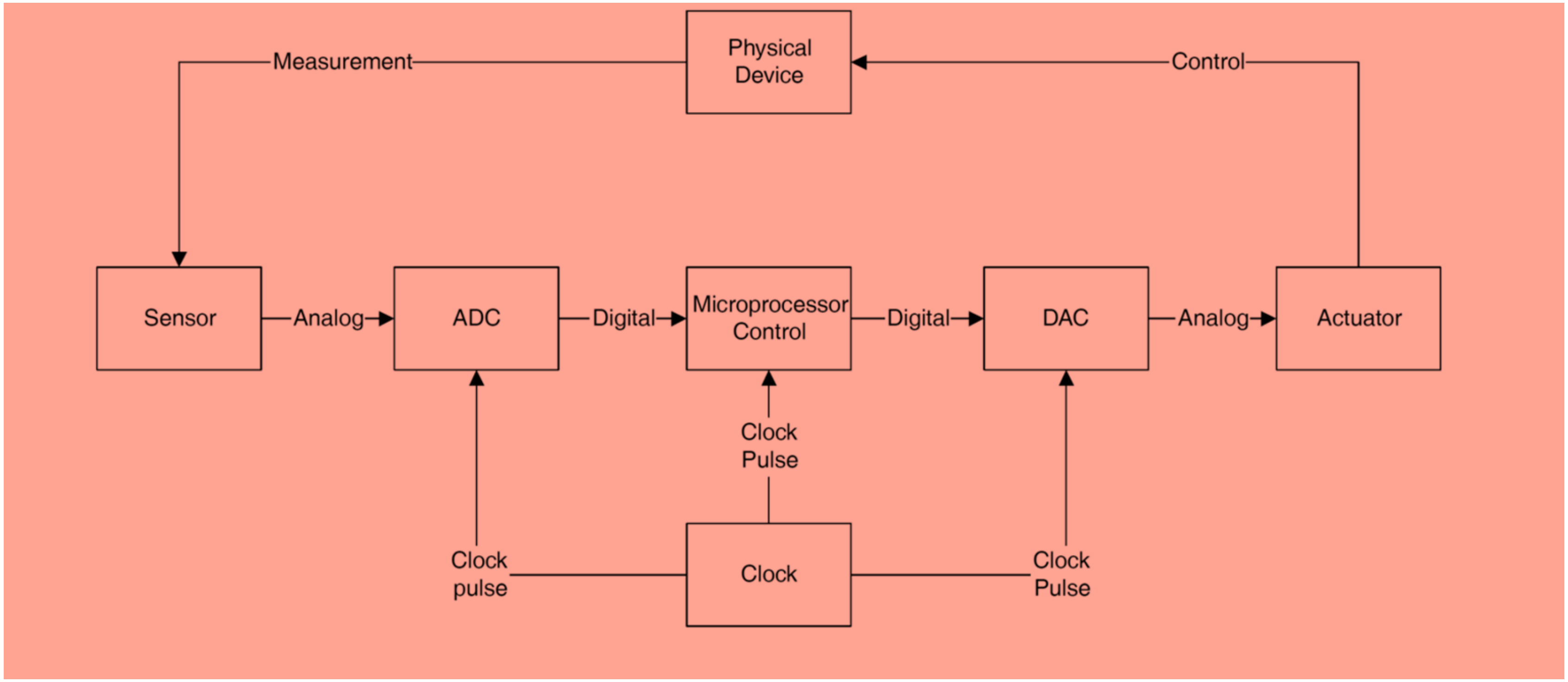
Mechatronics

Feedback Control Principals

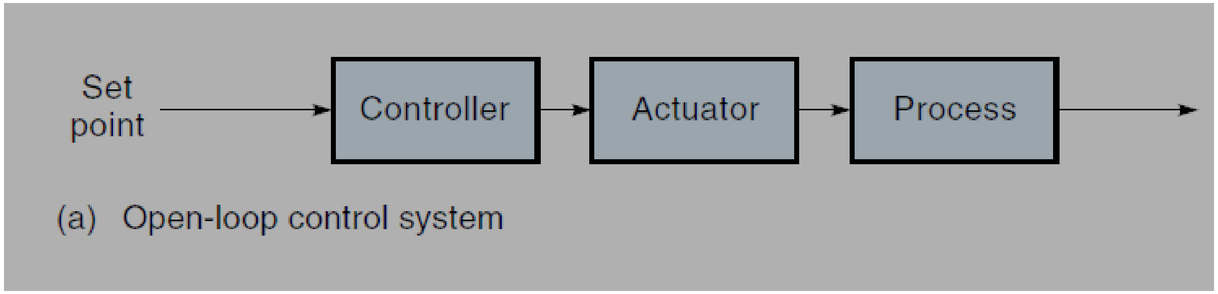
University of Mohaghegh Ardabili (UMA)
Dr. K Sabahi
ksabahi2005@gmail.com

Mechatronics system component

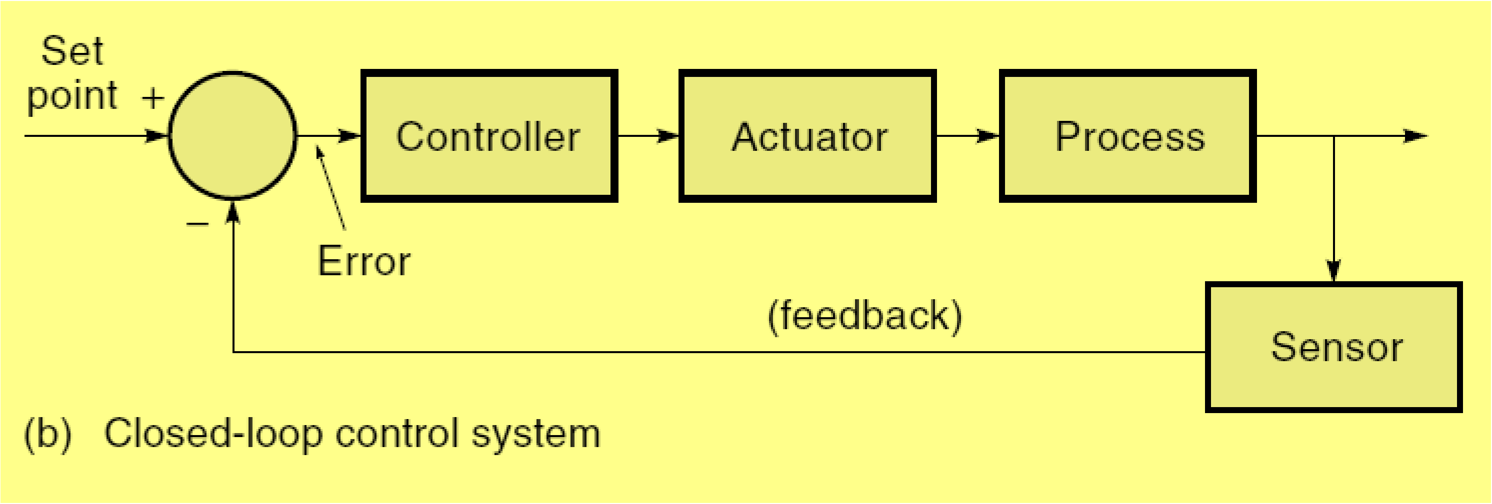


INTRODUCTION

control systems can be classified into two groups: open-loop and closed-loop.



