_3 \left(\frac{x_n - x_{n-1}}{T} - \frac{x_{n-1} - x_{n-2}}{T} \right)$$

$$\Delta Q = k_3 \left(\frac{x_n - x_{n-1} - x_{n-1} + x_{n-2}}{T} \right)$$

$$\Delta Q = k_3 \left(\frac{x_n - 2x_{n-1} + x_{n-2}}{T} \right)$$

11. Digital 3 Terms

For all three control terms, the incremental change to the output is

$$\Delta Q = k_1(x_n - x_{n-1}) + k_2Tx_n + k_3\left(\frac{x_n - 2x_{n-1} + x_{n-2}}{T}\right)$$

This value may be computed and then added to the previous value to give the new output Q.

Hence

$$Q_n = Q_{n-1} + k_1(x_n - x_{n-1}) + k_2Tx_n + k_3\left(\frac{x_n - 2x_{n-1} + x_{n-2}}{T}\right)$$

This is the signal that is added to the original input and then put out to the A/D converter and the power amplifier.

12. Note on Types of Control Systems

Note that in many control systems, we do not add the input to the processed error. In the case of systems such as an electric motors, we cannot drive the motor from the error alone because when there is no error, there will be no power. In position control systems (e.g. robots, satellite dishes and machine tools), when the actuators are in the correct position, we need no power to keep it there so we drive the system from the error alone. If there is any error power is provided to reduce it back to zero.

13. PLC Programming

Some PLC's have a CPU operation to calculate the incremental change. Typically this command is PIDINC. The sampling time may be set also by adding the sampling time or rate to the PIDINC command. A typical system would have an analogue/digital I/O module. The set value and output values would be typically analogue (but could be digital). These values would be read into the PLC and stored. This would be repeated at intervals of T seconds. Three samples would be stored and then updated with each sample. The numerical operation as detailed previously is carried out. A further command to add the incremental change to the last value must also be used and the resulting value placed out on the I/O module as an analogue signal. A typical ladder diagram would contain the sub-routine shown below to do this. The exact diagram will depend upon the type of PLC used.

