

[The PID Control Algorithm How it works, how to tune it, and how to use it]

#chapter1

Process control is the measurement of a process variable, the comparison of that variables with its respective set point, and the manipulation of the process in a way that will hold the variable at its set point when the set point changes or when a disturbance changes the process.

프로세스 제어는 프로세스의 변수 측정이며. 그 변수들이 각각의 set point를 비교하면서, set point가 바뀔때 또는 교란이 프로세스를 변화시킬때, set point가 유지하는 방식으로 프로세스를 조작한다.

An example is shown in Figure 1.

예가 그림 1에 나와있다.

In this example, the temperature of the heated water leaving the heat exchanger is to be held at its set point by manipulating the flow of steam to the exchanger using the steam flow valve.

예 에서, 가열된 물의 온도는 열 교환기가 스팀흐름밸브를 사용하여 교환기의 스팀흐름을 조작하면서 set point를 유지 해놓는다.

In this example, the temperature is known as the measured or controlled variable and the steam flow (or the position of the steam valve) is the manipulated variable.

이 예에서, 온도는 측정 또는 제어 변수로 알려졌으며, 스팀 흐름(또는 스팀밸브위치)는 조작변수이다.

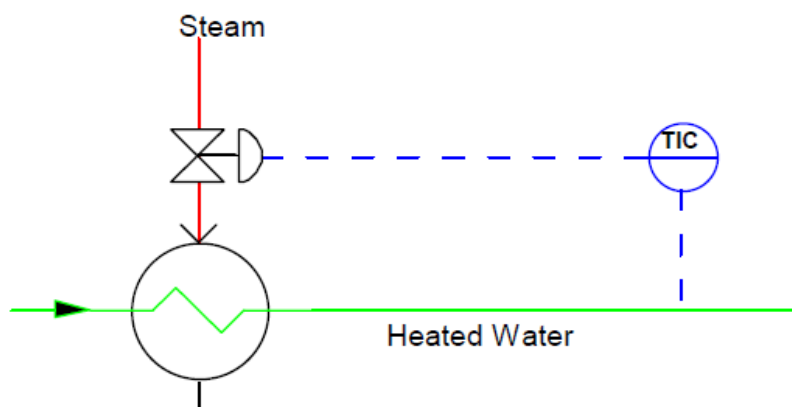


Figure 1 Typical process control loop – temperature of heated water.

Most processes contain many variables that need to be held at a set point and many variables that can be manipulated.

대부분의 프로세스들은 set point지점에서 유지해야하고 조작할수 있는 많은 다양한 변수를 담고있다.

Usually, each controlled variable may be affected by more than one manipulated variable and each manipulated variable may affect more than one controlled variable.

보통, 각각의 제어 된 변수는 하나 이상의 조작 된 변수 및 각 조작 된 변수의 영향을받을 수 있고, 하나 이상의 제어 변수에 영향을 미칠 수 있습니다.

However, in most process control systems manipulated variables and control variables are paired together so that one manipulated variable is used to control one controlled variable. 그러나 대부분의 공정 제어 시스템에서 조작 변수와 제어 변수는 함께 쌍을 이루어 하나의 조작 변수가 하나의 제어 변수를 제어하는 데 사용됩니다.

Each pair of controlled variable and manipulated variable, together with the control algorithm, is referred to as a control loop.

제어 알고리즘과 함께 제어 변수 및 조작 변수의 각 쌍을 제어 루프라고합니다.

The decision of which variables to pair is beyond the scope of this publication.

쌍을 만들 변수에 대한 결정은 본 간행물의 범위를 벗어납니다.

It is based on knowledge of the process and the operation of the process.

이 책은 프로세스에 대한 지식과 프로세스 작동에 기반합니다.

In some cases control loops may involve multiple inputs from the process and multiple outputs to the processes.

어떤 경우에는 제어 루프가 프로세스의 여러 입력과 프로세스의 여러 출력을 포함 할 수 있습니다.

The first part of this book will consider only single input, single output loops.

이 책의 첫 번째 부분에서는 단일 입력, 단일 출력 루프만을 고려합니다.

Later we will discuss some multiple loop control methods.

나중에 우리는 다중 루프 제어 방법을 논의 할 것이다.

There are a number of algorithms that can be used to control the process.

프로세스를 제어하는 데 사용할 수있는 여러 가지 알고리즘이 있습니다.

The most common is the simplest: an on/off switch.

가장 일반적인 방법은 ON / OFF 스위치입니다

For example, most appliances use a thermostat to turn the heat on when the temperature falls below the set point and then turn it off when the temperature reaches the set point.

예를 들어, 대부분의 기기는 온도가 set point 아래로 떨어지면 온도를 올리기 위해 온도 조절기를 사용하며, 온도가 설정 점에 도달하면 전원을 끕니다.

This results in a cycling of the temperature above and below the set point but is sufficient for most common home appliances and some industrial equipment.

이것은 설정 온도보다 높거나 낮은 온도 사이클링을 초래하지만 대부분의 일반 가전 제품 및 일부 산업용 장비에는 충분합니다.

To obtain better control there are a number of mathematical algorithms that compute a change in the output based on the controlled variable.

더 나은 제어를 얻으려면 제어 변수를 기반으로 출력의 변화를 계산하는 여러 수학적 알고리즘이 있습니다.

Of these, by far the most common is known as the PID (Proportional, Integral, and Derivative) algorithm, on which this publication will focus.

이 중 가장 많이 사용되는 것은 PID (Proportional, Integral, and Derivative) 알고리즘으로 알려져 있으며, 여기에 이 책자는 초점을 맞춥니다.

First we will look at the PID algorithm and its components.

먼저 PID 알고리즘과 그 구성 요소를 살펴 보겠습니다.

We will then look at the dynamics of the process being controlled.

그런 다음 제어되는 프로세스의 역학을 살펴 봅니다.

Then we will review several methods of tuning (or adjusting the parameters of) the PID control algorithm.

그런 다음 PID 제어 알고리즘을 조정 (또는 매개 변수 조정)하는 몇 가지 방법을 검토합니다.

Finally, we will look at several ways multiple loops are connected together to perform a control function.

마지막으로, 우리는 여러 루프가 함께 연결되어 제어 기능을 수행하는 몇 가지 방법을 살펴볼 것입니다.

1.1 THE CONTROL LOOP

1.1 제어루프

The process control loop contains the following elements:

프로세스 제어 루프에는 다음 요소가 포함된다:

- The measurement of a process variable.

프로세스 변수의 측정

A sensor, more commonly known as a transmitter, measures some variable in the process such as temperature, liquid level, pressure, or flow rate, and converts that measurement to a signal (typically 4 to 20 ma.) for transmission to the controller or control system.

더 일반적으로 송신기로 알려진 센서는 온도, 액체 압력 또는 유량을 측정하여 그 측정 값을 컨트롤러 또는 제어 시스템에 전송하는 신호 (보통 4 ~ 20 mA)로 변환합니다.

- The control algorithm.

제어 알고리즘.

A mathematical algorithm inside the control system is executed at some time period (typically every second or faster) to calculate the output signal to be transmitted to the final control element.

제어 시스템 내부의 수학 알고리즘은 최종 제어 요소에 전송 될 출력 신호를 계산하기 위해 일정 시간 (일반적으로 1 초마다 또는 그보다 더 빠름)에 실행됩니다.

- A final control element.

최종 제어 요소

A valve, air flow damper, motor speed controller, or other device receives a signal from the controller and manipulates the process, typically by changing the flow rate of some material.

밸브, 공기 흐름 댐퍼, 모터 속도 컨트롤러 또는 다른 장치는 컨트롤러로부터 신호를 수신하고 일반적으로 일부 재료의 유량을 변경하여 공정을 조작합니다.

- The process.

과정

The process responds to the change in the manipulated variable with a resulting change in the measured variable.

이 프로세스는 조작 된 변수의 변화에 측정 된 변수의 결과 변화에 응답합니다.

The dynamics of the process response are a major factor in choosing the parameters used in the control algorithm and are covered in detail in this publication.

프로세스 응답의 동적 특성은 제어 알고리즘에 사용되는 매개 변수를 선택하는 주요 요소이며 본 설명서에서 자세히 다룹니다.

이러한 요소들의 상호연결이 아래 그림2와같이설명 되었다.

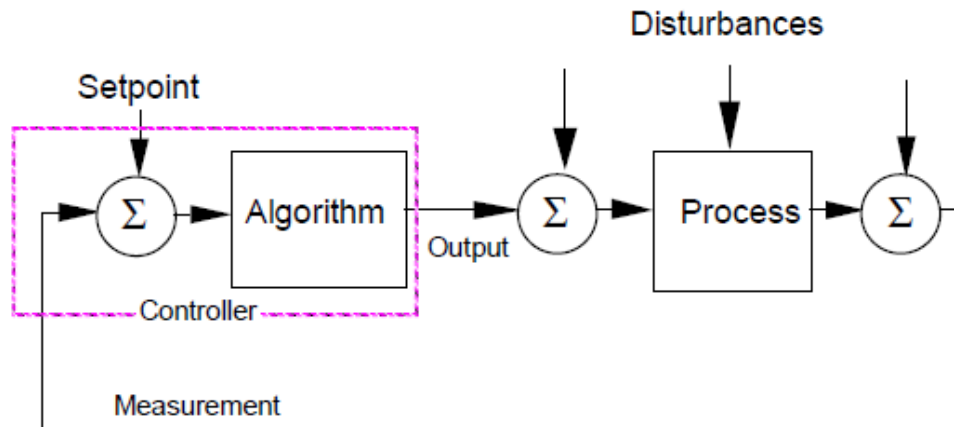


Figure 2 Interconnection of elements of a control loop.

The following signals are involved in the loop:

다음 신호가 루프에 포함된다.

- The process measurement, or controlled variable.
- 프로세스 측정 또는 제어 변수.

In the water heater example, the controlled variable for that loop is the temperature of the water leaving the heater.

온수기 예에서, 그 루프에 대한 제어 변수는 히터를 떠나는 물의 온도입니다

- The set point, the value to which the process variable will be controlled.
- 설정 값, 프로세스 변수가 제어되는 값.
- One or more load variables, not manipulated by this control loop, but perhaps manipulated by other control loops.
- 하나 이상의 load 변수. 이걸 제어 루프에 의해 조작되지는 않지만 다른 제어 루프에 의해 조작 될 수 있습니다.

In the steam water heater example, there are several load variables.

스팀 온수기의 예에서, 몇 가지 부하 변수가 있습니다.

The flow of water through the heater is one that is likely controlled by some other loop.

히터를 통과하는 물의 흐름은 다른 루프에 의해 제어 될 가능성이 높습니다.

The temperature of the cold water being heated is a load variable.

가열되는 냉수의 온도는 load 변수입니다.

If the process is outside, the ambient temperature and weather (rain, wind, sun, etc.) are load variables outside of our control.

프로세스가 실외에있는 경우 주변 온도와 날씨 (비, 바람, 태양 등)는 우리가 통제 할 수없는 load 변수입니다.

A change in a load variable is a disturbance.

부하 변수의 변화는 교란이다.

Other measured variables may be displayed to the operator and may be of importance, but are not a part of the loop.

다른 측정 변수는 operator에게 중요하게 표시 될 수 있지만, 루프의 일부는 또 아닌것도 있다.

1.2 ROLE OF THE CONTROL ALGORITHM

제어 알고리즘의 역할

The basic purpose of process control systems such as is two-fold:

다음과 같은 프로세스 제어 시스템의 기본 목적은 두 가지입니다.

To manipulate the final control element in order to bring the process measurement to the set point whenever the set point is changed, and to hold the process measurement at the set point by manipulating the final control element.

설정 점이 변경 될 때마다 프로세스 측정을 설정 점으로 가져오고 최종 제어 요소를 조작하여 설정 점에서 프로세스 측정을 유지하기 위해 최종 제어 요소를 조작합니다.

The control algorithm must be designed to quickly respond to changes in the set point (usually caused by operator action) and to changes in the loads (disturbances).

제어 알고리즘은 설정 포인트 (일반적으로 operator 동작으로 인해 발생) 및 부하 (교란)의 변화에 신속하게 대응할 수 있도록 설계되어야합니다.

The design of the control algorithm must also prevent the loop from becoming unstable, that is, from oscillating.

제어 알고리즘의 설계는 또한 루프가 불안정해지는 것을 방지해야합니다

1.3 AUTO/MANUAL

자동/수동

Most control systems allow the operator to place individual loops into either manual or automatic mode.

대부분의 제어 시스템을 통해 operator는 개별 루프를 수동 또는 자동 모드로 설정할 수 있습니다.

In manual mode the operator adjusts the output to bring the measured variable to the desired value.

수동 모드에서 operator는 측정 된 변수를 원하는 값으로 가져 오도록 출력을 조정합니다.

In automatic mode the control loop manipulates the output to hold the process measurements at their set points.

자동 모드에서 제어 루프는 프로세스 측정 값을 설정 값으로 유지하십시오.

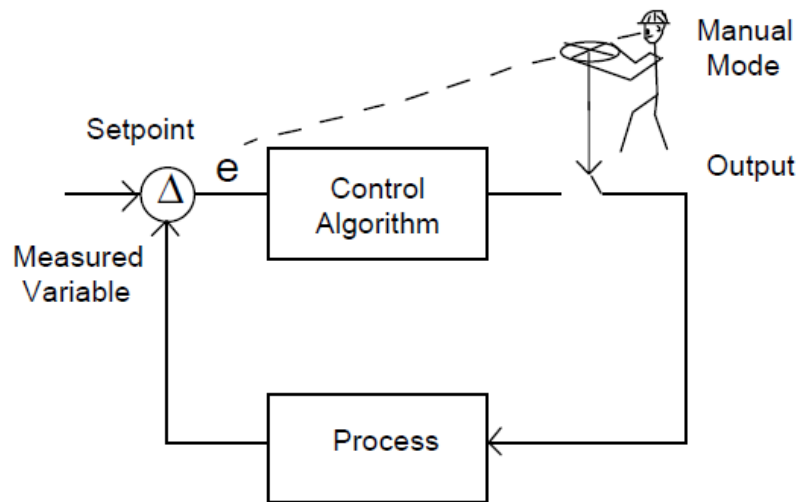


Figure 3 A control loop in manual.

In most plants the process is started up with all loops in manual.

대부분의 시설에서 수동으로 모든 루프를 사용하여 프로세스가 시작된다.

During the process startup loops are individually transferred to automatic.

프로세스가 시작되는 동안 루프는 개별적으로 자동으로 전송됩니다.

Sometimes during the operation of the process certain individual loops may be transferred to manual for periods of time.

때로는 프로세스를 실행하는 동안 특정 루프가 일정 기간 동안 수동으로 전환 될 수 있습니다.