(a) Open-loop control system



(b) Closed-loop control system

INTRODUCTION

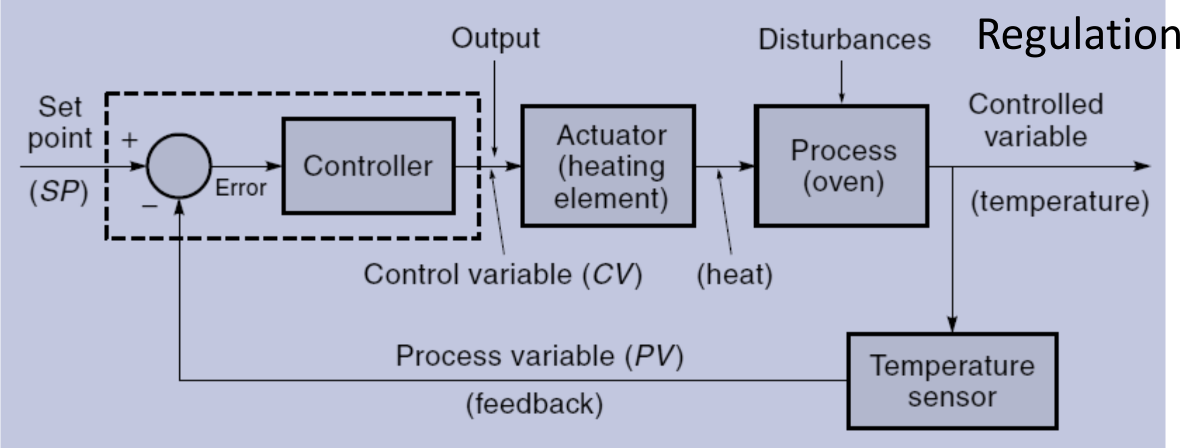
In an **open-loop system** , no feedback is used, so the controller must independently determine what signal to send to the actuator. The trouble with this approach is that *the controller never actually knows if the actuator did what it was supposed to do.*

In a **closed-loop system,** also known as a *feedback control system*, the output of the process is constantly monitored by a sensor

Clearly, the "heart" of the control system is the **controller,** an analog or digital circuit that accepts data from the sensors, makes a decision, and sends the appropriate commands to the actuator.

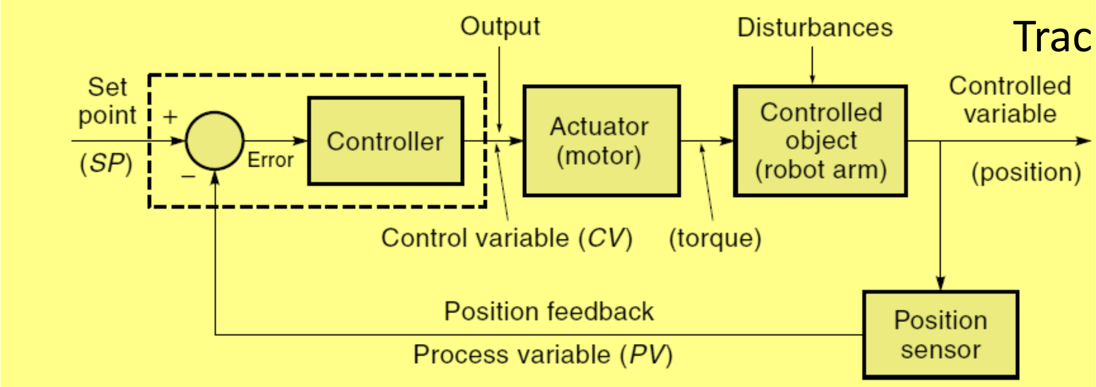
classifications of feedback control systems

Regulation problem



(a) Process control system (heating system)

Tracking problem



(b) Servomechanism (robot arm)

classifications of feedback control systems

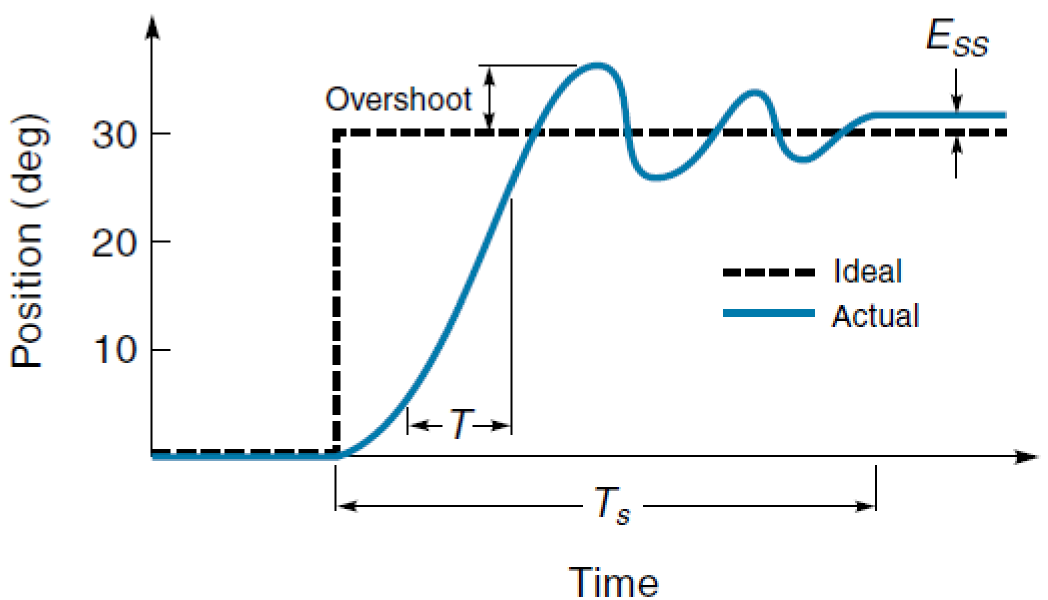
a **process control system (regulation)**, the job of the controller is to maintain a stationary set point despite disturbances—for example, maintaining a constant temperature in an oven whether the door is opened or closed.

In a **servomechanism (tracking)** , the job of the controller is to have the controlled variable track a moving set point—for example, moving a robot arm from one position to another.

PERFORMANCE CRITERIA

What are our criteria for design a controller?

These are divided into transient (moving) and steady-state (not changing) parameters.



Type of closed-loop control strategy

- **ON–OFF CONTROLLERS**
- **PROPORTIONAL CONTROL**
- **INTEGRAL CONTROL**
- **DERIVATIVE CONTROL**
- **PROPORTIONAL + INTEGRAL + DERIVATIVE CONTROL**

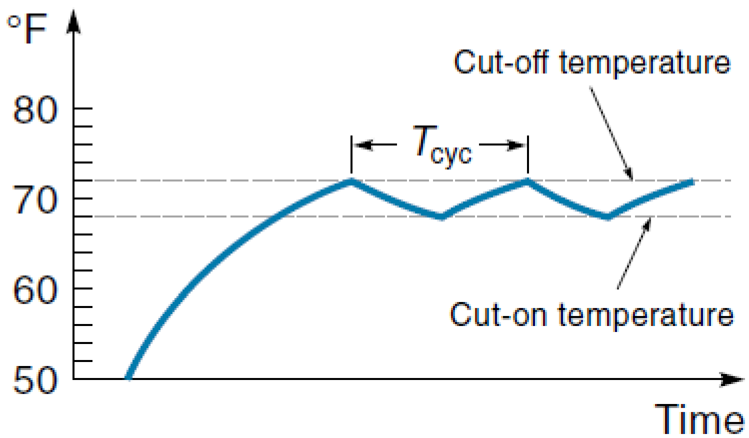
ON–OFF CONTROLLERS

Two-point control (also called **on–off control**) is the simplest type of closed-loop control strategy.

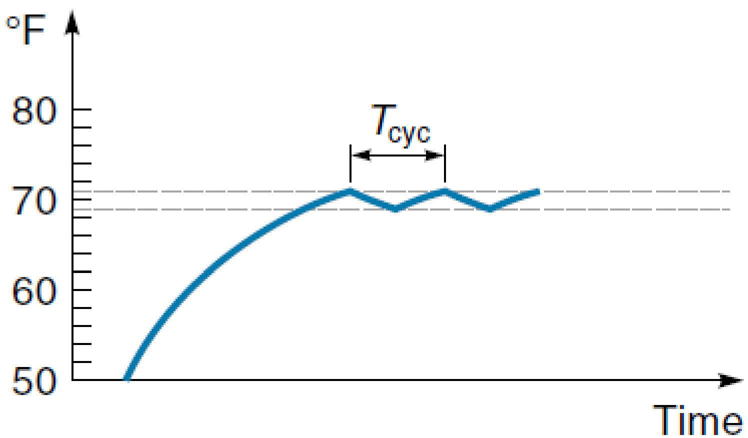
The actuator can push the controlled variable with only full force or no force. When the actuator is off, the controlled variable settles back to some rest state. A good example of two-point control is a thermostatically controlled heating system.

two-point control has only limited applications, mostly on slow-moving systems where it is acceptable for the controlled variable to move back-and-forth between the two limit points.

ON–OFF CONTROLLERS



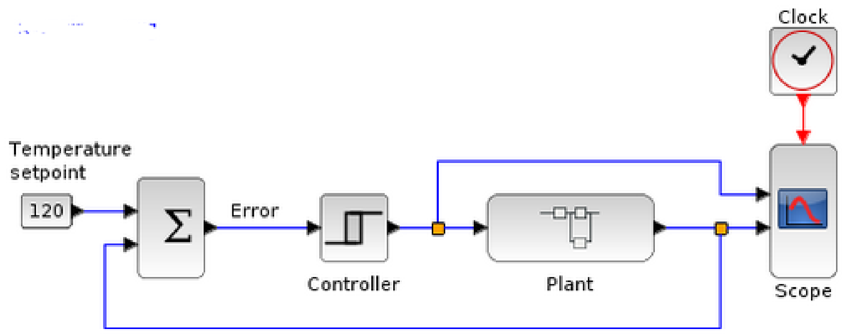
(a) Cut-on = 68°; cut-off = 72°

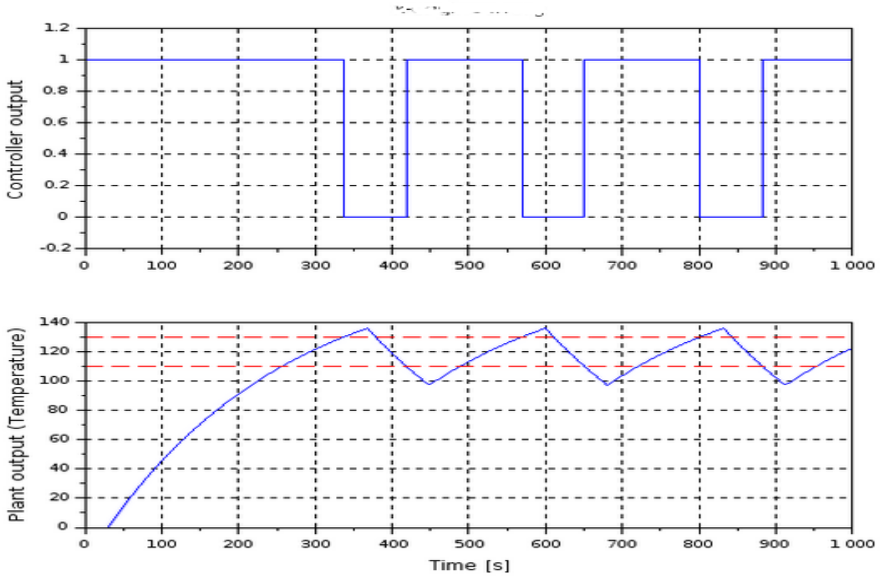


(b) Cut-on = 69°; cut-off = 71°

Generally, a high cycle rate is undesirable because of wear on motors and switches.

Example: ON–OFF CONTROLLERS



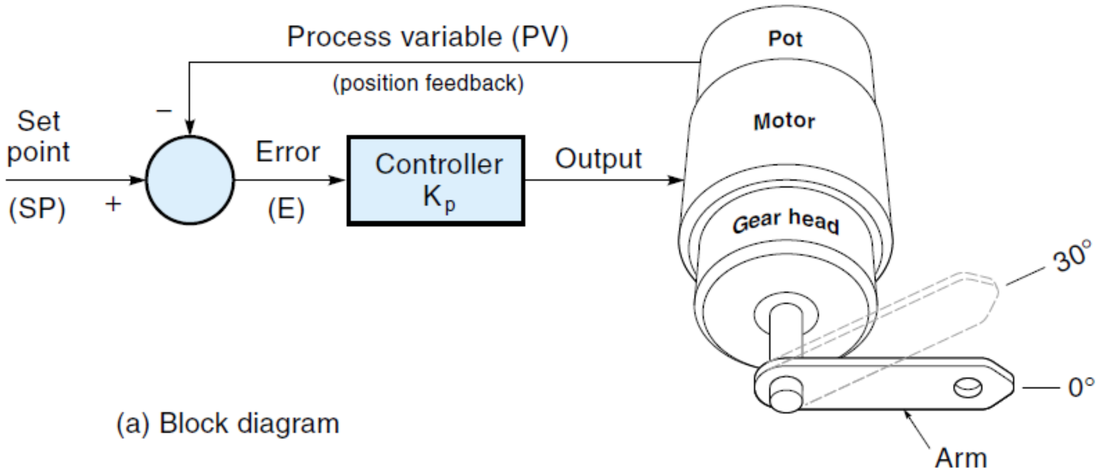


PROPORTIONAL CONTROL

more sophisticated control strategies that require "smart" controllers that use op-amps or a microprocessor.

The first and most basic of these strategies is called proportional control.

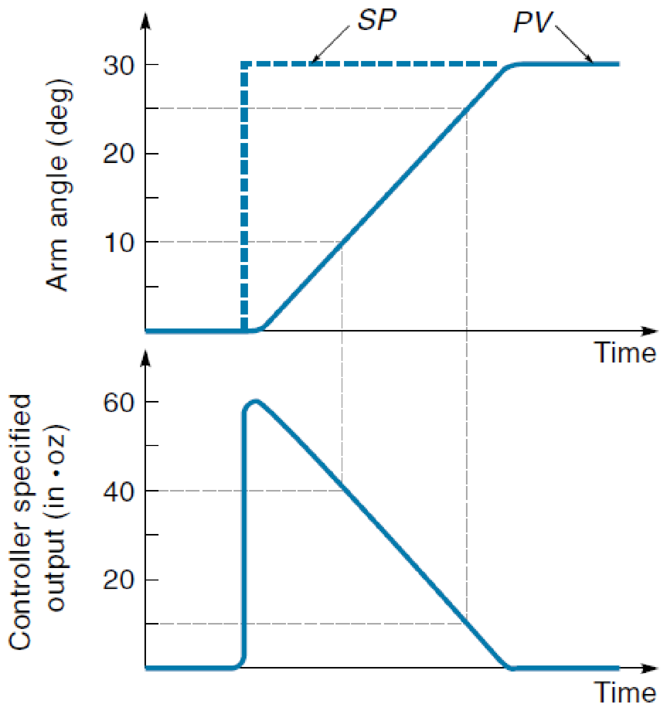
With proportional control, the actuator applies a corrective force that is proportional to the amount of error.



(a) Block diagram

Output = $KP*E$

PROPORTIONAL CONTROL



(b) Graphs showing response to change in set point

The gain of the controller is KP = 2 in. · oz/deg.

A large error implies there is a long way to go, and so you want some speed to get there (which requires a large torque from the motor). However, when the error is small, the arm should slow down (small torque) so as not to overshoot.