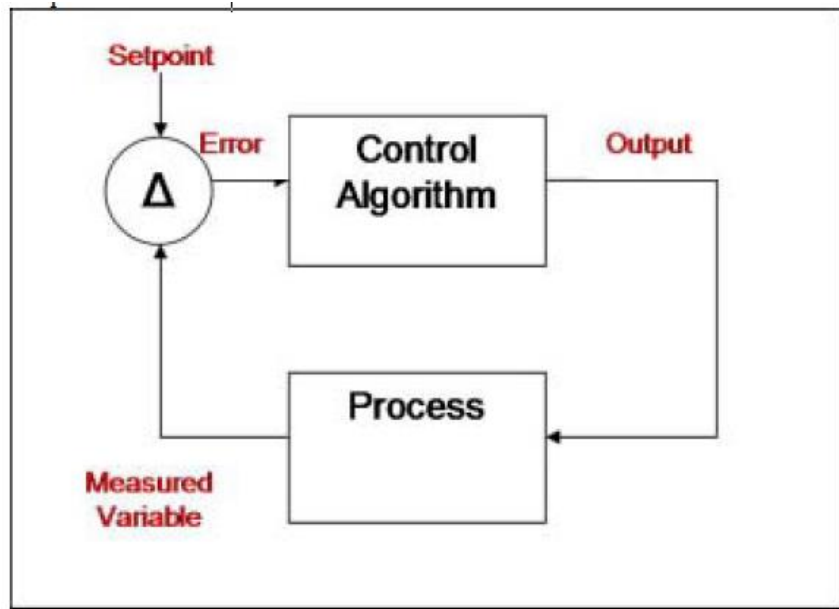


Figure 4 A control loop in automatic

#CHAPTER 2

THE PID ALGORITHM

PID 알고리즘

In industrial process control, the most common algorithm used (almost the only algorithm used) is the time-proven PID—Proportional, Integral, Derivative—algorithm.

산업 프로세스 제어에서 가장 많이 사용되는 알고리즘 (거의 유일한 알고리즘 사용)은 시간에 입증된 PID 비례, 적분, 미분 알고리즘입니다.

In this chapter we will look at how the PID algorithm works from both a mathematical and an implementation point of view.

이 장에서는 수학적 관점과 구현 관점에서 PID 알고리즘이 어떻게 작동하는지 살펴 보겠습니다.

2.1 KEY CONCEPTS

주요 개념

- The PID control algorithm does not “know” the correct output that will bring the process to the set point.

PID 제어 알고리즘은 프로세스를 설정치로 가져올 정확한 출력을 "알지 못합니다".

The PID algorithm merely continues to move the output in the direction that should move the process toward the set point until the process reaches the set point.

PID 알고리즘은 프로세스가 설정치에 도달 할 때까지 프로세스를 설정 포인트쪽으로 이동해야하는 방향으로 출력을 계속 이동합니다.

The algorithm must have feedback (process measurement) to perform.

수행 할 알고리즘에는 피드백 (프로세스 측정)이 있어야합니다

If the loop is not closed, that is, the loop is in manual or the path between the output to the input is broken or limited, the algorithm has no way to “know” what the output should be.

루프가 닫혀 있지 않은 경우, 즉 루프가 수동이거나 출력과 입력 사이의 경로가 끊어 지거나 제한되어있는 경우 알고리즘은 출력 내용을 "알 수 있는"방법이 없습니다.

Under these (open loop) conditions, the output is meaningless.

이러한 (개방 루프) 조건에서 출력은 의미가 없습니다.

- The PID algorithm must be “tuned” for the particular process loop.
- PID 알고리즘은 특정 프로세스 루프에 대해 "조정"되어야합니다.

Without such tuning, it will not be able to function.

이러한 조정이 없으면 작동하지 않습니다

To be able to tune a PID loop, each of the terms of the PID equation must be understood.

PID 루프를 조정할 수 있으려면 PID 방정식의 각 항을 이해해야합니다

The tuning is based on the dynamics of the process response and is will be discussed in later chapters.

튜닝은 프로세스 응답의 다이내믹을 기반으로하며 이후 장에서 논의 될 것입니다.

2.2 ACTION

행동

The most important configuration parameter of the PID algorithm is the action.

PID 알고리즘의 가장 중요한 구성 매개 변수는 동작입니다.

Action determines the relationship between the direction of a change in the input and the resulting change in the output.

동작은 변경된 방향의 입력값과 출력의 결과 변경 간의 관계를 결정합니다.

If a controller is direct acting, an increase in its input will result in an increase in its output.

컨트롤러가 직접 작용 인 경우, 입력이 증가하면 출력이 증가합니다.

With reverse action an increase in its input will result in a decrease in its output.

역 동작으로 입력이 증가하면 출력이 감소합니다.

The controller action is always the opposite of the process action.

컨트롤러 동작은 항상 프로세스 동작의 반대입니다.

2.3 THE PID RESPONSES

PID 응답

The PID control algorithm is made of three basic responses, Proportional (or gain), integral (or reset), and derivative.

PID 제어 알고리즘은 비례 (또는 이득), 적분 (또는 리셋) 및 미분의 세 가지 기본 응답으로 구성됩니다.

In the next several sections we will discuss the individual responses that make up the PID controller.

다음 몇 섹션에서는 PID 컨트롤러를 구성하는 개별 응답에 대해 설명합니다.

In this book we will use the term called “error” for the difference between the process and the set point.

이 책에서는 프로세스와 설정 값의 차이점에 대해 "오류"라는 용어를 사용합니다.

If the controller is direct acting, the set point is subtracted from the measurement; 컨트롤러가 직접 형인 경우 설정치가 측정에서 제외됩니다.

if reverse acting the measurement is subtracted from the set point.

측정 값의 역 동작이 설정 점에서 뺄 때.

Error is always in percent.

오류는 항상 백분율입니다.

Error = Measurement-Set point (Direct action)

오류 = 측정 - 설정 지점 (직접 동작)

Error = Set point-Measurement (Reverse action)

오류 = 설정 포인트 - 측정 (역 동작)

2.4 PROPORTIONAL

비례적

The most basic response is proportional, or gain, response.

가장 기본적인 응답은 비례 또는 이득입니다.

In its pure form, the output of the controller is the error times the gain added to a constant known as “manual reset”.

순수한 형태에서 컨트롤러의 출력은 "수동 리셋"으로 알려진 상수에 게인이 추가 된 시간의 오차입니다.

Output = E x G + k

출력 = error * gain + k

where:

Output = the signal to the process

출력 = 프로세스 신호

E = error (difference between the measurement and the set point.

E = 오류 (측정 값과 설정 값의 차이).

G = Gain

G = 이득

k = manual reset, the value of the output when the measurement equals the set point.

k = 수동 리셋, 측정 값이 설정 값과 같을 때의 출력 값.

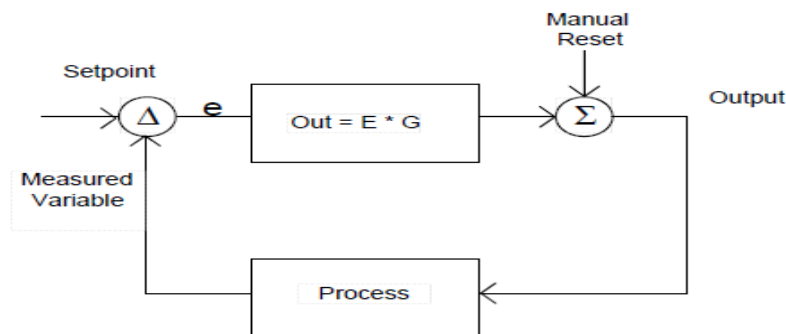


Figure 5 A control loop using a proportional only algorithm.

The output is equal to the error time the gain plus manual reset.

출력은 gain+수동 재설정 오류 시간과 같다.

A change in the process measurement, the set point, or the manual reset will cause a change in the output.

프로세스에서 설정점 또는 수동재설정 값이 변경되면 출력이 변경된다.

If the process measurement, set point, and manual reset are held constant the output will be constant Proportional control can be thought of as a lever with an adjustable fulcrum.

만약 프로세스 측정, 설정 점 및 수동 리셋이 일정하게 유지되면 출력은 비례 제어가 조정 가능한 받침대가있는 레버로 생각할 수 있습니다.

The process measurement pushes on one end of the lever with the valve connected to the other end.

프로세스 측정은 밸브의 다른 쪽 끝에 연결된 레버의 한쪽 끝을 밀었습니다.

The position of the fulcrum determines the gain.

받침점의 위치가 이득을 결정합니다.

Moving the fulcrum to the left increases the gain because it increases the movement of the valve for a given change in the process measurement.

받침대를 왼쪽으로 움직이면 프로세스 측정에서 주어진 변화에 대한 밸브의 움직임이 증가하므로 게인이 증가합니다.

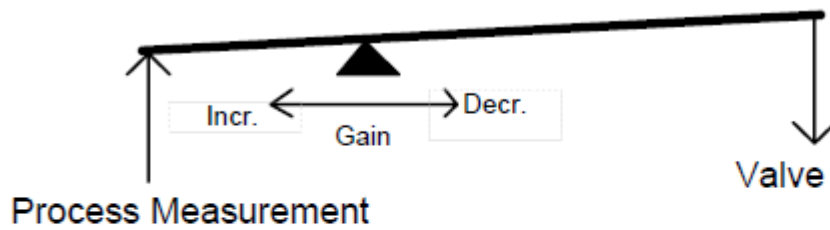


Figure 6 A lever used as a proportional only reverse acting controller.

2.5 PROPORTIONAL—OUTPUT VS. MEASUREMENT

비례 - 출력 VS 측정

One way to examine the response of a control algorithm is the open loop test.
제어 알고리즘의 응답을 검사하는 한 가지 방법은 개방 루프 테스트입니다.

To perform this test we use an adjustable signal source as the process input and record the error (or process measurement) and the output.

이 테스트를 수행하기 위해 조정 가능한 신호 소스를 프로세스 입력으로 사용하고 오류 (또는 프로세스 측정) 및 출력을 기록합니다.

As shown below, if the manual reset remains constant, there is a fixed relationship between the set point, the measurement, and the output.

아래에 표시된 것처럼 수동 재설정이 일정하게 유지되면 설정 지점, 측정 및 출력간에 고정 된 관계가 있습니다.

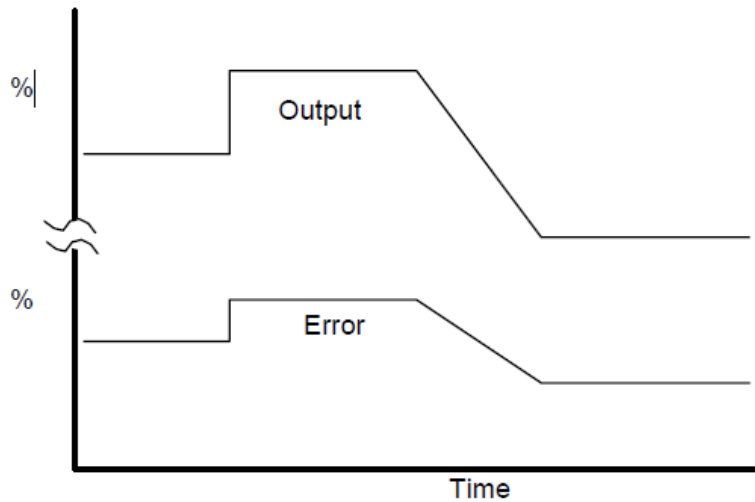


Figure 7 Proportional only controller: error vs. output over time.

2.6 PROPORTIONAL—OFFSET

비례 오프셋

Proportional only control produces an offset.

비례 제어만으로 오프셋이 생성됩니다.

Only the adjustment of the manual reset removes the offset.

수동 재설정 만 조정하면 오프셋이 제거됩니다.

Take, for example, the tank in Figure 8 with liquid flowing in and flowing out under control of the level controller.

예를 들어, 그림 8의 탱크는 레벨 제어기의 제어하에 액체가 흘러 들어가고 흘러 나옵니다.

The flow in is independent and can be considered a load to the level control.

유입수는 독립적이며 레벨 컨트롤에 대한 부하로 간주 될 수 있습니다.

The flow out is driven by a pump and is proportional to the output of the controller.

유출은 펌프에 의해 구동되며 컨트롤러의 출력에 비례합니다.

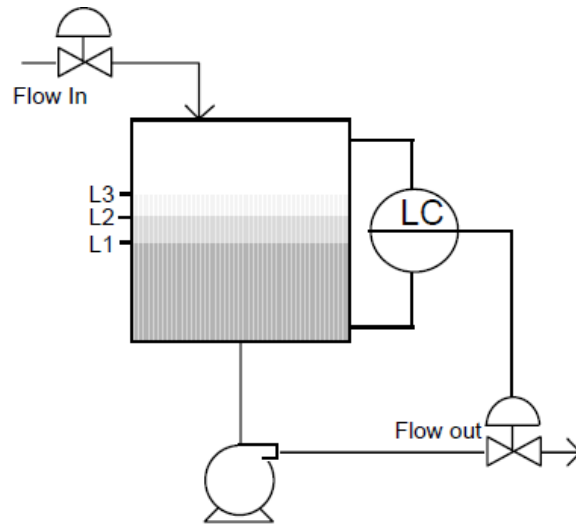


Figure 8 Proportional only level control

The flow from the tank is proportional to the level.

탱크에서의 흐름은 레벨에 비례합니다.

Because the flow out eventually will be equal to the flow in, the level will be proportional to the flow in.

들어오는 흐름이 결국 나가는 흐름과 같기 때문에 흐름의 레벨은 비례합니다.

An increase in flow in causes a higher steady state level.

유량이 증가하면 정상 상태 수준이 높아집니다.

This is called "offset".

이를 "오프셋"이라고합니다.

Assume first that the level is at its set point of 50%, the output is 50%, and both the flow in and the flow out are 500 gpm.

처음에는 레벨이 50 %의 설정치에 있고 출력이 50 %이고 유입 및 유출 모두 500gpm이라고 가정합니다.

Then let's assume the flow in increases to 600 gpm.

그런 다음 유량이 600gpm으로 증가한다고 가정합니다.

The level will rise because more liquid is coming in than going out.

나가는것보다 많은 액체가 들어 오기 때문에 레벨이 올라갑니다.

As the level increases, the valve will open and more flow will leave.

레벨이 높아지면 밸브가 열리고 더 많은 유량이 남습니다.

If the gain is 2, each one percent increase in level will open the valve 2% and will increase the flow out by 20 gpm.

게인이 2이면 레벨이 1 % 증가 할 때마다 밸브가 2 % 열리 며 20gpm만큼 유량이 증가합니다.

Therefore by the time the level reaches 55% (5% error) the output will be at 60% and the flow out will be 600 gpm, the same as the flow in.

따라서 레벨이 55 % (5 % 오류)에 도달 할 때까지 출력은 60 %가되고 유량은 600gpm이 될 것입니다.

The level will then be constant.

레벨은 일정해질 것입니다.

This 5% error is known as the offset.

이 5 % 오류를 오프셋이라고합니다.

Offset can be reduced by increasing gain.

gain(밸브)을 증가시킴으로써 오프셋(오류)을 줄일 수 있습니다.

Let's repeat the above "experiment" but with a gain of 5.

위의 "실험"을 반복하되 5의 gain을 얻도록하겠습니다.

For each 1% increase in level will increase the output by 5% and the flow out by 50 gpm.

레벨이 1 % 증가 할 때마다 출력이 5 % 증가하고 유량이 50gpm만큼 증가합니다.

The level will only have to increase to 52% to result in a flow out of 600 gpm, causing the level to be constant.

레벨은 600gpm 밖에 나오지 않으므로 레벨이 52 %로 높아져서 레벨이 일정 해집니다.

Increasing the gain from 2 to 5 decreases the offset from 5% to 2%.

2에서 5로 게인을 높이면 오프셋이 5 %에서 2 %로 감소합니다

However, only an infinite gain will totally eliminate offset.

무한대의 이득만으로도 오프셋을 완전히 제거 할 수 있습니다.

Gain, however, cannot be made infinite.

그러나 게인은 무한대로 만들 수 없습니다.

In most loops there is a limit to the amount of gain that can be used.

대부분의 루프에서 사용할 수있는 게인의 양에는 제한이 있습니다.

If this limit is exceeded the loop will oscillate.

이 한도를 초과하면 루프가 진동합니다.