PROPORTIONAL CONTROL: The Steady-State-Error Problem

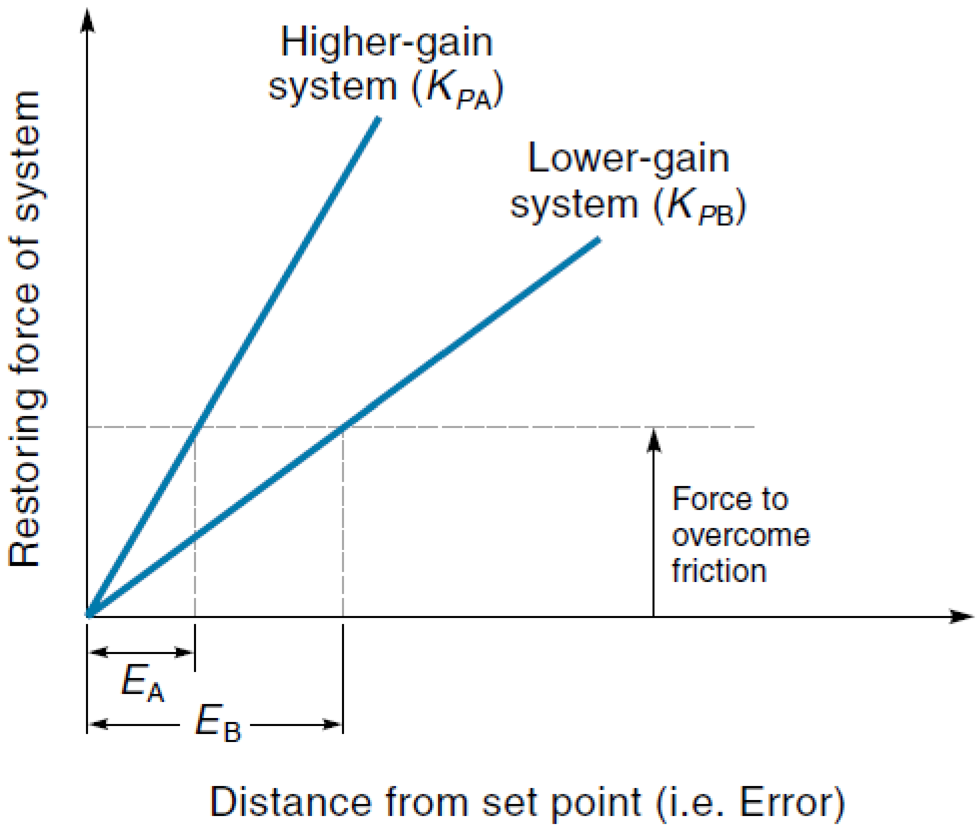
❑ The Friction Problem

Proportional control is simple, makes sense, and is the basis of most control systems, but it has one fundamental problem—steady-state error

In practical systems, proportional control cannot drive the controlled variable to zero error because as the load gets close to the desired position, the correcting force drops to near zero. This small force may not be enough to overcome **friction**,

One way to decrease the steady-state error due to friction is to **increase the controller gain** which could be done in the model of stiffer springs.

PROPORTIONAL CONTROL:
The Steady-State-Error Problem



PROPORTIONAL CONTROL: The Steady-State-Error Problem

- **The Gravity Problem**

Another source of steady-state error is the *gravity problem*, which occurs when a constant external force is pushing on the controlled variable

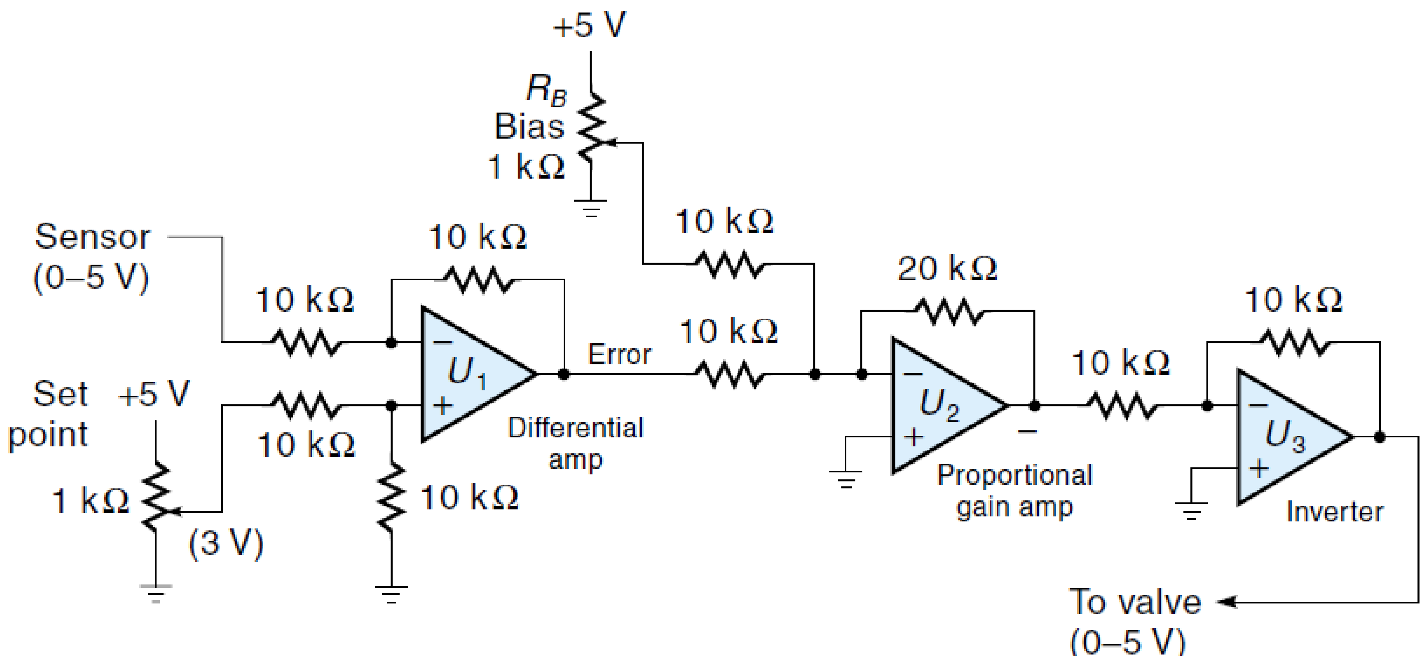
Solution: Bias

One way to deal with the gravity problem is to have the controller add in a constant value (to its output) that is just sufficient to support the weight

$$\text{Output}P = KPE + \text{bias}$$

Analog Proportional Controllers

An analog controller typically uses op-amps to provide the necessary gain and signal processing

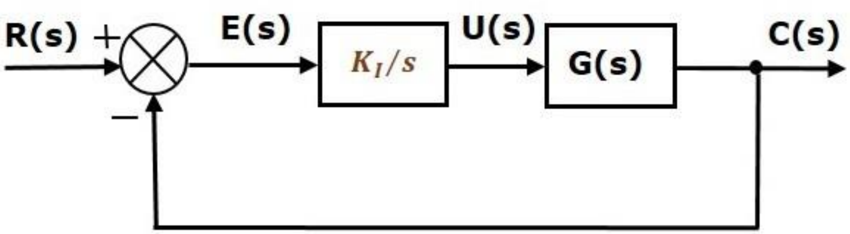


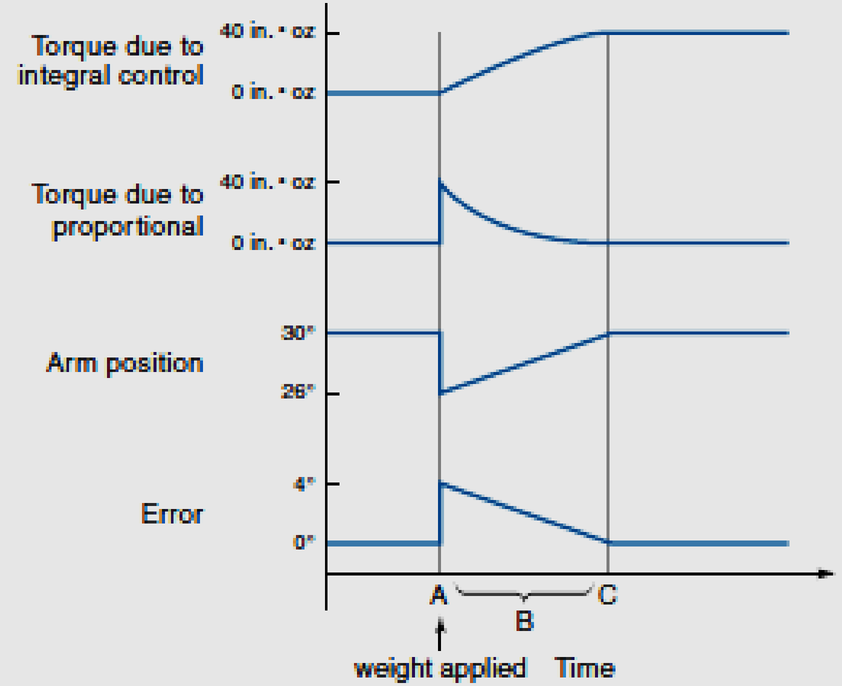
(b) Circuit diagram of an analog controller

INTEGRAL CONTROL

The introduction of **integral control** in a control system can reduce the steady-state error to zero. Integral control creates a restoring force that is proportional to the sum of all past errors multiplied by time

Control Output= $KI*KP*\Sigma(E\Delta t)$





(b) Proportional and integral contributions

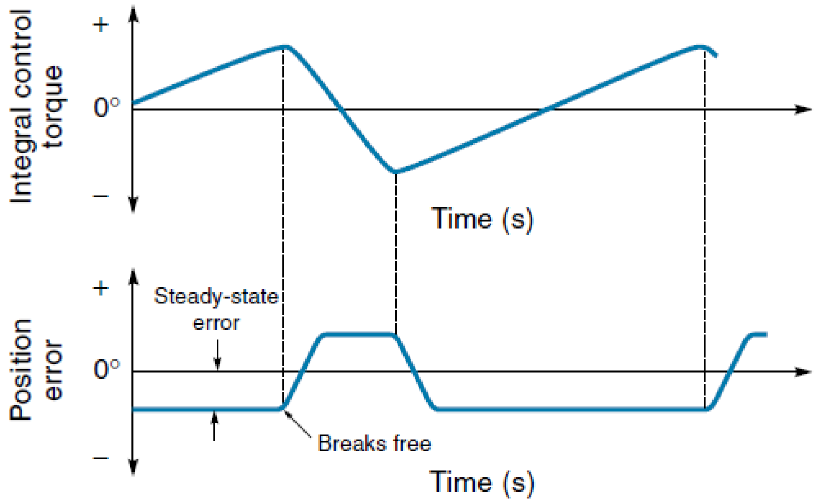
INTEGRAL CONTROL

Integral control response can easily be observed on industrial robots. When a weight is placed on the arm, it will visibly sag and then restore itself to the original position.

The problem is that the proportional-integral system has no way (other than friction) to slow the object *before* it gets to the new set point. The system must overshoot before any active braking will be applied. So, unfortunately, the addition of integral feedback solves one problem, steady-state error, but it creates others: overshoot and decreased stability

INTEGRAL CONTROL Problem

Integral control may cause overshoot and oscillations

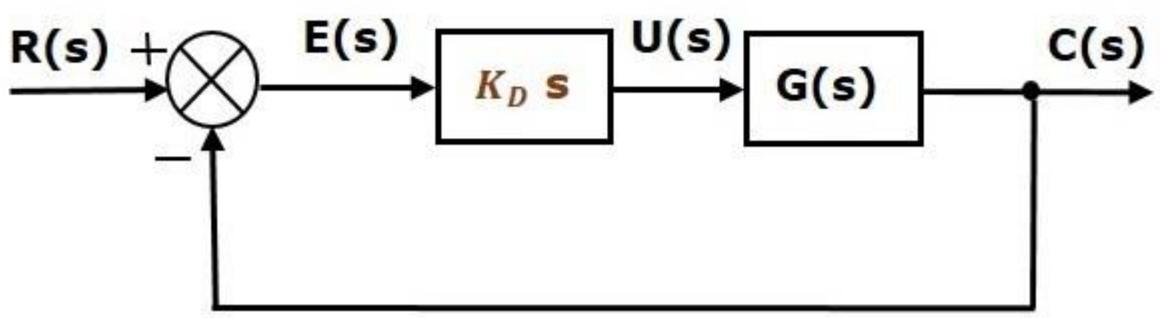


Also, the response of integral feedback is relatively slow because it takes a while for the error time area to build up.

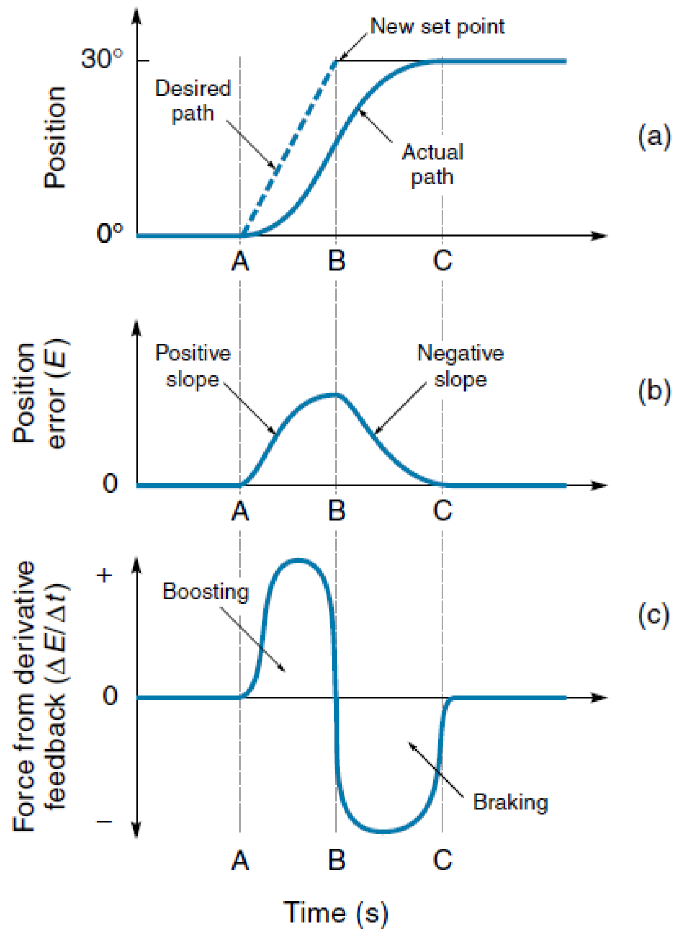
DERIVATIVE CONTROL

One solution to the overshoot problem is to include derivative control. **Derivative control** "applies the brakes," slowing the controlled variable just before it reaches its destination.

Output$D$ = KD*KP* **Δ*E*/ Δ *t***



DERIVATIVE CONTROL



Contribution of derivative control, showing boosting and braking.

derivative control improves system performance in two ways. First, it provides an extra boost of force at the beginning of a change to promote faster action; second, it provides for braking when the object is closing in on the new set point.