2.7 PROPORTIONAL—ELIMINATING OFFSET WITH MANUAL RESET

수동리셋을 이용한 P제거 오프셋

Offset can also be eliminated by adjusting manual reset.

수동 재설정을 조정하여 오프셋을 제거 할 수도 있습니다.

In the above example (with a gain of two) if the operator increased the manual reset the valve would open further, increasing the flow out.

위의 예 (2의 이득)에서 operator가 수동 리셋을 증가 시키면 밸브가 더 열리 며 유량이 증가합니다.

This would cause the level to drop.

이로 인해 레벨이 떨어집니다.

As the level dropped, the controller would bring the valve closed.

레벨이 떨어지면 컨트롤러가 밸브를 닫습니다.

This would stabilize the level but at a level lower than before.

이것은 수준을 안정시킬 것이나 이전보다 낮은 수준으로 유지할 것입니다.

By gradually increasing the manual reset the operator would be able to bring the process to the set point.

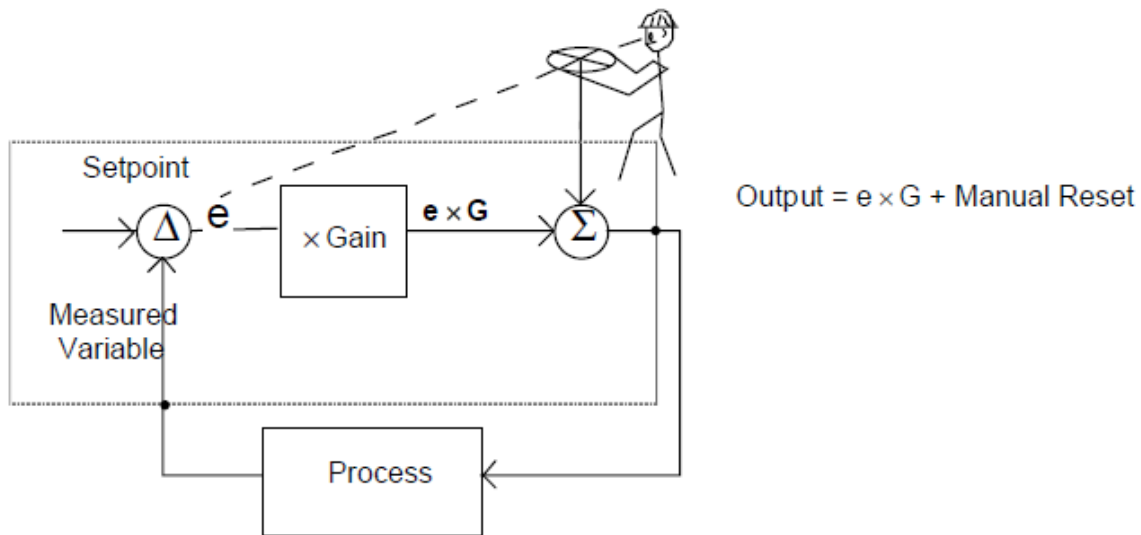
수동 리셋을 점진적으로 증가시킴으로써 운영자는 process를 설정점으로 가져올 수 있습니다.

2.8 ADDING AUTOMATIC RESET

자동 리셋 추가

With proportional only control, the operator “resets” the controller (to remove offset) by adjusting the manual reset:

비례 제어 만 사용하면 작업자가 수동 재설정을 조정하여 컨트롤러를 "재설정"(오프셋 제거) 할 수 있습니다.



The operator may adjust the manual reset to bring the measurement to the set point, eliminating the offset.

operator는 수동 리셋을 조정하여 측정 값을 설정치로 가져 옵셋을 제거 할 수 있습니다.

If the process is to be held at the set point the manual reset must be changed every time there is a load change or a set point change.

프로세스가 설정 포인트에서 유지되어야하는 경우 수동 리셋은 부하가 변경되거나 설정 포인트가 변경 될 때마다 변경되어야합니다.

With a large number of loops the operator would be kept busy resetting each of the loops in response to changes in operating conditions.

많은 수의 루프를 사용하면 작동 조건의 변화에 따라 작동자(operator)가 루프의 각 설정을 재설정 할 수 있습니다.

The manual reset may be replaced by automatic reset, a function that will continue to move the output as long as there is any error:

수동 재설정은 오류가있는 한 출력을 계속 이동시키는 기능인 자동 재설정으로 대체 될 수 있습니다.

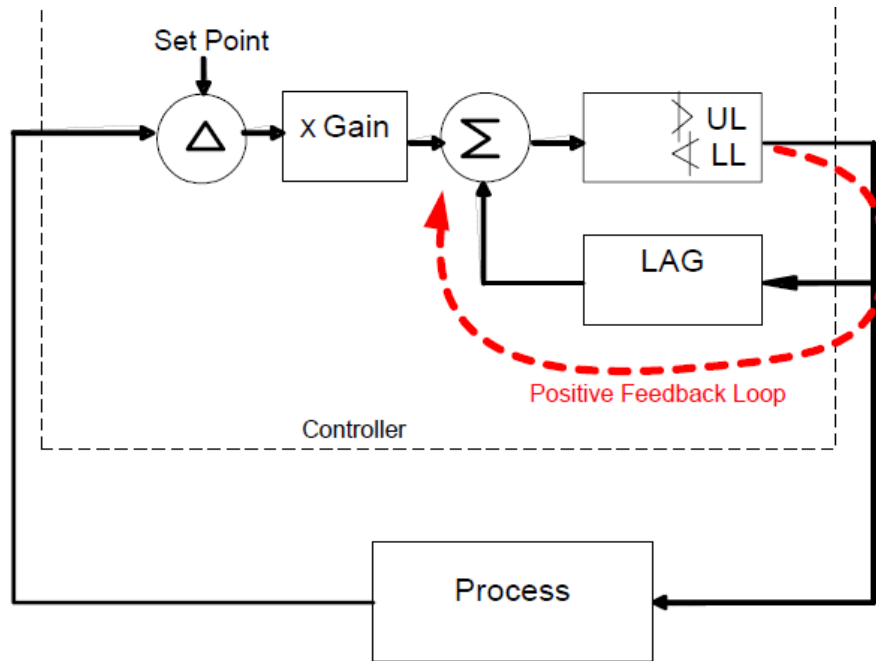


Figure 10 Addition of automatic reset to a proportional controller

The positive feedback loop will cause the output to ramp whenever the error is not zero.
 긍정적 인 피드백 루프는 오류가 0이 아닐 때마다 출력이 램프됩니다.

There is an output limit block to keep the output within specified range, typically 0 to 100%.
 지정된 범위 (일반적으로 0 ~ 100 %) 내에서 출력을 유지하는 출력 제한 블록이 있습니다.

This is called "Reset" or Integral Action.
 이를 "재설정" 또는 적분 동작이라고 합니다.

Note the use of the positive feedback loop to perform integration.
 통합을 수행하기 위해 양의 피드백 루프를 사용합니다.

As long as the error is zero, the output will be held constant.
 오류가 0이면 출력은 일정하게 유지됩니다.

However, if the error is non-zero the output will continue to change until it has reached a limit.
 그러나 오류가 0이 아니면 출력은 한계에 도달 할 때까지 계속 변경됩니다.

The rate that the output ramps up or down is determined by the time constant of the lag and the amount of the error and gain.
 출력이 상승 또는 하강하는 속도는 지연의 시간 상수와 오류 및 게인의 양에 의해 결정됩니다.

2.9 INTEGRAL MODE (RESET)

적분 모드(reset)

If we look only at the reset (or integral) contribution from a more mathematical point of view, the reset contribution is:

보다 수학적인 관점에서의 리셋 (또는 적분) 기여도 만 살펴 본다면 재설정 기여도는 다음과 같습니다.

$$\text{Out} = g \times K_r \times \int e \, dt$$

where g = gain

K_r = reset setting in repeats per minute.

K_r = 분당 반복 설정을 재설정합니다.

At any time the rate of change of the output is the gain times the reset rate times the error.

언제든지 출력의 변화율은 리셋 속도와 오류의 곱 시간입니다.

If the error is zero the output does not change;

오류가 0이면 출력이 변경되지 않습니다.

if the error is positive the output increases.

오류가 양수이면 출력이 증가합니다.

Shown below is an open loop trend of the error and output.

다음은 오류 및 출력의 개방 루프 추세입니다.

We would obtain this trend if we recorded the output of a controller that was not connected to a process while we manipulated the error.

오류를 조작하는 동안 프로세스에 연결되지 않은 컨트롤러의 출력을 기록하면 이 추세를 얻을 수 있습니다.

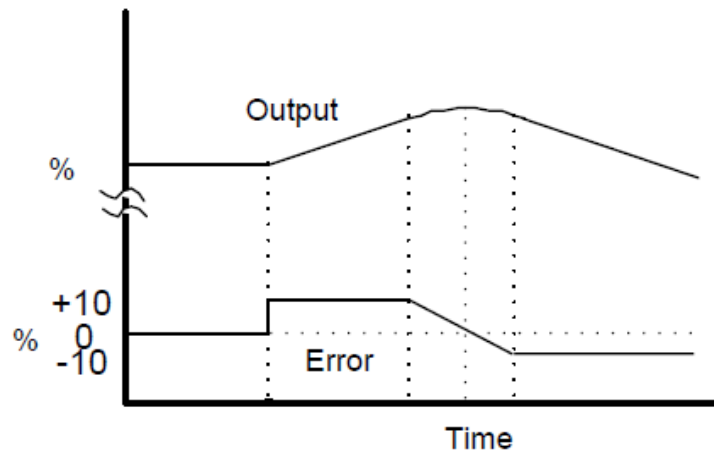


Figure 11 Output vs. error over time.

While the error is positive, the output ramps upward.

오류가 양수이면 출력이 위로 기울어집니다.

While the error is negative the output ramps downward.

오류가 음수 인 동안 출력은 아래쪽으로 기울어집니다.

2.10 CALCULATION OF REPEAT TIME

2.10 반복 시간의 계산

Most controllers use both proportional action (gain) and reset action (integral) together. The equation for the controller is:

대부분의 컨트롤러는 비례 동작 (게인)과 재설정 동작 (적분)을 함께 사용합니다.

컨트롤러의 방정식은 다음과 같습니다.

$$\text{Out} = g (e + K_r \int e \, dt)$$

where g = gain

K_r = reset setting in repeats per minute.

If we look an open loop trend of a PI controller after forcing the error from zero to some other value and then holding it constant, we will have the trends shown in .Figure 1

오류를 0에서 다른 값으로 강제 변경 한 후 PI 컨트롤러의 개방형 루프 추세를 살펴본 후 일정한 값을 유지한다면 추세를 볼 수 있습니다 (그림 1).

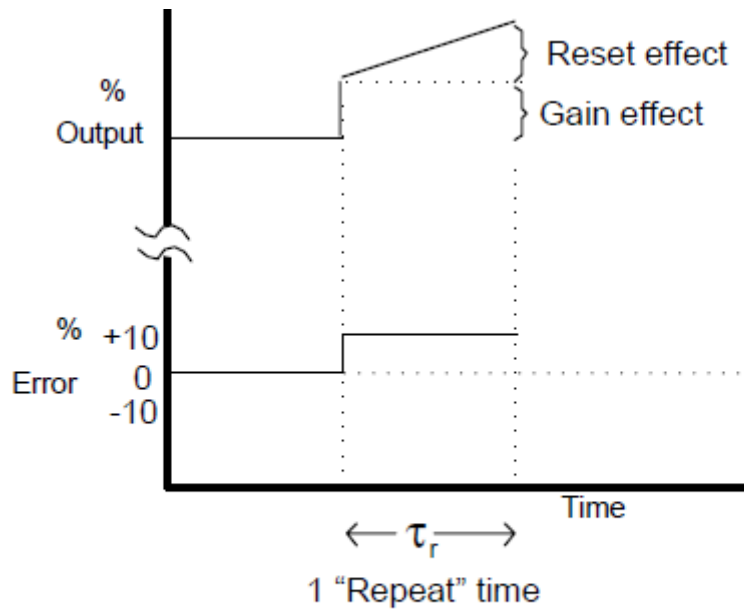


Figure 12 Calculation of repeat time

Figure 12 Calculation of repeat time

반복시간 계산

We can see two distinct effects of the change in the error.

오류의 변화로 인해 두 가지 별개의 효과를 볼 수 있습니다.

At the time the error changed the output also changed.

당시에 오류가 변경되어 출력도 변경되었습니다.

This is the "gain effect" and is equal to the product of the gain and the change in the error.

이것은 "이득(gain) 효과"이며 이득의 변화와 오차의 곱과 동일합니다.

The second effect (the "reset effect") is the ramp of the output due to the error.

두 번째 효과 ("재설정 효과")는 오류로 인한 출력의 상승입니다.

If we measure the time from when the error is changed to when the reset effect is equal to the gain effect we will have the "repeat time."

오류가 변한 때부터 리셋 효과가 게인 효과와 같을 때까지의 시간을 측정하면 "반복 시간"이 생깁니다.

Some control vendors measure reset by repeat time (or "reset time" or "integral time") in minutes.

일부 제어 공급 업체는 반복 시간 (또는 "재설정 시간"또는 "적분 시간")으로 분 단위로 재설정을 측정합니다.

Others measure reset by "repeats per minute."

다른 것들은 "분당 반복"으로 재설정을 측정합니다.

Repeats per minute is the inverse of minutes of repeat.

분당 반복은 반복 분의 역수 입니다.

The derivative contribution can be expressed mathematically:

미분 기여도는 수학적으로 표현 될 수 있습니다.

$$\text{Out} = g \times K_d \times \frac{de}{dt}$$

where g is gain,

여기서, g 는 이득이고,

K_d is the derivative setting in minutes, and

K_d 는 분 단위의 미분 설정입니다.

e is the error

e 는 오류입니다.

The open loop response of controller with proportional and derivative is shown graphically:

비례 및 미분 컨트롤러의 개방 루프 응답은 그래픽으로 표시됩니다.

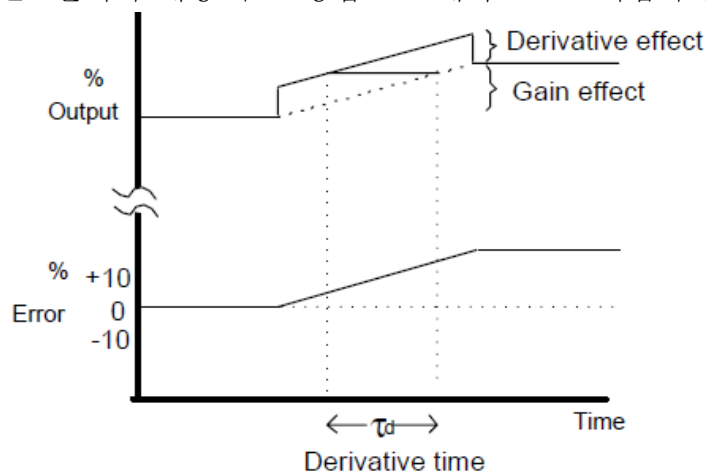


Figure 13 Output vs. error of derivative over time

The derivative advances the output by the amount of derivative time.

미분시간의 양만큼 출력이 미분의 advances(총액)가 진행된다.

This diagram compares the output of a controller with gain only (dashed line) with the output of a controller with gain and derivative (solid line).

이 다이어그램에서는 gain(이득) 만있는 컨트롤러의 출력 (점선)과 이득 및 미분 (실선)이있는 컨트롤러의 출력을 비교합니다.

The solid line is higher than the dashed line for the time that the process is increasing due the addition of the rate of change to the gain effect.

실선은 이득 효과에 대한 변화율의 추가로 인한 process가 증가하는 동안 점선보다 높다.

We can also look at the solid line as being “leading” the dashed line by some amount of time (τ_d).

우리는 또한 실선을 일정 시간 (τ_d) 동안 점선으로 "인도"하는 것으로 볼 수 있습니다.

The amount of time that the derivative action advances the output is known as the “derivative time” (or Preact time or rate time) and is measured in minutes.

미분 작업이 출력을 진행시키는 시간은 "미분 시간"(또는 Preact 시간 또는 속도 시간)으로 알려져 있으며 분 단위로 측정됩니다.

All major vendors measure derivative the same: in minutes.

모든 주요 공급은 미분 상품을 분 단위로 동일하게 측정합니다.

2.12 COMPLETE PID RESPONSE

완전한 PID응답

If we combine the three terms (Proportional gain, Integral, and Derivative) we obtain the complete PID equation.

비례, 적분및미분의 세가지 조건을 결합하면 완전한 PID방정식을 얻을 수있다.

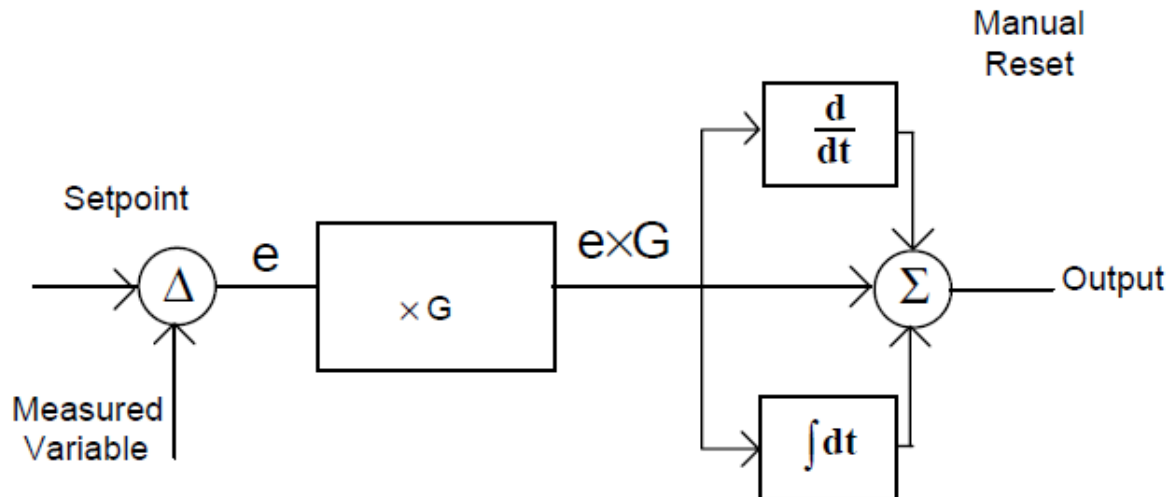


Figure 14 Combined gain, integral, and derivative elements.

This is a simplified version of the PID controller block diagram with all three elements, gain, reset, and derivative.

이건 gain, reset 및 미분요소의 세가지 요소가 모두 포함된 PID컨트롤러 블록다이어그램의 단순화 버전이다.

$$\text{Out} = G(e + R) \left[e \, dt + D \, \frac{de}{dt} \right]$$

Where

G = Gain

R = Reset (repeats per minute)

D = Derivative (minutes)

This is a general form of the PID algorithm and is close to, but not identical to, the forms actually implemented in industrial controllers.

이것은 PID 알고리즘의 일반적인 형태이며 산업 컨트롤러에 실제로 구현된 형태에 가깝지만 동일하지는 않습니다.

Modifications of this algorithm are described in the next chapter.

이 알고리즘의 수정은 다음 장에서 설명합니다.

2.13 RESPONSE COMBINATIONS

응답조합

Most commercial controllers allow the user to specify Proportional only controllers, proportional-reset (PI) controllers, and PID controllers that have all three modes.

대부분의 상용 제어기는 사용자가 비례 제어 (Proportional only) 제어기, 비례 재설정 (PI) 제어기 및 모든 세 가지 모드를 갖는 PID 제어기를 지정할 수 있도록합니다.

The majority of loops employ PI controllers.

대부분의 루프는 PI 컨트롤러를 사용합니다.

Most control systems also allow all other combinations of the responses: integral, integral-derivative, derivative, and proportional-derivative.

대부분의 제어 시스템은 적분, 적분 미분, 미분 및 비례 미분 값의 모든 다른 조합도 허용합니다.

When proportional response is not present the integral and derivative is calculated as if the gain were one.

비례 응답이 없으면 적분과 미분은 게인이 하나 인 것처럼 계산됩니다

#CHAPTER 3

IMPLEMENTATION DETAILS OF THE PID EQUATION

PID 등식의 구현 세부사항

The description of the PID algorithm shown on the previous page is a “text book” form of the algorithm.

이전 페이지에 나온 PID 알고리즘에 대한 설명은 "텍스트 북"형태의 알고리즘입니다.

The actual form of the algorithm used in most industrial controllers differs somewhat from the equation and diagram of shown on the previous page.

대부분의 산업용 컨트롤러에서 사용되는 알고리즘의 실제 형태는 이전 페이지에서 설명한 방정식 및 다이어그램과 다소 다릅니다.

3.1 SERIES AND PARALLEL INTEGRAL AND DERIVATIVE

시리즈와 병렬 통합 및 미분

The form of the PID equation shown , which is the way the PID is often represented in text books, differs from most industrial implementations in the basic structure.

제시된 PID 방정식의 형태는 PID가 텍스트 북에서 종종 표현되는 방식이며, 기본 구조의 대부분의 산업 구현과 다릅니다.

Most implementations place the derivative section in series with the integral or reset section. 대부분의 구현은 미분 섹션을 적분 또는 재설정 섹션과 직렬로 배치합니다.

We can modify the diagram shown above to reflect the series algorithm:

우리는 시리즈 알고리즘을 반영하기 위해 위의 다이어그램을 수정할 수 있습니다 :

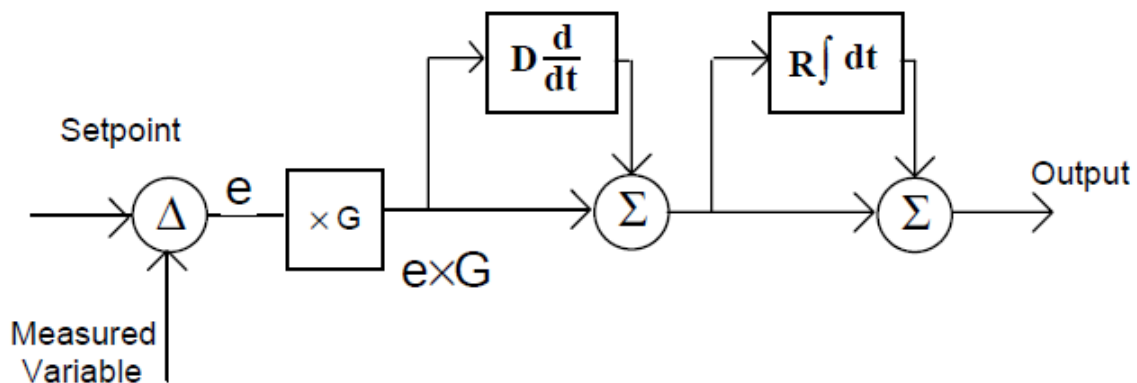


Figure 15 The series form of the complete PID response.

The difference between this implementation and the parallel one is that the derivative has an effect on the integration.

이 구현과 병렬구현의 차이점은 미분이 적분에 영향을 미친다는 점이다.

The equation becomes:

방정식은 다음과 같다.

$$\text{Out} = (RD+1)G(\text{Error} + R(RD+1)) \int \text{edt} + D(RD+1) \frac{d\text{edt}}{dt}$$

where R = the reset rate in repeats per minute,

여기서 R = 분당 반복 수의 리셋 율,

D = the derivative in minutes,

D = 분 단위의 미분,

and G = the gain.

The effect is to increase the gain by a factor of $RD + 1$, while reducing the reset rate and derivative time by the same factor.

그 효과는 리셋 속도와 미분 시간을 같은 요소로 줄이면서 $RD + 1$ 의 계수만큼 이득을 증가시키는 것입니다.

Based on common tuning methods, the derivative time is usually no more than about $\frac{1}{4}$ the reset time ($1/R$), therefore the factor $RD+1$ is usually 1.25 or less.

일반적인 튜닝 방법에 따라 파생 시간은 일반적으로 리셋 시간 ($1 / R$)의 약 1/4에 불과하므로 $RD + 1$ 팩터는 일반적으로 1.25 이하입니다.

Almost all analog controllers and most commercial digital control systems use the series form.

거의 모든 아날로그 컨트롤러와 대부분의 상용 디지털 제어 시스템은 시리즈 형식을 사용합니다.

Such tuning methods as the Ziegler-Nichols methods (discussed in Chapter 6) were developed using series form controllers.

Ziegler-Nichols 방법 (6 장에서 논의 됨)과 같은 튜닝 방법은 직렬 폼 컨트롤러를 사용하여 개발되었습니다.

Unless derivative is used there is no difference between the parallel (non-interactive) and series (interactive) forms.

미분을 사용하지 않는 한 병렬 (비대화 형) 및 시리즈 (대화식) 형식간에 차이가 없습니다.

3.2 GAIN ON PROCESS RATHER THAN ERROR

에러는 process에서 보다 많은 gain을 얻는다.

The gain causes the output to change by an amount proportional to the change in the error.
게인은 출력 변화를 에러의 변화에 비례하여 변화시킵니다.

Because the error is affected by the set point, the gain will cause any change in the set point to change the output.

오류가 설정 점의 영향을 받기 때문에 게인은 설정 점의 변경으로 인해 출력이 변경됩니다.

This can become a problem in situations where a high gain is used where the set point may be suddenly changed by the operator, particularly where the operator enters a new set point into a CRT.

이것은 특히 operator가 CRT에 새로운 설정 점을 입력 할 때 설정 점이 갑자기 변경 될 수있는 높은 게인이 사용되는 상황에서 문제가 될 수 있습니다.

(*CRT는 음극선관이라고 불리며, 진공관으로 되었고 영상을 표시하는데 사용된다)

This will cause the set point, and therefore the output, to make a step change.

이렇게하면 설정 점과 출력이 단계적으로 변경됩니다.

In order to avoid the sudden output change when the operator changes the set point of a loop, the gain is often applied only to the process.

작업자가 루프의 설정 값을 변경할 때 갑자기 출력이 변경되는 것을 피하기 위해 이득(gain)은 종종 process에만 적용됩니다.

Set point changes affect the output due to the loop gain and due to the reset, but not due to the derivative.

설정 값 변경은 루프 게인 및 리셋으로 인해 출력에 영향을 미치지 만 미분으로 인해 영향을받지는 않습니다.

3.3 DERIVATIVE ON PROCESS RATHER THAN ERROR

에러는 process에서 보다는 많은 미분을 한다.

The derivative acts on the output by an amount proportional to the rate of change of the error.

미분은 오류의 변화율에 비례하는 양만큼 출력에 작용합니다.

Because the error is affected by the set point, the derivative action will be applied to the change in the set point.

오차는 설정 점의 영향을 받기 때문에 미분 동작이 설정 점의 변화에 적용됩니다.

This can become a problem in situations where the set point may be suddenly changed by the operator, particularly in situations where the operator enters a new set point into a CRT.

이는 특히 운전자가 CRT에 새로운 설정 점을 입력하는 상황에서 설정 점이 작업자에 의해 갑자기 변경되는 상황에서는 문제가 될 수 있습니다.

This causes the set point to have a step change.

이로 인해 설정 점이 단계 변경됩니다.

Applying derivative to a step change, even a small step change, will result in a “spike” on the output.

작은 스텝 변경에서도 스텝 변경에 미분을 적용하면 출력에 '스파이크'가 발생합니다.

In order to avoid the output spike when the operator changes the set point of a loop, the derivative is often applied only to the process.

작업자가 루프의 설정 값을 변경할 때 출력 스파이크를 피하기 위해 미분은 종종 process에만 적용됩니다.

Set point changes affect the output due to the loop gain and due to the reset, but not due to the derivative.

설정 값 변경은 루프 게인 및 리셋으로 인해 출력에 영향을 미치지 만 파생으로 인해 영향을받지는 않습니다.

Most industrial controllers offer the option of derivative on process or derivative on error.

대부분의 산업용 컨트롤러는 오류시 프로세스 또는 파생에서 파생옵션을 제공합니다.

3.4 DERIVATIVE FILTER

미분필터

The form of derivative implemented in controllers also includes filtering.

컨트롤러에 구현 된 파생물 형태에는 필터링도 포함됩니다.

The filter differs among the various manufactures.

필터는 다양한 제조업체에 따라 다릅니다.

A typical filter comprises two first order filters that follow the derivative.

전형적인 필터는 미분을 따르는 2 개의 1 차 필터를 포함한다.

The time constant of the filters depends upon the derivative time and the scan rate of the loop.

필터의 시간 상수는 미분 시간 및 루프의 스캔 속도에 따라 달라집니다.

3.5 COMPUTER CODE TO IMPLEMENT THE PID ALGORITHM

PID알고리즘을 구현하기 위한 컴퓨터 코드

There are many ways to implement the PID algorithm digitally.

디지털 방식으로 PID 알고리즘을 구현하는 많은 방법이 있습니다.

Two will be discussed here. In each case, there will be a section of code (in structured Basic, easily convertible to any other language) that will be executed by the processor every second.

여기서 두 가지를 논의 할 것입니다. 각각의 경우에 매 초마다 프로세서에서 실행되는 코드 섹션 (구조화 된 기본 언어로, 다른 언어로 쉽게 변환 가능)이 있습니다.

(some other scan rate may be used, change the constant 60 to the number of times per minute it is executed.)

(다른 스캔 속도가 사용될 수도 있습니다. 상수 60을 실행되는 분당 횟수로 변경하십시오.)

In each code sample there is an IF statement to execute most of the code if the loop is in the auto mode.

각 코드 샘플에는 루프가 자동 모드 인 경우 대부분의 코드를 실행하는 IF 문이 있습니다.

If the loop is in manual mode only a few lines are executed in order to allow for bumpless transfer to auto.

루프가 수동 모드 인 경우 뾰족없는 자동 전송을 허용하기 위해 몇 줄만 실행됩니다.

Also, while the control loop is in manual, the output (variable OutP) will be operator adjustable using the operator interface software.

또한 제어 루프가 수동으로 작동하는 동안 출력 (변수 OutP)은 operator 인터페이스 소프트웨어를 사용하여 조정 가능한 operator가됩니다.

3.5.1 Simple PID code

간단한 PID 코드

One method of handling the integration and bumpless transfer to automatic mode is an algorithm that calculates the change in output from one pass to the next using the derivative of the PID algorithm, or:

자동 모드로의 통합 및 충돌없는 전송을 처리하는 한 가지 방법은 PID 알고리즘의 미분을 사용하여 한 패스에서 다음 패스로의 출력 변화를 계산하는 알고리즘입니다.

$$\frac{dOut}{dt} = Gain \times \left(ResetRate \times Error + Derivative \times \frac{d^2 Error}{dt^2} \right)$$

This program is run every second. If the control loop is in manual, the output is adjusted by the operator through the operator interface software.

이 프로그램은 매초마다 실행됩니다. 제어 루프가 수동 상태 인 경우 출력은 operator 인터페이스 소프트웨어를 통해 operator가 조정합니다.