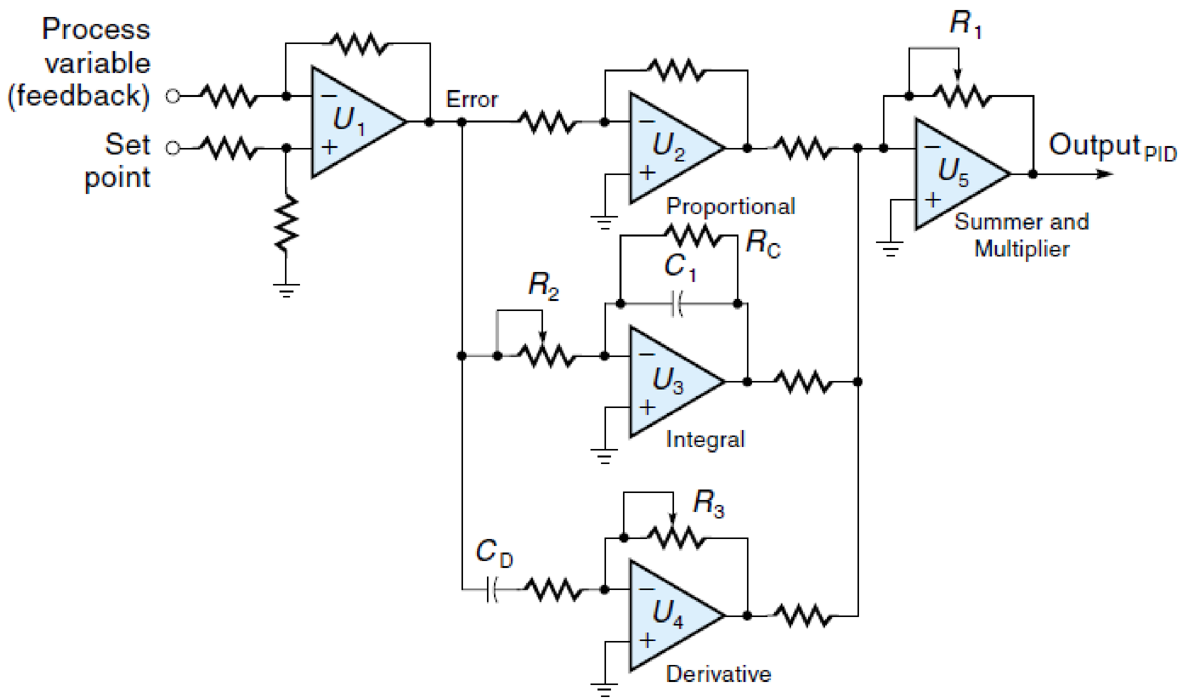
PROPORTIONAL + INTEGRAL + DERIVATIVE CONTROL

Many control systems use a combination of the three types of feedback already discussed: **P**roportional + **I**ntegral + **D**erivative (**PID**) Control

The foundation of the system is proportional control. Adding integral control provides a means to eliminate steady-state error but may increase overshoot. Derivative control is good for getting sluggish systems moving faster and reduces the tendency to overshoot.

PROPORTIONAL + INTEGRAL + DERIVATIVE CONTROL

OutputPID = $K_p[E + K_I\Sigma E\Delta t + K_D\Delta E/\Delta t]$

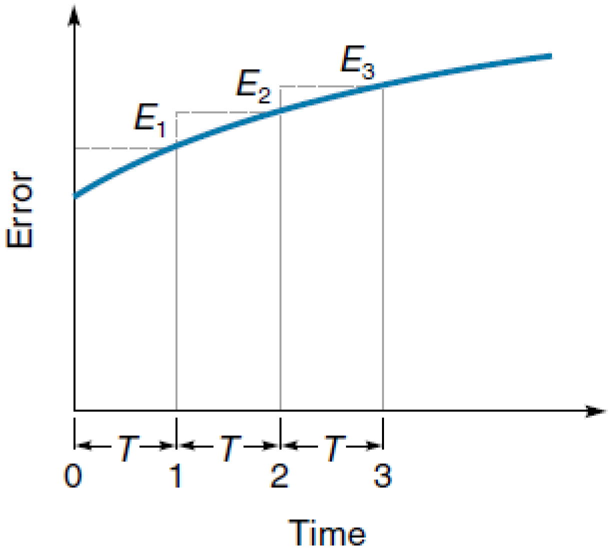


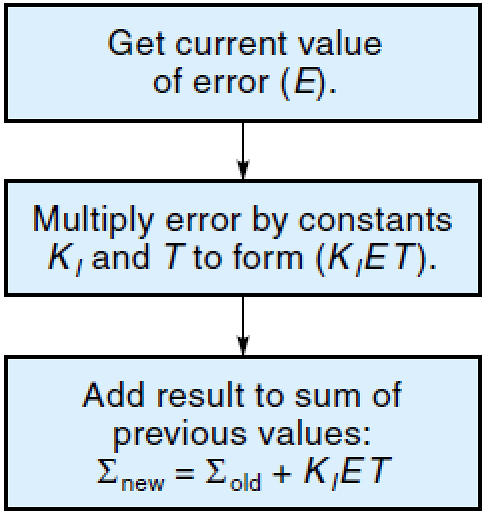
Realize the Digital PID Controllers

A PID digital controller is essentially a computer, most likely microprocessor-based.