If the control loop is in Automatic, the output is computed by the PID algorithm.

제어 루프가 자동이면 출력은 PID 알고리즘에 의해 계산됩니다.

Each pass the output is changed by adding the change in output to the previous pass output. That change is found by adding:

각 패스는 이전 패스 출력에 대한 출력 변경을 추가하여 출력을 변경합니다. 그 변화는 다음을 추가함으로써 발견된다.

- the change in error (Err-ErrLast)
오류의 변화 (Err-ErrLast)
- the error multiplied by the reset rate, and
오류에 재설정 률을 곱한 값
- the second derivative of the error
(Err-2*ErrLast+ErrLastLast) times the derivative.
오류의 2 차 미분
(Err-2 * ErrLast + ErrLastLast) 곱하기 미분.

The total is then multiplied by the gain.
합계에 게인이 곱해집니다.

This simple version of the PID controller work well in most cases, and can be tuned by the standard PID tuning methods (some of which are discussed later).

PID 컨트롤러의 이 간단한 버전은 대부분의 경우 잘 작동하며 표준 PID 튜닝 방법으로 튜닝 할 수 있습니다 (일부는 나중에 설명합니다).

It has "Parallel" rather than "Series" reset and derivative, and derivative is applied to the error rather than the input only.

"시리즈"재설정 및 과생물이 아닌 "병렬"이 있으며 미분은 입력이 아닌 오류에만 적용됩니다.

The only serious problem with this form of the algorithm occurs when the output has reached an upper or lower limit.

이 알고리즘의 유일한 심각한 문제는 출력이 상한 또는 하한에 도달했을 때 발생합니다.

When it does, a change in the measurement can unexpectedly pull the output away from the limit.

이 경우 측정 값이 변경되면 예기치 않게 출력이 한계에서 벗어날 수 있습니다.

For example, Figure 16 illustrates the set point, measurement, and output of an open loop direct acting controller with a high gain and slow reset.

예를 들어, 그림 16은 높은 이득과 느린 리셋이있는 개방 루프 직결 컨트롤러의 설정치, 측정 및 출력을 보여줍니다.

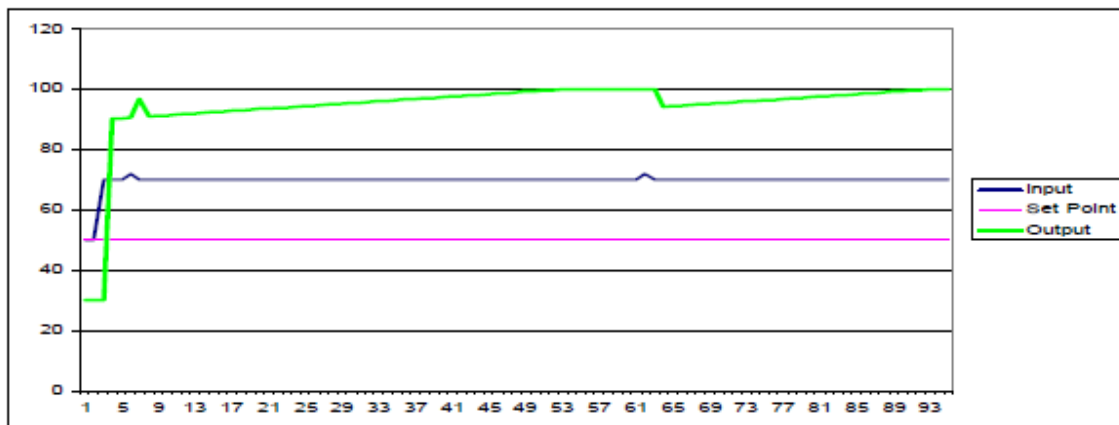


Figure 16 - Effect of input spike

When the input (blue line) rises above the set point (red line) the output (green line) first increases due to the proportional response and then continues to ramp up due to the reset response.

입력 (파란색 선)이 설정 점 (빨간색 선) 이상으로 상승하면 출력 (녹색 선)은 먼저 비례 응답으로 인해 증가하고 재설정 응답으로 인해 계속 상승합니다.

The ramp ends when the output is limited at 100%.

램프 출력은 100 %로 제한되면 끝납니다.

Note the spike (noise) in the input at about 5 minutes.

약 5 분 후에 입력의 스파이크 (잡음)를 확인하십시오.

That spike results in a spike in the output, in the same direction.

그러한 스파이크는 동일한 방향으로 산출량을 급증시킵니다.

Compare this with the similar spike at about 61 minutes.

이것을 약 61 분에 비슷한 스파이크와 비교하십시오.

Rather than cause an upward output spike as expected, the spike causes the output to pull away from the upper limit.

예상대로 상승 출력 스파이크를 유발하지 않고, 스파이크는 출력을 상한선에서 끌어 당깁니다.

It slowly ramps back to the limit.

천천히 다시 한도로 기울입니다.

This is because the limit blocks the leading (increasing) side of the spike but does nothing to the trailing (decreasing) side of the spike.

이것은 한계가 스파이크의 앞쪽 (증가) 쪽을 차단하지만 스파이크의 뒤쪽 (감소) 쪽에 아무 것도하지 않기 때문입니다.

When an input noise spike occurs while the output is below the limit, it causes the output to spike upwards.

출력이 한계보다 낮을 때 입력 잡음 스파이크가 발생하면 출력이 위쪽으로 급상승합니다.

When the same spike occurs while the output is limited, the spike causes the output to pull away from the limit.

출력이 제한되어있는 동안 동일한 스파이크가 발생하면 스파이크가 출력을 제한에서 벗어나게합니다.

3.5.2 Improved PID code

개선된 PID 코드

The method of implementing automatic reset described in section 2.8, using a positive feedback loop (see Figure 10) was first used with pneumatic analog controllers.

포지티브 피드백 루프 (그림 10 참조)를 사용하여 2.8 절에 설명 된 자동 리셋 구현 방법은 공압식 아날로그 컨트롤러에서 처음 사용되었습니다.

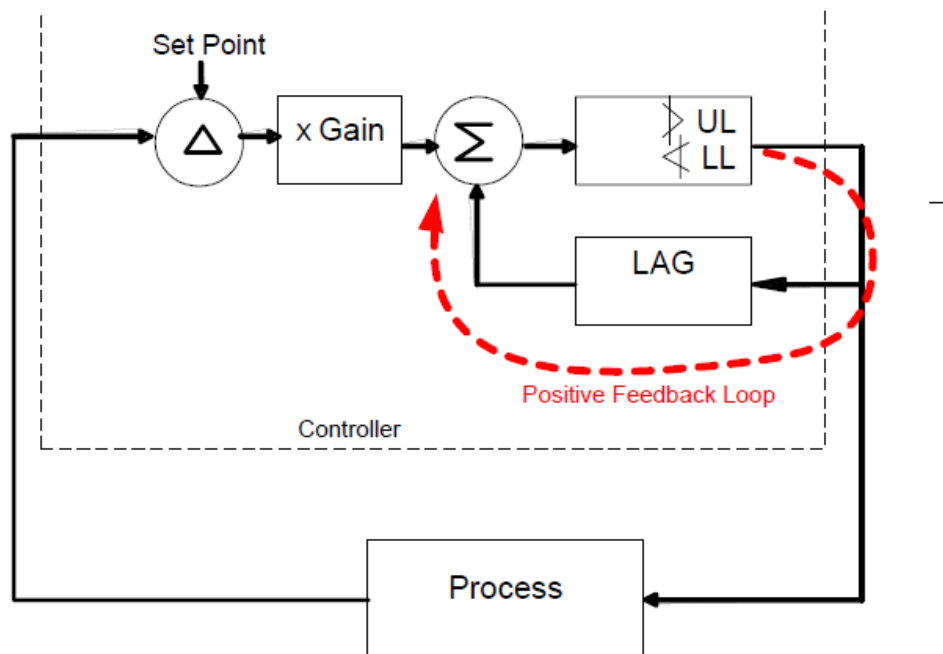


Figure 10 Addition of automatic reset to a proportional controller

It can easily be implemented digitally.

디지털 방식으로 쉽게 구현할 수 있습니다.

There are several advantages of this algorithm implementation.

이 알고리즘 구현의 몇 가지 장점이 있습니다.

Most important, it eliminates the problems that cause the output to pull away from a limit inappropriately.

가장 중요한 것은 출력이 부적절하게 한도에서 벗어나는 문제를 제거합니다.

It also allows the use of external feedback when required.

또한 필요한 경우 외부 피드백을 사용할 수 있습니다.

Variables:

Input	<i>Process input</i>
InputD	<i>Process input plus derivative</i>
InputLast	<i>Process input from last pass, used in deriv. calc.</i>
Err	<i>Error, Difference between input and set point</i>
SetP	<i>Set point</i>
OutPutTemp	<i>Temporary value of output</i>
OutP	<i>Output of PID algorithm</i>
Feedback	<i>Result of lag in positive feedback loop.</i>
Mode	<i>value is 'AUTO' if loop is in automatic</i>
Action	<i>value is 'DIRECT' if loop is direct acting</i>
Gain	<i>value of Proportional Gain tuning parameter, dimensionless</i>
ResetRate	<i>value of Reset or Integral tuning parameter, repeats per minute</i>
Derivative	<i>value Derivative tuning parameter, minutes</i>

The PID emulation code:

```
1.  IF Mode = 'AUTO' THEN
2.      InputD=Input+(Input-InputLast)*Derivative*60  derivative.
3.      InputLast = Input
4.      Err= SetP - InputD  Error based on reverse action.
5.      IF Action = 'DIRECT' THEN Err=0 - Err  Change sign if direct.
6.      ENDIF
7.      OutPutTemp = Err*Gain+Feedback  Calculate the gain time the error and add the feedback.
8.      IF OutPutTemp > 100 THEN OutPutTemp =100  Limit output to between
9.      IF OutPutTemp < 0 THEN OutPutTemp =0  0 and 100 percent.
10.     OutP = OutPutTemp  The final output of the controller.
11.     Feedback=Feedback+ (OutP-Feedback) *ResetRate/60
12. ELSE
13.     InputLast=Input  While loop in manual, stay ready for bumpless switch to Auto.
14.     Feedback=OutP
15. ENDIF
```

외부 피드백이 사용되면 11 행의 변수 "OutP"가 외부 피드백을 포함하는 변수로 바뀝니다.

#CHAPTER 4

ADVANCED FEATURES OF THE PID ALGORITHM

PID 알고리즘의 고급 기능

4.1 RESET WINDUP

상한 리셋

One problem with the reset function is that it may “wind up”.

리셋 기능의 한 가지 문제점은 "와인드업 (wind up)"될 수 있다는 것입니다.

Because of the integration of the positive feedback loop, the output will continue to increase or decrease as long as there is an error (difference between set point and measurement) until the output reaches its upper or lower limit.

긍정적인 피드백 루프가 통합되어 있기 때문에 출력이 상한 또는 하한에 도달 할 때까지 오류 (설정 점과 측정 사이의 차이)가있는 한 출력은 계속 증가 또는 감소합니다.

This normally is not a problem and is a normal feature of the loop.

이것은 일반적인 문제가 아니고 루프의 정상적인 기능입니다.

For example, a temperature control loop may require that the steam valve be held fully open until the measurement reaches the set point.

예를 들어, 온도 제어 루프는 측정이 설정치에 도달 할 때까지 스팀 밸브가 완전히 열리도록 요구할 수 있습니다.

At that point, the error will be cross zero and change signs, and the output will start decreasing, “throttling back” the steam valve.

이 시점에서 오류는 십자 기호 0이되어 부호가 바뀌고 출력이 감소하기 시작하여 스팀 밸브가 "스로틀 링 (throttling back)"됩니다.

Sometime, however, reset windup may cause problems.

그러나 때때로 언 와인드 리드가 문제를 일으킬 수 있습니다.

Actually, the problem is not usually the windup but the “wind down” that is then be required.

사실, 문제는 일반적으로 와인드업이 아니지만 "wind down"이 요구됩니다.

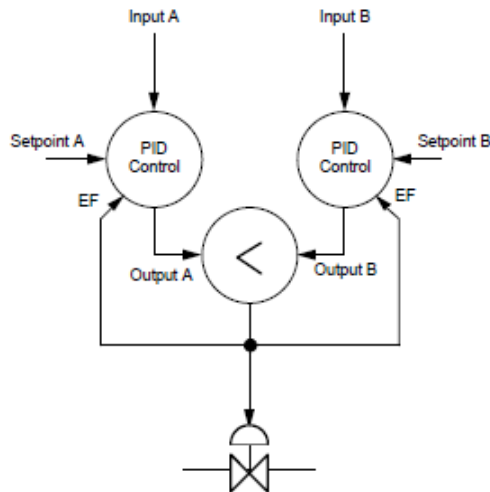


Figure 17 Two PID controllers that share one valve.

Suppose the output of a controller is broken by a selector, with the output of another controller taking control of the valve.

컨트롤러의 출력이 선택기에 의해 끊어지고 다른 컨트롤러의 출력이 밸브를 제어한다고 가정합니다.

In the diagram the lower of the two controller outputs is sent to the valve.

다이어그램에서 두 개의 컨트롤러 출력 중 낮은 쪽이 밸브로 보내집니다.

Which ever controller has the lower output will control the valve.

어느 컨트롤러가 낮은 출력을 가지면 밸브를 제어 할 것입니다.

The other controller is, in effect, open loop.

다른 컨트롤러는 사실상 개방 루프입니다.

If its error would make its output increase, the reset term of the controller will cause the output to increase until it reaches its limit.

오류가 출력을 증가 시키면 제어기의 재설정 기간은 출력이 한계에 도달 할 때까지 증가시킵니다.

The problem is that when conditions change and the override controller no longer needs to hold the valve closed the primary controller's output will be very far above the override signal.

문제는 조건이 변경되고 오버라이드 컨트롤러가 더 이상 밸브를 닫지 않아도 될 때 기본 컨트롤러의 출력이 오버라이드 신호보다 훨씬 높다는 것입니다.

Before the primary controller can have any effect on the valve, it will have to "wind down" until its output equals the override signal.

주 컨트롤러가 밸브에 영향을 미치기 전에 출력이 오버라이드 신호와 같아 질 때까지 "감속"해야 합니다.

4.2 EXTERNAL FEEDBACK

외부 피드백

The positive feedback loop that is used to provide integration can be brought out of the controller.

integration을 제공하는 데 사용되는 긍정적인 피드백 루프는 컨트롤러 밖으로 가져올 수 있습니다.

Then it is known as external feedback:

그런것을 외부 피드백이라고합니다.

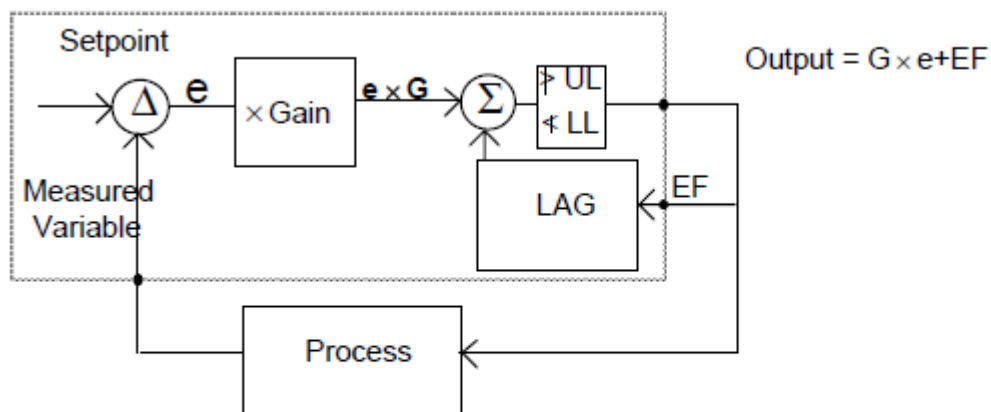


Figure 18 A proportional-reset loop with the positive feedback loop used for integration.

Figure 18 A proportional-reset loop with the positive feedback loop used for integration.

그림 18 적분에 사용되는 긍정적인 피드백 루프가있는 비례 리셋 루프이다.

If there is a selector between the output of the controller and the valve (used for override control) the output of the selector is connected to the external feedback of the controller.

컨트롤러의 출력과 밸브 (오버라이드 제어에 사용되는) 사이에 선택기가있는 경우 선택기의 출력이 컨트롤러의 외부 피드백에 연결됩니다.

This puts the selector in the positive feedback loop.

그러면 선택기가 긍정적인 피드백 루프에 놓입니다.

If the output of the controller is overridden by another signal, the overriding signal is brought into the external feedback.

컨트롤러의 출력이 다른 신호에 의해 무시되면, 오버라이드 신호는 외부 피드백으로 보내집니다.

After the lag, the output of the controller is equal to the override signal plus the error times gain.

지체 후 컨트롤러의 출력은 오버라이드 신호에 오류 시간 계인을 더한 것과 같습니다.

Therefore, when the error is zero, the controller output is equal to the override signal.
따라서 오류가 0 일 때 컨트롤러 출력은 오버라이드 신호와 동일합니다.

If the error becomes negative, the controller output is less than the override signal, so the controller regains control of the valve.

오류가 음수가되면 컨트롤러 출력이 오버라이드 신호보다 작으므로 컨트롤러가 밸브를 다시 제어합니다.

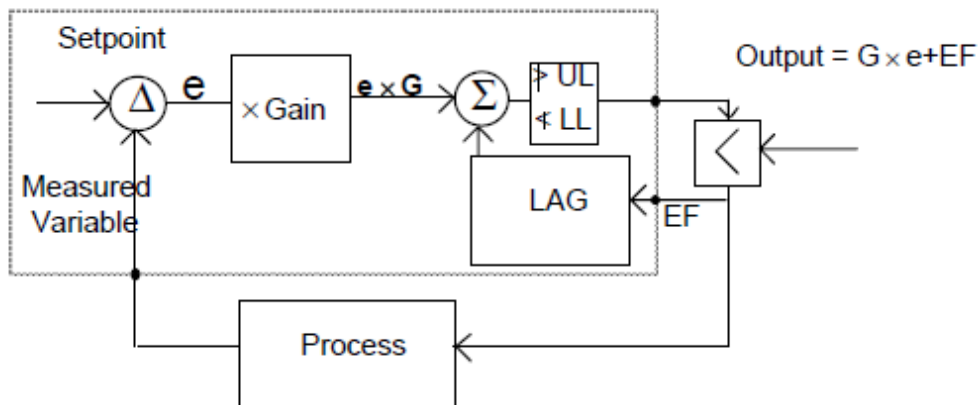


Figure 19 The external feedback is taken from the output of the low selector.

4.3 SET POINT TRACKING

설정점 트래킹

If a loop is in manual and the set point is different from the process value, when the loop is switched to auto the output will start moving, attempting to move the process to the set point, at a rate dependent upon the gain and reset rate.

루프가 수동으로 설정 값이 프로세스 값과 다른 경우 루프가 자동으로 전환되면 출력이 움직이기 시작하여 이득 및 재설정 속도에 따라 속도가 설정 점으로 이동합니다 .

Take for example a typical flow loop, with a gain of 0.6 and a reset rate of 20.

예를 들어 0.6의 게인 및 20의 리셋 율을 갖는 전형적인 플로우 루프를 예로들 수 있습니다.

If difference between the set point and process is 50% at the time the loop is switched to automatic the output will ramp at a rate of 10%/second.

루프가 자동으로 전환 될 때 설정 점과 프로세스 사이의 차이가 50 %이면 출력은 초당 10 %의 속도로 증가합니다.

Often, when a loop has been in manual for a period of time the value of the set point is meaningless.

종종 루프가 일정 시간 동안 수동 상태에 있었을 때 설정 값의 값은 의미가 없습니다.

It may have been the correct value before a process upset or emergency shutdown caused the operator to place the loop into manual and change the process operation.

process의 혼란이나 비상 정지로 인해 작업자가 루프를 수동으로 배치하고 process 작동을 변경하기 전에 올바른 값이었을 수 있습니다.

On a return to automatic the previous set point value may have no meaning.

자동으로 되돌아 가면 이전 설정치 값은 의미가 없을 수 있습니다.

However, to prevent a process upset the operator must change the set point to the current process measurement before switching the loop from manual to automatic.

그러나 프로세스가 뒤집히는 것을 방지하려면 루프를 수동에서 자동으로 전환하기 전에 운영자가 설정 포인트를 현재 process 측정으로 변경해야 합니다.

Some industrial controllers offer a feature called “set point tracking” that causes the set point to track the process measurement when the loop is not in automatic control.

일부 산업용 컨트롤러는 루프가 자동 제어 상태가 아닐 때 세트 포인트가 process 측정을 추적하도록 하는 "세트 포인트 트래킹"기능을 제공합니다.

With this feature when the operator switches from manual to automatic the set point is already equal to the process, eliminating any bump in the process.

이 기능을 사용하면 작업자가 수동에서 자동으로 전환 할 때 설정 점이 이미 process와 동일하므로 process의 충동을 제거 할 수 있습니다.

The set point tracking feature is typically used with loops that are tuned for fast reset, where a change from manual to automatic could cause the output to rapidly move, and for loops where the set point is not always the same.

설정 점 추적 기능은 일반적으로 빠른 재설정을 위해 조정 된 루프와 함께 사용되며 수동에서 자동으로 변경되면 출력이 빠르게 이동하고 설정 점이 항상 동일하지 않은 루프가 발생합니다.

For example, the temperature of a room or an industrial process usually should be held to some certain value.

예를 들어, 방의 온도 또는 산업 공정은 보통 특정 값으로 유지되어야 합니다.

The set point for the rate of flow of fuel to the heater is “whatever it takes” to maintain the temperature at its set point.

히터로의 연료 흐름 속도에 대한 설정 점은 설정 점에서 온도를 유지하기 위해 "무엇이든간에"입니다.

Therefore flow loops are more likely to use set point tracking.

따라서 플로우 루프는 셋 포인트 추적을 사용하는 경향이 더 큽니다.

However, this is a judgment that must be made by persons knowledgeable in the operation of the process.

그러나 이것은 process 운영에 지식이있는 사람들이 판단해야 합니다.

#CHAPTER 5

PROCESS RESPONSES

프로세스 응답

Loops are tuned to match the response of the process.

루프는 프로세스의 응답과 일치하도록 조정됩니다.

In this chapter we will discuss the responses of the process to the control system.

이 장에서는 제어 시스템에 대한 프로세스의 응답에 대해 설명합니다.

The dynamic and steady state response of the process signal to changes in the controller output.

제어기 출력의 변화에 대한 프로세스 신호의 동적 및 정상 상태 응답.

These responses are used to determine the gain, reset, and derivative of the loop.

이러한 응답은 루프의 이득, 재설정 및 파생을 결정하는 데 사용됩니다.

While discussing single loop control, we will consider the process response to be the effect on the controlled variable cause by a change in the manipulated variable (controller output).

단일 루프 제어에 대해 논의하는 동안, 우리는 조작 된 변수 (컨트롤러 출력)의 변경으로 인해 프로세스 응답이 제어 변수에 미치는 영향을 고려할 것입니다.

5.1 STEADY STATE RESPONSE

안정적인 상태의 응답

The steady state process response to controller output changes is the condition of the process after sufficient time has passed so that the process has settled to new values.

컨트롤러 출력 변화에 대한 정상 상태 프로세스 응답은 프로세스가 새로운 값으로 안정화되도록 충분한 시간이 지난 후 이다.

The steady state response of the process to the controller output is characterized primarily by process action, gain, and linearity.

제어기 출력에 대한 프로세스의 정상 상태 응답은 주로 프로세스 동작, 이득 및 선형성으로 특징 지워집니다.

5.1.1 Process Action

프로세스 작업

Action describes the direction the process variable changes following a particular change in the controller output.

동작은 컨트롤러 출력의 특정 변경에 따라 프로세스 변수가 변경되는 방향을 설명합니다.

A direct acting process increases when the final control element increases (typically, when the valve opens);

직접 형 프로세스는 최종 제어 요소가 증가 할 때 증가합니다 (일반적으로 밸브가 열리는 경우).

a reverse acting process decreases when the final control element increases.

최종 제어 요소가 증가 할 때 역 작용 프로세스가 감소한다.

For example, if we manipulate the inlet valve on a tank to control level, an increase in the valve position will cause the level to rise.

예를 들어, 탱크의 입구 밸브를 조작하여 레벨을 조절하면 밸브 위치가 증가하면 레벨이 상승합니다.

This is a direct acting process.

이것은 직접 행동 과정입니다.

On the other hand, if we manipulate the discharge valve to control the level, opening the valve will cause the level to fall.

반대로, 배출 밸브를 조작하여 레벨을 조절하면 밸브를 열면 레벨이 떨어집니다.

This is a reverse acting process.

이것은 역행 과정입니다.

5.1.2 Process Gain

프로세스 gain

Next to action, process gain is the most important process characteristic.

action 이후, 프로세스 이득이 가장 중요한 프로세스 특성입니다.

The process gain (not to be confused with controller gain) is the sensitivity of the controlled variable to changes in a controller output.

프로세스 게인 (컨트롤러 게인과 혼동하지 말 것)은 컨트롤러 출력의 변화에 대한 제어 변수의 민감도입니다.

Gain is expressed as the ratio of change in the process to the change in the controller output that caused the process change.

게인은 프로세스 변경의 원인이 된 컨트롤러 출력의 변화에 대한 프로세스 변경의 비율로 표시됩니다.

From the standpoint of the controller, gain is affected by the valve itself, by the process, and by the measurement transmitter.

제어기의 관점에서, 이득은 밸브 자체, process 및 측정 트랜스미터에 의해 영향을받습니다.

Therefore the size of the valve and the span of the transmitter will affect the process gain.

따라서 밸브의 크기와 트랜스미터의 스패는 process 이득에 영향을 미칩니다.

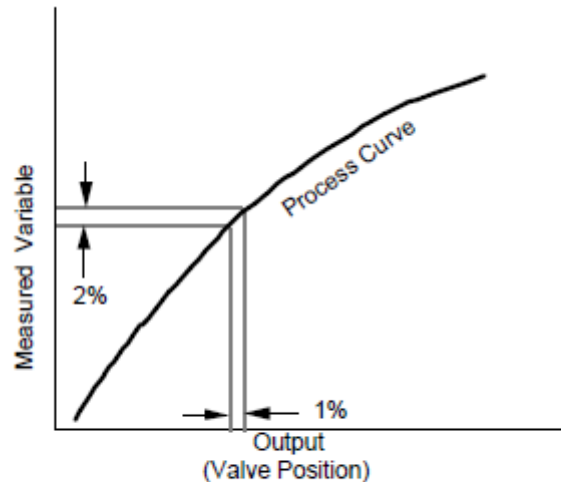


Figure 20 The direct acting process with a gain of 2.

In Figure 20 a 1% increase in the controller output causes the measured variable to increase by 2% .

그림 20에서 컨트롤러 출력이 1 % 증가하면 측정 변수가 2 % 증가합니다.

Therefore the process is direct acting and has a process gain of two.

따라서 프로세스는 직접 작용이며 프로세스 이득은 2입니다.

5.1.3 Process Linearity

process 선형성

The gain of the process often changes based on the value of the controller output.

프로세스의 이득은 종종 컨트롤러 출력의 값에 따라 변경됩니다.

That is, with the output at one value, a small change in the output will result in a larger change in the process measurement than the same output change at some other output value.

즉, 하나의 값에서 출력을 사용하면 출력에서 약간의 변화가 발생하면 다른 출력 값에서 동일한 출력 변화보다 프로세스 측정에서 더 큰 변화가 발생합니다.

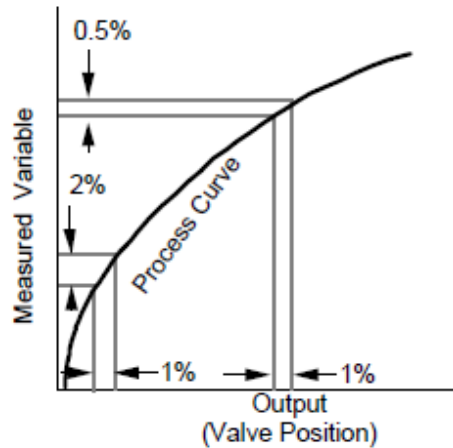


Figure 21 A non-linear process.

The process shown in Figure 21 is non linear.

그림 21에 표시된 프로세스는 비선형적이다.

With controller output very low, a 1% increase in the output causes the measured variable to increase by 2%.

제어기 출력이 매우 낮 으면 출력이 1 % 증가하면 측정 변수가 2 % 증가합니다.

-읽기시작-----

When the output is very high, the same 1% output increase causes the process to increase by only 0.5%.

출력이 매우 높으면 동일한 1 % 출력 증가로 인해 프로세스가 0.5 % 만 증가합니다.

The process gain decreases when the output increases.

출력이 증가하면 프로세스 이득이 감소합니다.

From the standpoint of controller tuning, the process linearity includes the linearity of the process, the final control element, and the measurement.

컨트롤러 튜닝의 관점에서 프로세스 선형성에는 프로세스의 선형성, 최종 제어 요소 및 측정이 포함됩니다.

It also includes any control functions between the PID algorithm and the output to the valve.

또한 PID 알고리즘과 밸브 출력 사이의 모든 제어 기능을 포함합니다.

5.1.4 Valve Linearity

밸브 선형성

Valves may be linear or non-linear.

밸브는 선형 또는 비선형 일 수 있습니다.

A linear valve is one in which the flow through the valve is exactly proportional to the position of the valve (or the signal from the control system).

선형 밸브는 밸브를 통과하는 유량이 밸브의 위치 (또는 제어 시스템의 신호)에 정확하게 비례하는 밸브입니다.

Valves may fall into three classes (illustrated in): linear, equal percentage, and quick opening.

밸브는 3 가지 등급으로 분류 될 수 있습니다 (그림 참조):선형, 동등한 비율 및 빠른 개방.

Linear valves have the same gain regardless of the valve position.

선형 밸브는 밸브 위치에 관계없이 동일한 게인을 갖습니다.

That is, at any point a given increase in the valve position will cause the same increase in the flow as at any other point.

즉, 임의의 지점에서 밸브 위치의 주어진 증가는 임의의 다른 지점에서와 동일한 흐름의 증가를 유발할 것이다.

Equal percentage valves have a low gain when the valve is nearly closed, and a higher gain when the valve is nearly open.

동등한 백분율 밸브는 밸브가 거의 닫히면 낮은 게인을, 밸브가 거의 열려있을 때는 더 높은 게인을 갖습니다.

Quick opening valves have a high gain when the valve is nearly closed and a lower gain when the valve is nearly open.

빠른 개방 밸브는 밸브가 거의 닫혀있을 때 높은 게인을 갖고 밸브가 거의 열려있을 때는 낮은 게인을 갖습니다.

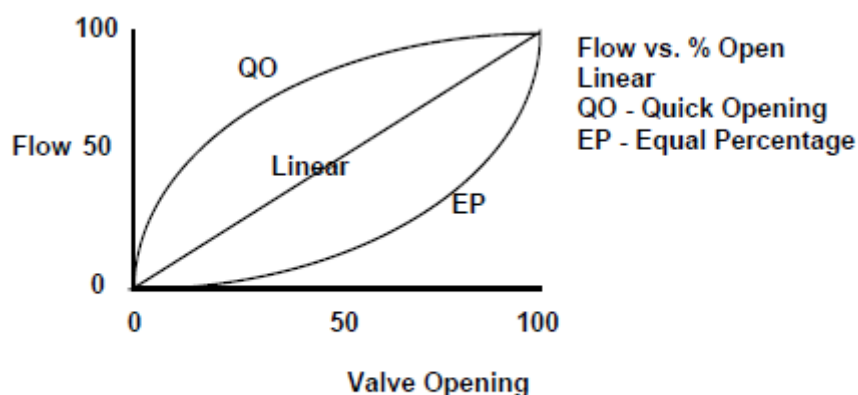


Figure 22 Types of valve linearity.

5.1.5 Valve Linearity: Installed characteristics

밸브 선형성 : 설치된 특성

Even a linear valve does not necessarily exhibit linear characteristics when actually installed in a process.

선형 밸브조차도 공정에 실제로 설치 될 때 반드시 선형 특성을 나타내지는 않습니다.

The characteristics described in the previous section are based on a constant pressure difference across the flanges of the valve.

이전 섹션에서 설명한 특성은 밸브의 플랜지에서 일정한 압력 차이를 기반으로 합니다.

However, the pressure difference is not necessarily constant.

그러나 압력 차는 반드시 일정하지는 않습니다.

When the pressure is a function of valve position, the actual characteristics of the valve are changed.

압력이 밸브 위치의 함수 인 경우, 밸브의 실제 특성이 변경됩니다.

Take for example the flow through a pipe and valve combination shown in Figure 23.

예를 들어 그림 23에 표시된 파이프와 밸브 조합을 통해 흐름을 가져옵니다.

Liquid flows from a pump with constant discharge pressure to the open air.

액체는 일정한 토출 압력의 펌프에서 외기로 흐릅니다.

There is a pressure drop through the valve that is proportional to the square of the flow.

밸브의 압력 강하는 유량의 제곱에 비례합니다.

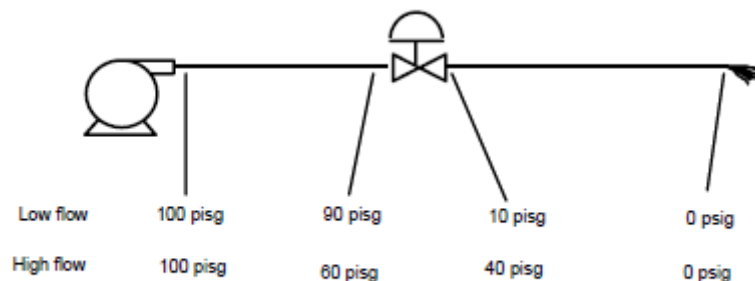


Figure 23 A valve installed a process line.

At high flow, the head loss through the pipe is more, leaving a smaller differential pressure across the valve.

유량이 많으면 파이프를 통한 수두 손실이 더 커지고 밸브를 가로 지르는 차압이 낮아집니다.

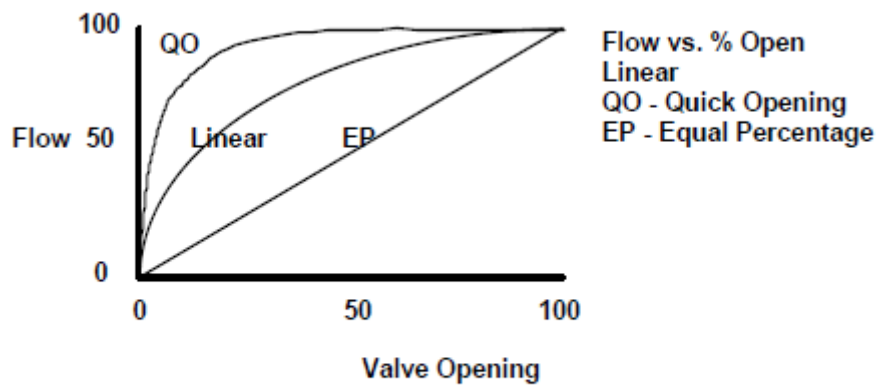


Figure 24 Installed valve characteristics.

5.2 PROCESS DYNAMICS

과정 역학

The measured variable does not change instantly with the controller output changes.
 측정 된 변수는 컨트롤러 출력이 변경되면 즉시 변경되지 않습니다.

Instead, there is usually some delay or lag between the controller output change and the measured variable change.

대신 컨트롤러의 출력 변화와 측정 된 변수 변화 사이에 약간의 지연 또는 지연이 있습니다.

Understanding the dynamics of the loop is required in order to know how to properly control a process.

프로세스의 적절한 제어 방법을 이해하려면 루프의 역학을 이해해야 합니다.

There are two basic types of dynamics: simple lag and dead time.

역학에는 기본적으로 두 가지 유형이 있습니다. 단순 지연과 데드 타임입니다.

Most processes are a combination of several individual lags, each of which can be classed as simple lag or dead time.

대부분의 프로세스는 몇 가지 개별 래그의 조합으로, 각각 단순 지연 또는 불감 시간으로 분류 할 수 있습니다.

5.2.1 Dead time

데마타임

Dead Time is the delay in the loop due to the time it takes material to flow from one point to another.

데드 타임은 소재가 한 지점에서 다른 지점으로 이동하는 데 걸리는 시간 때문에 루프의 지연입니다.

For example, in the temperature control loop shown below, it takes some amount of time for the liquid to travel from the heat exchanger to the point where the temperature is measured.

예를 들어, 아래에 표시된 온도 제어 루프에서 액체가 열 교환기에서 온도가 측정되는 지점까지 이동하는 데 어느 정도 시간이 걸립니다.

If the temperature at the exchanger outlet has been constant and then changes, there will be some period of time before any change can be observed by the temperature measurement element.

교환기 출구의 온도가 일정하게 변한 후 온도 변화 요소가 관찰되기까지 약간의 시간이 걸립니다.

Dead time is also called distance velocity lag and transportation lag.

데드 타임은 거리 속도 지연 및 운송 지연이라고도 합니다.

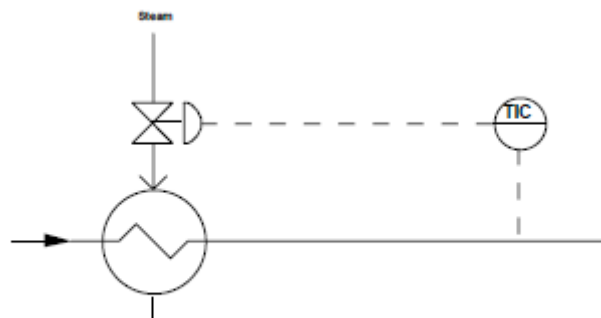


Figure 25 Heat exchanger with dead time

The distance between the heat exchanger and the temperature measurement creates a dead time.

열 교환기와 온도 측정 사이의 거리는 불감 시간을 만듭니다.

Dead time is often considered to be the most difficult dynamic element to control.

데드 타임은 종종 제어하기 가장 어려운 동적 요소로 간주됩니다.

This will become apparent in Chapter 6 , controller tuning.

이것은 6 장 컨트롤러 튜닝에서 명백해질 것입니다.

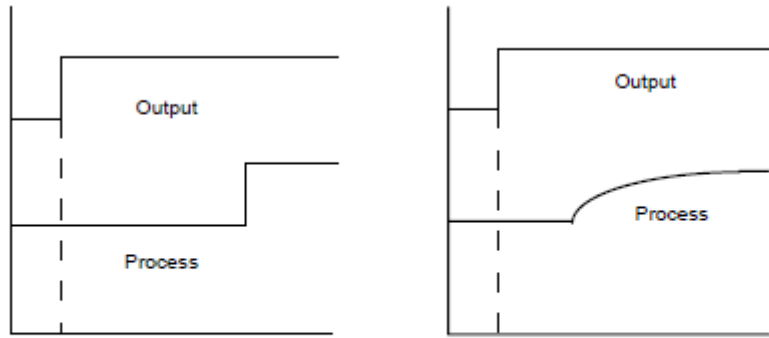


Figure 26 Pure dead time. Figure 27 Dead time and lag.

If a process contains both dead time and a lag, the beginning of the lag will be at the end of the dead time.

프로세스에 데드 타임과 래그가 모두 포함되어있는 경우, 래그의 시작은 데드 타임의 끝 부분에있게됩니다.

5.2.2 Self Regulation

셀프 규칙

For most processes, as a variable increases it will tend to reduce its rate of increase and eventually level off even without any change in the manipulated variable.

대부분의 프로세스에서 변수가 증가하면 증가율이 감소하고 결국 조작 된 변수를 변경하지 않고도 수준이 떨어집니다.

This is referred to as self regulation.

이를 자체 규제라고합니다.

Self regulation does not usually eliminate the need for a controller, because usually the value at which the variable will settle will be unacceptable.

일반적으로자가 조정은 변수가 정착 될 값이 용납 될 수 없으므로 일반적으로 제어가 필요합니다.

The control system will need to act to bring the controlled variable back to its set point.

제어 시스템은 제어 변수를 설정 값으로 되돌리기 위해 작동해야합니다.

An example of self regulation is a tank with flow in and out. The manipulated variable is the liquid flow into a tank.

자체 규제의 한 예가 유입 및 유출되는 탱크입니다. 조작 된 변수는 탱크로의 액체 흐름입니다.

The controlled variable is the flow out of the tank.

제어 변수는 탱크에서 유출되는 유량입니다.

The load is the valve position of the discharge flow.

부하는 배출 흐름의 밸브 위치입니다.

With the valve position constant, the flow out of the tank is determined by the valve position and the level (actually the square of the level).

밸브 위치가 일정하면 탱크의 유출은 밸브 위치와 레벨 (실제로는 레벨의 제곱)에 의해 결정됩니다.

As the level in the tank falls, the pressure (or liquid head) decreases, decreasing the flow rate.

탱크의 레벨이 떨어지면 압력 (또는 액체 헤드)이 감소하여 유속이 감소합니다.

Eventually the discharge flow will decrease to the point that it equals the inlet flow, and the level will maintain a constant value.

결국 배출 유량은 입구 유량과 같아 지도록 레벨이 감소하고 레벨은 일정한 값을 유지합니다.

Likewise, if the flow into the tank increases, the level will begin to increase until the discharge flow equaled the inlet flow (unless the tank became full and overflowed first).

마찬가지로, 탱크로 유입되는 유량이 증가하면 배출 유량이 유입 유량과 같아지기까지 레벨이 증가하기 시작합니다 (탱크가 가득 차서 넘치지 않는 한).

This also occurs in temperature loops.

이는 온도 루프에서도 발생합니다.

Take example, a room with an electric space heater with no thermostat.

예를 들어, 온도 조절 장치가없는 전기 공간 히터가있는 방이있다.

The room is too cool, so we turn on the space heater.

방이 너무 차갑습니다. 그래서 우리는 우주 난방기를 켭니다.

As more heat enters the room, the room temperature increases.

더 많은 열이 실내로 들어 오면 실내 온도가 높아집니다.

However, the flow of heat out through the walls is proportional to the difference between the inside and the outside temperatures.

그러나 벽을 통한 열의 흐름은 내부 온도와 외부 온도의 차이에 비례합니다.

As the room temperature increases, that difference increases, and the heat flow from the room eventually equals the amount of heat produced by the space heater.

실내 온도가 증가함에 따라, 그 차이가 증가하고, 실내로부터의 열 흐름은 결국 공간 가열기에 의해 생성 된 열의 양과 동일해진다.

As the temperature increases the rate of change decreases until the temperature levels off at a higher temperature.

온도가 증가함에 따라 온도가 더 높은 온도에서 떨어질 때까지 변화율이 감소합니다.

Sometimes the self regulation is sufficient to eliminate any need for feedback control.

때로는 자기 통제가 피드백 제어의 필요성을 없애기에 충분합니다.

However, more often the self regulation is not sufficient (the tank overflows or the room becomes too hot), therefore control is still needed.

그러나자가 조절이 충분하지 않은 경우가 더 많습니다 (탱크가 넘치거나 실내 온도가 너무 높아짐). 따라서 제어가 여전히 필요합니다.

Because of self regulations, for at least some range of controller outputs there will be a corresponding process value.

자체 규정으로 인해 컨트롤러 출력 범위 중 적어도 일부에는 해당 프로세스 값이 있습니다.

The self regulation is responsible for the curve shown in the dynamic response of a controlled variable to a change in the measured variable.

자기 제어는 측정 변수의 변화에 대한 제어 변수의 동적 응답에 표시된 곡선을 담당합니다.

5.2.3 Simple lag

심플한 지연

The most common dynamic element is the simple lag.

가장 일반적인 동적 요소는 단순 지연입니다.

If a step change is made in the controller output , the process variable will change as shown in Figure 28.

컨트롤러 출력에서 단계가 변경되면 프로세스 변수가 그림 28과 같이 변경됩니다.

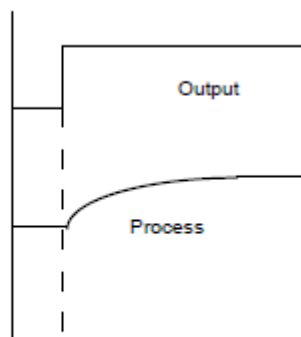


Figure 28 Process with a single lag.

An example of a process dominated by one loop is shown in Figure 29.

하나의 루프에 의해 지배되는 프로세스의 예가 그림 29에 나와 있습니다.

The flow of the liquid out of the vessel is proportional to the level.

용기에서 나오는 액체의 흐름은 레벨에 비례합니다.

If the inlet valve is opened, increasing the flow into the vessel, the level will rise.

입구 밸브가 열리면 용기로 유입되는 유량이 증가하여 레벨이 상승합니다.

As the level rises, the flow output will rise, slowing the rate of increase in the level.

레벨이 올라감에 따라 유량 출력이 상승하여 레벨 증가 속도가 느려집니다.

Eventually, the level will be at the point where the flow out will be equal to the flow in.

결국, 유량은 유량이 유량과 같아지는 지점에있게됩니다.

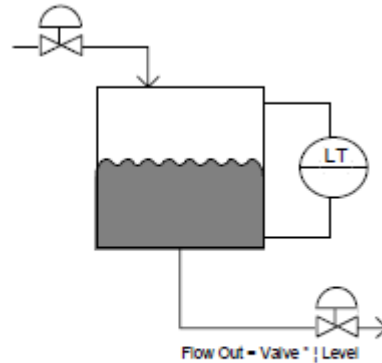


Figure 29 Level is a typical one lag process.

5.2.4 Multiple Lags

다중 지연

Most processes have more than one lag, although some of the lags may be insignificant.

대부분의 프로세스는 하나 이상의 지연이 있지만 일부 지연은 중요하지 않을 수 있습니다.

Lags are not additive. A response of a multiple lag is illustrated in .

지연은 부가적인 것이 아닙니다. 다중 지연에 대한 응답은에 설명되어 있습니다.

The process measured variable begins to change very slowly, and the rate of change increases up to a point, known as the point of inflection, where the rate of change decreases as the measurement approaches its asymptote.

공정 측정 변수는 매우 천천히 변화하기 시작하며, 변화율은 변곡점으로 알려진 지점까지 증가합니다. 변곡점은 측정 값이 점근선에 가까워짐에 따라 변화율이 감소합니다.

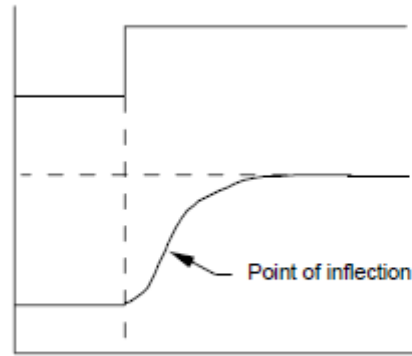


Figure 30 Process with multiple lags.

The first part of the curve, where the rate of change is increasing, is governed primarily by the second largest lag.

변화율이 증가하는 곡선의 첫 번째 부분은 주로 두 번째로 큰 지연에 의해 지배됩니다.

The second part of the curve, beyond the point of inflection, is governed primarily by the largest lag.

굴곡 지점을 넘어서는 곡선의 두 번째 부분은 주로 최대 지연으로 제어됩니다.

5.2.5 Process Order

프로세스 오더

Often processes have been described as first order, second order, etc., based on the number of first order linear lags included in the process dynamics.

종종 공정은 공정 동역학에 포함 된 1 차 선형 지체의 수에 기초하여 1 차, 2 차 등으로 기술되어왔다.

It can be argued that all processes are of a higher order, with a minimum of three lags and a dead time.

모든 프로세스는 3 개의 래그와 데드 타임이 최소 인 고차원이라고 주장 할 수 있습니다.

These lags, which are present in all processes, include the lag inherent in the sensing device, the primary lag of the process, and the time that the valve (or other final control element) takes to move.

모든 프로세스에 존재하는 이러한 지체는 감지 장치에 내재 된 지연, 프로세스의 일차 지연 및 밸브 (또는 다른 최종 제어 요소)가 이동하는 데 걸리는 시간을 포함합니다.

However, in many processes the smaller lags are so much smaller than the largest lag that their contributions to the process dynamics are negligible.

그러나 많은 프로세스에서 작은 지체는 프로세스 지연에 대한 기여도를 무시할 수 있는 가장 큰 지연보다 훨씬 더 작습니다.

Dead time is also present in all processes.

데드 타임은 모든 프로세스에도 존재합니다.

With pneumatic control, there is some dead time due to the transmission of the pressure signal from the process to the controller, and from the controller to the valve.

공압 제어의 경우 프로세스에서 컨트롤러로, 컨트롤러에서 밸브로 압력 신호가 전송되기 때문에 다소 시간이 걸립니다.

This is eliminated by electronic controls (unless one considers the transmission of the electric signal, usually a few microseconds or less).

이것은 전자 제어 (일반적으로 수 마이크로 초 이하의 전기 신호의 전송을 고려하지 않는 한)에 의해 제거됩니다.

With digital controls, there is an effective dead time equal to one half the loop scan rate [2].

디지털 컨트롤을 사용하면 루프 스캔 속도의 절반에 해당하는 유효 불감 시간이 있습니다 [2].

In most cases, the loop will be scanned fast enough so that this dead time is insignificant. In some cases, such as liquid flow loops, this dead time is significant and affects the amount of gain that can be used.

대부분의 경우, 이 데드 타임이 중요하지 않도록 루프가 충분히 빠르게 스캔됩니다. 액체 흐름 루프와 같은 일부 경우, 이 데드 타임은 중요하며 사용될 수 있는 이득의 양에 영향을 미칩니다.

Rather than consider a process to be first order, second order, etc., it may be better to consider all loops to be higher order to a degree.

프로세스가 1 차, 2 차 등으로 간주하기보다는 모든 루프가 어느 정도 더 높은 차수라고 생각하는 것이 좋습니다.

As an alternative to process order, we will characterize processes by the degree to which one first order lag dominates the other lags in the process (not considering any true dead time).

프로세스 순서에 대한 대안으로 프로세스의 다른 지연 시간을 1 차 래그가 지배하는 정도로 프로세스를 특성화합니다 (실제 데드 타임을 고려하지 않음).

Dominant-lag processes are those that consist of a dead time plus a single significant lag, with all other lags small compared to the major lag.

지배적 지연 과정은 주요 지연과 비교하여 다른 모든 지연이 있는 데드 타임과 단일 중요한 지연으로 구성된 프로세스입니다.

Multiple-lag or non-dominant-lag processes are those in which the longest lag is not significantly longer than the next longest lag.

다중 지연 또는 비 지배 지연 프로세스는 가장 긴 지연이 다음으로 긴 지연보다 상당히 길지 않은 프로세스입니다.

One measure of the dominance of a single lag is the value of the process measurement at which the point of inflection (POI) occurs.(see Figure 31)

단일 지연의 우세를 측정하는 한 가지 방법은 변곡점 (POI)이 발생하는 공정 측정값입니다 (그림 31 참조)

In the most extreme case (only a single lag) the POI occurs at the initial process value.

가장 극단적 인 경우 (단일 지연 만) POI는 초기 프로세스 값에서 발생합니다.

With about three equal, non-interacting lags the POI occurs at about 33% of the difference between the initial and the final process value.[7]

약 3 개의 동일하고 상호 작용하지 않는 지체 시간과 함께 POI는 초기 및 최종 프로세스 값의 차이의 약 33 %에서 발생합니다. [7]

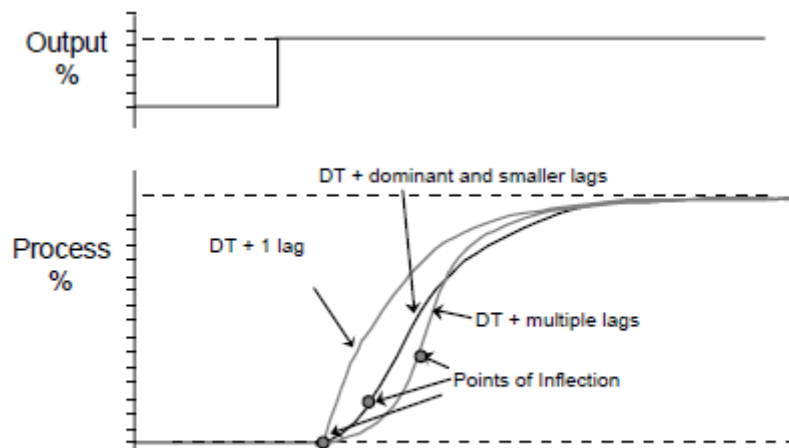


Figure 31 The step response for different numbers of lags.

As the number of lags increase, the value of the process at the point of inflection increases.

지연의 수가 증가함에 따라 굴곡 지점에서의 프로세스 값이 증가합니다.

5.3 MEASUREMENT OF PROCESS DYNAMICS

프로세스 동역학의 측정

Process dynamics usually consist of several lags and dead time.

프로세스 역학은 일반적으로 여러 개의 지연 시간과 데드 타임으로 구성됩니다.

The dynamics differ from one loop to another.

동역학은 루프마다 다릅니다.

The dynamics can be expressed by a detailed list of all of the lags and the dead time of the loop, or they can be approximated using a simpler model.

역학은 루프의 모든 지연과 데드 타임의 세부 목록으로 표현하거나 간단한 모델을 사용하여 근사시킬 수 있습니다.

One such model is a dead time and a first order lag.

이러한 모델 중 하나는 데드 타임 (dead time)과 일차 주문 래그 (first order lag)입니다.

Graphically, the process response of such a model is:

그래픽 적으로 그러한 모델의 프로세스 응답은 다음과 같습니다.

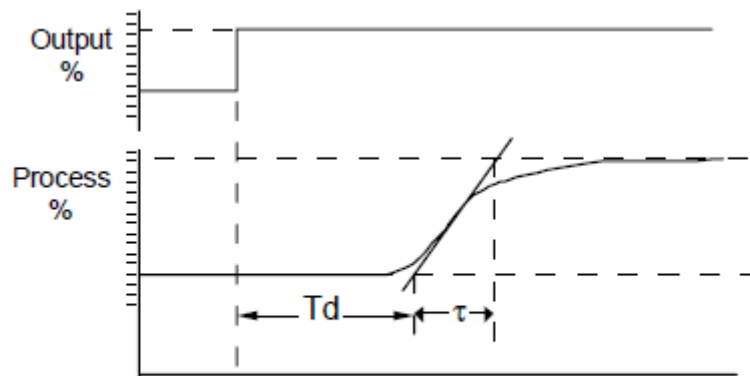


Figure 32 Pseudo dead time and process time constant.

The dynamics can be approximated by two numbers:

τ is the process time constant.

동역학은 두 개의 수로 근사 될 수 있습니다 :

τ 는 공정 시간 상수이다.

It is approximately equal to the largest lag in the process.

그것은 프로세스에서 가장 큰 지연과 거의 같습니다.

T_d is the pseudo dead time and approximates the sum of the dead time plus all lags other than the largest lag.

T_d 는 의사 데드 타임이며 데드 타임에 최대 래그를 제외한 모든 래그를 합한 것입니다.

5.3.1 First Order Plus Dead Time Approximation

첫번째 오더 플러스 데드타임의 근사

Several tuning methods (such as the Ziegler-Nichols open loop method) are based on an approximation of the process as a combination of a single first order lag and a dead time, known as the First Order Plus Dead Time (FOPDT) model.

지글러 - 니콜스 개방 루프 방법과 같은 여러 튜닝 방법은 FOPDT (First Order Plus Dead Time) 모델로 알려진 단일 일차 지연과 데드 타임의 결합으로 프로세스를 근사하는 것을 기반으로합니다.

These methods identify the process by making a step change in the controller output.
이러한 방법은 컨트롤러 출력을 단계적으로 변경하여 프로세스를 식별합니다.

The process trend is recorded and graphical or mathematical methods are used to determine the process gain, dead time, and first order lag.
프로세스 추세가 기록되고 그래픽 또는 수학적 방법을 사용하여 프로세스 이득, 데드 타임 및 일차 지연을 결정합니다.

Process gain is the ratio of the change in the process to the change in the controller output signal.

프로세스 이득은 컨트롤러 출력 신호의 변화에 대한 프로세스의 변화의 비율입니다.

It depends upon the range of the process measurement and includes effects of the final control element.

이는 공정 측정 범위에 따라 달라지며 최종 제어 요소의 영향을 포함합니다.

Pseudo dead time (T_d) is the time between the controller output change and the point at which the tangent line crosses the original process value.

의사 데드 타임 (T_d)은 컨트롤러 출력 변경과 접선이 원래 프로세스 값을 교차하는 시점 사이의 시간입니다.

The pseudo dead time is influenced by the dead time and all of the lags smaller than the longest lag in the process.

의사 데드 타임은 프로세스의 가장 긴 지연보다 작은 데드 타임 및 모든 래그의 영향을받습니다.

Process time constant (τ) is the rate of change of the process measurement at the point at which the rate of change is the highest.

프로세스 시간 상수 (τ)는 변화율이 가장 높은 지점에서 프로세스 측정 값의 변화율입니다.

The time constant is strongly influenced by the longest lag in a multiple lag process.

시간 상수는 다중 지연 과정에서 가장 긴 지연의 영향을 강력하게받습니다.

The ratio of the pseudo dead time to the process time constant is often referred to as an "uncontrollability" factor (F_c) that is an indication of the quality of control that can be expected.

프로세스 시간 상수에 대한 의사 데드 타임의 비율은 예상 할 수있는 제어 품질을 나타내는 "제어 불능"인자 (F_c)라고도합니다.

The gain (for a P, PI, and PID controller) at which oscillation will become unstable is inversely proportional to this factor.

발진이 불안정해지는 게인 (P, PI 및 PID 컨트롤러의 경우)은 이 요소에 반비례합니다.

Smith, Murrill, and Moore, [5] proposed that the factor be modified by adding one half of the sample time to the dead time for digital controllers.

Smith, Murrill 및 Moore [5]는 디지털 컨트롤러의 데드 타임에 샘플 시간의 절반을 더함으로써이 요소가 수정 될 것을 제안했다.

5.4 LOADS AND DISTURBANCES

부하 및 불확실

The process measurement is affected not only by the output of the control loop but by other factors called loads.

공정 측정은 제어 루프의 출력뿐만 아니라 부하라는 다른 요소의 영향을받습니다.

These can include such factors as the weather, the position of other valves, and many other factors.

여기에는 날씨, 다른 밸브의 위치 및 기타 여러 요소가 포함될 수 있습니다.

An example is shown in Figure 33.

그림 33에 예가 나와 있습니다.

The level of the tank is controlled by manipulating the valve on the discharge line.

탱크의 높이는 배출 라인의 밸브를 조작하여 제어됩니다.

However, the level is also affected by the flow into the tank.

그러나, 수위는 탱크로 유입되는 물의 영향을받습니다.

In fact, the flow into the tank has just as much effect on the level as the flow out of the tank.

실제로, 탱크로 유입되는 유량은 탱크에서 배출되는 유량만큼 영향을 미칩니다.

The inlet flow is therefore a load.

따라서 입구 흐름은 부하입니다.

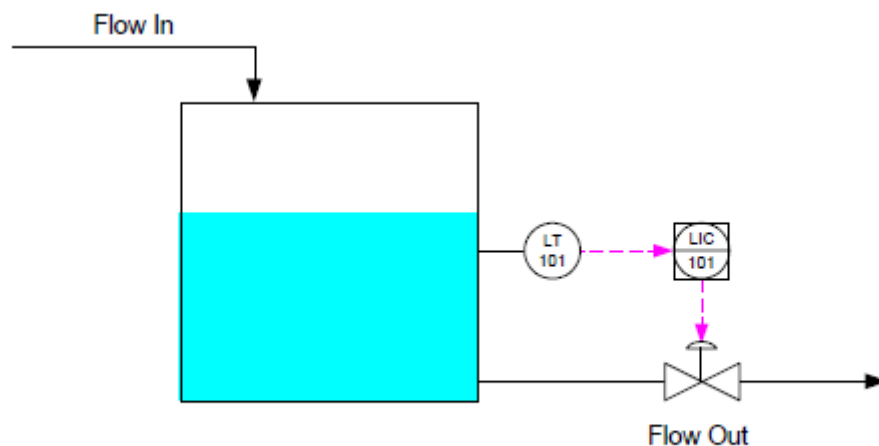


Figure 33 Level control

The level is the controlled variable, the flow out is the manipulated variable, and the flow in is the load.

레벨은 제어 된 변수이고, 흐름은 조작 된 변수이고 흐름은 부하입니다.

A change in the flow in is a disturbance that requires a response by the controller.
흐름의 변화는 컨트롤러의 응답이 필요한 교란입니다.

In a steam heater, the temperature is controlled by the valve on the steam line.
스팀 히터에서 온도는 스팀 라인의 밸브로 제어됩니다.

However, the temperature is also affected by the temperature of the air around the vessel, although not nearly as much as by the steam valve.

그러나 온도는 증기 밸브 만큼은 아니지만 용기 주위의 공기 온도에 의해 영향을받습니다.

This temperature outside of the vessel is also a load.
선박 외부의 온도도 부하입니다.

The changes in a load are called disturbances.
부하의 변화를 외란이라고합니다.

Almost all processes contain disturbances.
거의 모든 프로세스에 장애가 있습니다.

They can be as major as the effect of the inlet flow on the vessel or as minor as the effect of weather on the temperature loop.

그것들은 용기에 유입되는 흐름의 영향만큼 중요 할 수도 있고, 기상이 온도 루프에 미치는 영향만큼 작을 수도 있습니다.

#CHAPTER 6

LOOP TUNING