❖ Integral term calculation

$K_I\Sigma(E\Delta t) = K_I E_1 T + K_I E_2 T + K_I E_3 T$

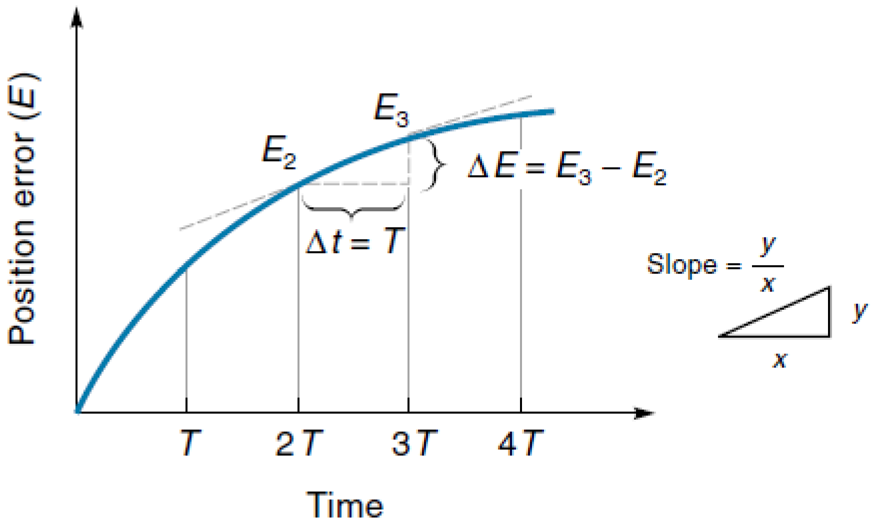


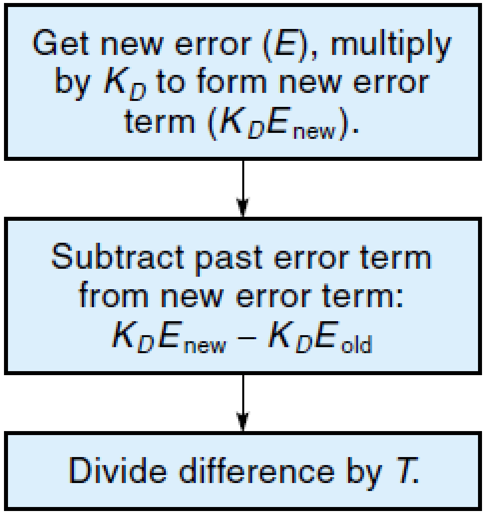


Realize the Digital PID Controllers

❖ Derivative term calculation

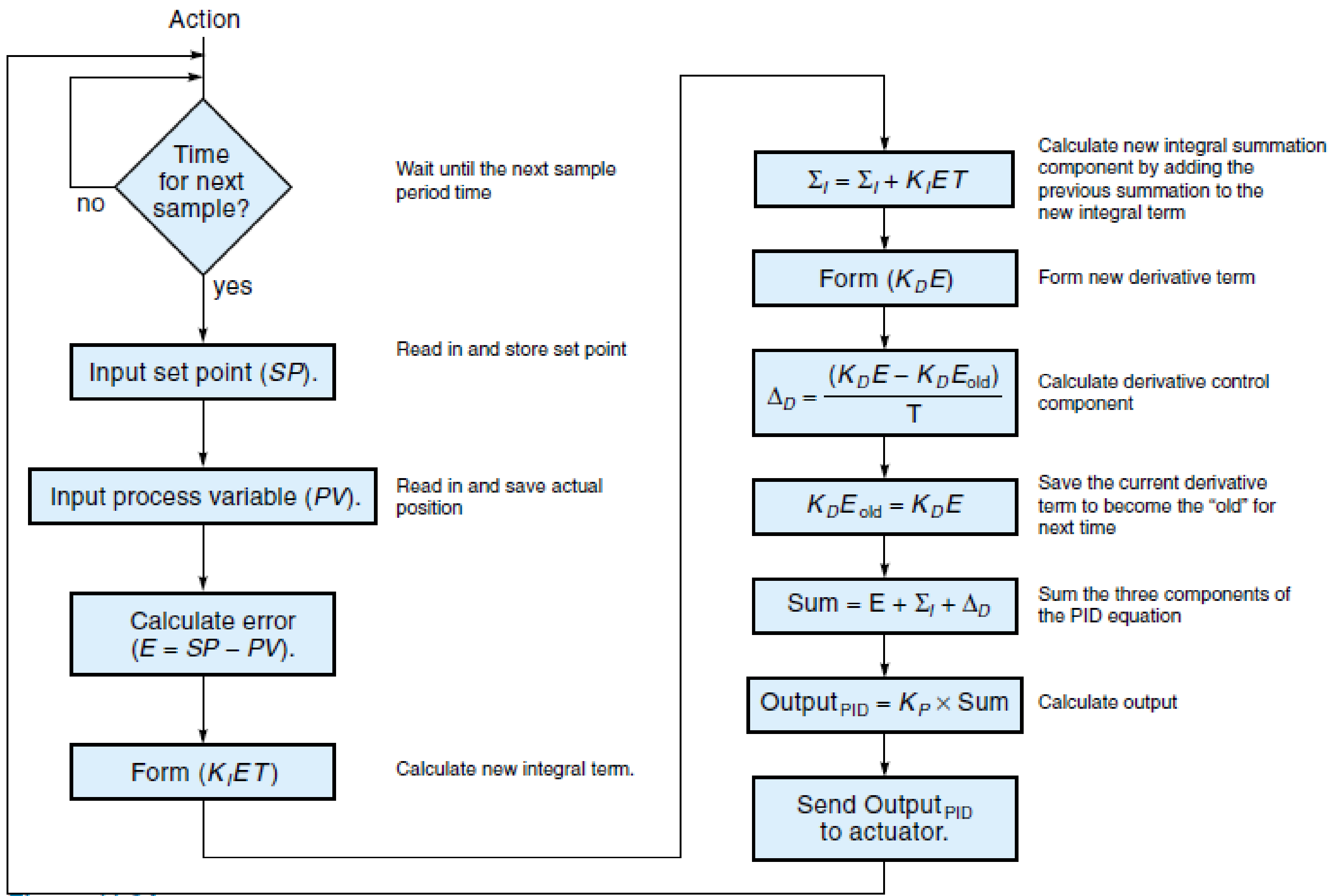
$$\text{Slope} = \frac{\Delta E}{\Delta t} = \frac{(E_3 - E_2)}{T} \qquad K_D \frac{\Delta E}{\Delta T} = \frac{(K_D E_3 - K_D E_2)}{T}$$





Realize the Digital PID Controllers

Flowchart for computing the PID equation



Tuning the PID Controller

The method of arriving at numerical values for the constants *KP* *KI* and *KD* depends on the application. Traditionally, PID control was applied to process control systems. However, with the advent of small, fast, off-the-shelf PID modules, PID control is being applied to position control systems (such as robots) as well.

- ❑ **Trial and error Approach**

- ❑ **Ziegler and Nichols Methods**
- ❖ **Open-loop method**
- ❖ **Closed-loop method**

How to adjust PID Parameters?
Trial and error Approach

Do the following steps:

- the constants KP, KI and KD are set to initial values
- connect the controller to the system
- the system is operated, and the response is observed
- Based on the response, adjustments are made to *KP*, *KI* and *KD*, and the system is operated again

This iterative process of adjusting each constant in an orderly manner until the desired system response is achieved is called **tuning.**

Ziegler and Nichols Tuning Methods

I. the continuous-cycle method and
II. the reaction-curve method

- The **continuous-cycle method** (closed-loop method) can be used when harm isn't done if the system goes into oscillation.

- The **reaction-curve method** (open-loop method) is another way of determining initial settings of the PID parameters. This method does not require driving the system to oscillation.

Ziegler and Nichols Tuning Methods: continuous-cycle method

1- Set *KP* = 1, *KI* = 0, and *KD* = 0 and connect the controller to the system.

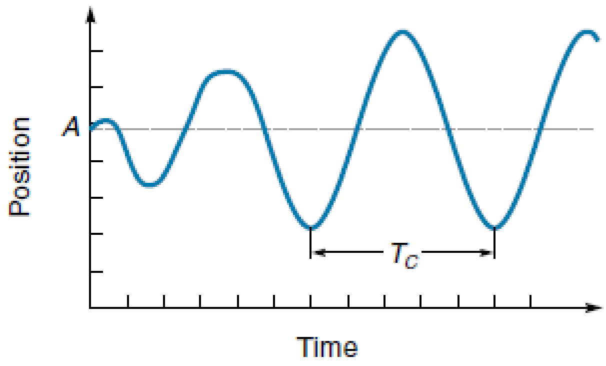
2- increase the proportional gain (*K′P*) while forcing small disturbances to the set point (or the process) until the system oscillates with a constant amplitude,

3- Based on the values of *K′*P and *TC* from step 2, calculate the initial settings of *K*P, *KI* * , and *KD** as follows:

$$K_P = 0.6\,K'_P$$

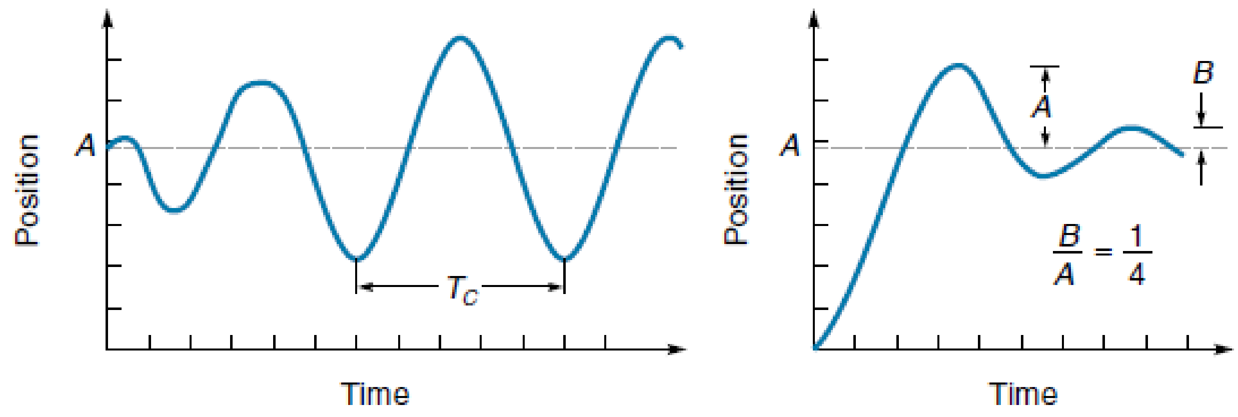
$$K_I = \frac{2}{T_C}$$

$$K_D = \frac{T_C}{8}$$



Ziegler and Nichols Tuning Methods: continuous-cycle method

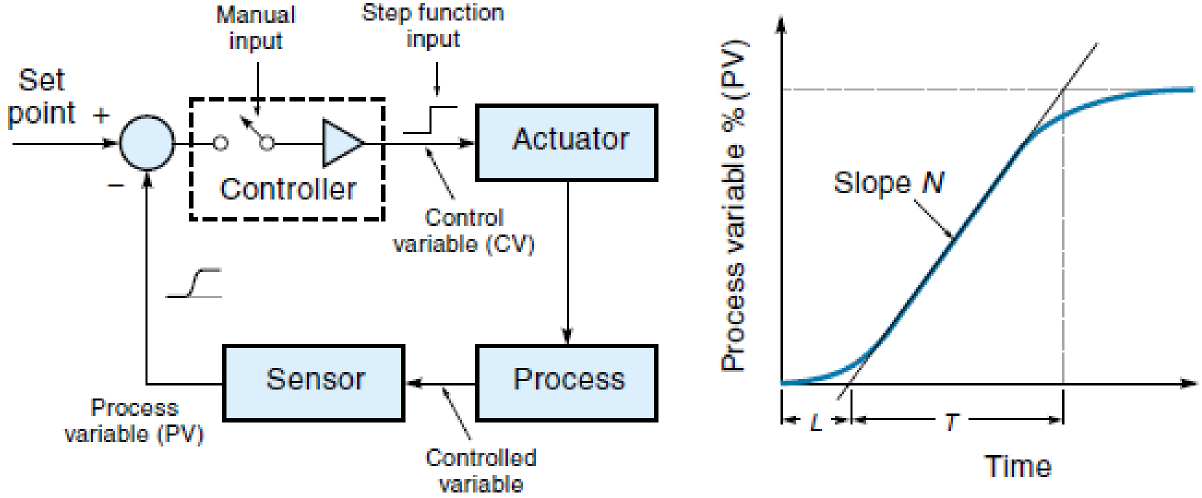
4. Using the settings from step 3, operate the system, note the response, and make adjustments as called for. Increasing *KP* will produce a stiffer and quicker response, increasing *KI* will reduce the time it takes to settle out to zero error, and increasing *KD* will decrease overshoot. Of course, *KP KI* and *KD* do not act independently, so changing one constant will have an effect across the board on system response.



(a) System as forced into oscillation (b) Resulting response after tuning

Ziegler and Nichols Tuning Methods: The reaction-curve method

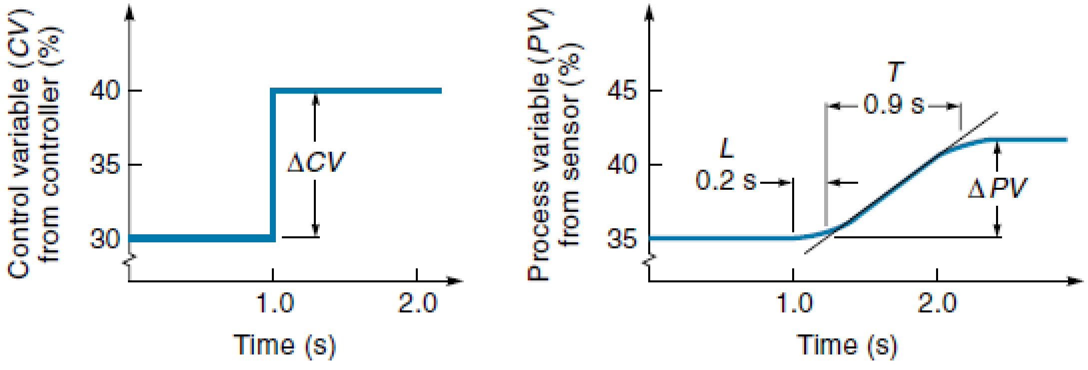
This method does not require driving the system to oscillation. Instead, the feedback loop is opened, and the controller is manually directed to output a small step function to the actuator.



(a) Block diagram of test setup

(b) Graph of response to step input

Ziegler and Nichols Tuning Methods: The **reaction-curve method**



(a) Step input

(b) Response to step input

$$N = \frac{\Delta PV}{T}$$

$$K_P = \frac{1.2 \Delta CV}{NL}$$

$$K_I = \frac{1}{2L}$$

$$K_D = 0.5\ L$$

Example:

$$K_P = \frac{1.2 \Delta CV}{NL} = \frac{1.2 \times 10\%}{7.8\%/\text{s} \times 0.2\ \text{s}} = 7.7$$

$$K_I = \frac{1}{2L} = \frac{1}{2 \times 0.2\ \text{s}} = 2.5/\text{s}$$

$$K_D = 0.5L = 0.5 \times 2\ \text{s} = 0.1\ \text{s}$$

$$N = \frac{\Delta PV}{T} = \frac{7\%}{0.9\ \text{s}} = 7.8\%/\text{s}$$

PIP CONTROLLERS

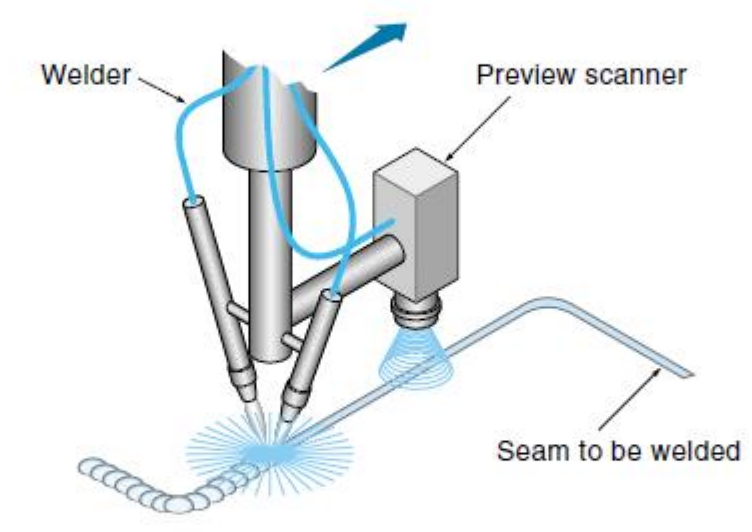
- The PID controller has information about only the past and the present, not where it is going, which is a severe handicap to place on the system.

- Without the ability to look ahead, the driver would have to go very slow or risk driving off the road (overshoot) on a sharp turn

- A **P**roportional + **I**ntegral + **P**review (**PIP**) controller is a system that incorporates information of the future path in its current output.

PIP CONTROLLERS

- Many systems have this information available—either the entire path is stored in memory or the system is equipped with a preview sensor



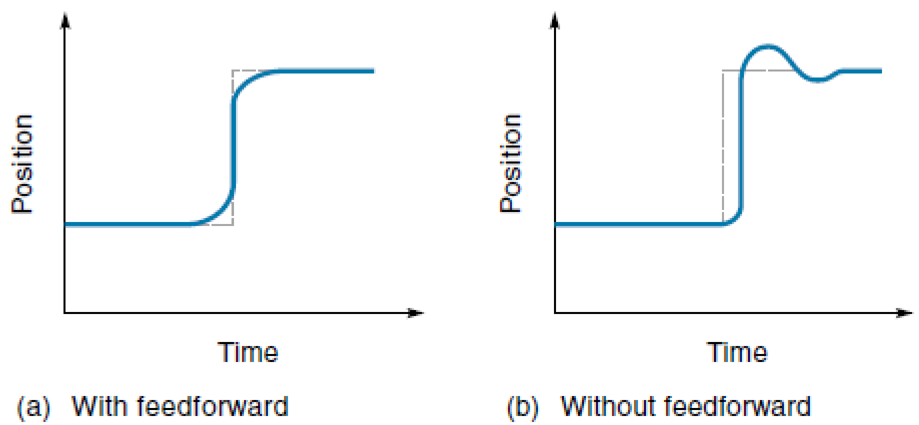
$$\text{Output} = K_P E + K_I K_P \Sigma(E\Delta t) + K_F(P_{T+1} - P_T)$$

KP = proportional gain constant
KI = integral gain constant
KF = feedforward gain constant
E = error (*SP* – *PV*)
PT = position it should be in *now*
PT + 1 = position it should be in, in the future (at *T* + 1)

PIP CONTROLLERS

- Notice that the feedforward term, *KF* (*PT* + 1 – *PT*), is proportional to the difference between where the controlled object is and where it must be in the future.

Improved path control with feedforward.



Sampling Rate: important Parameters

- In a digital control system, **sample rate** is the number of times per second a controller reads in sensor data and produces a new output value

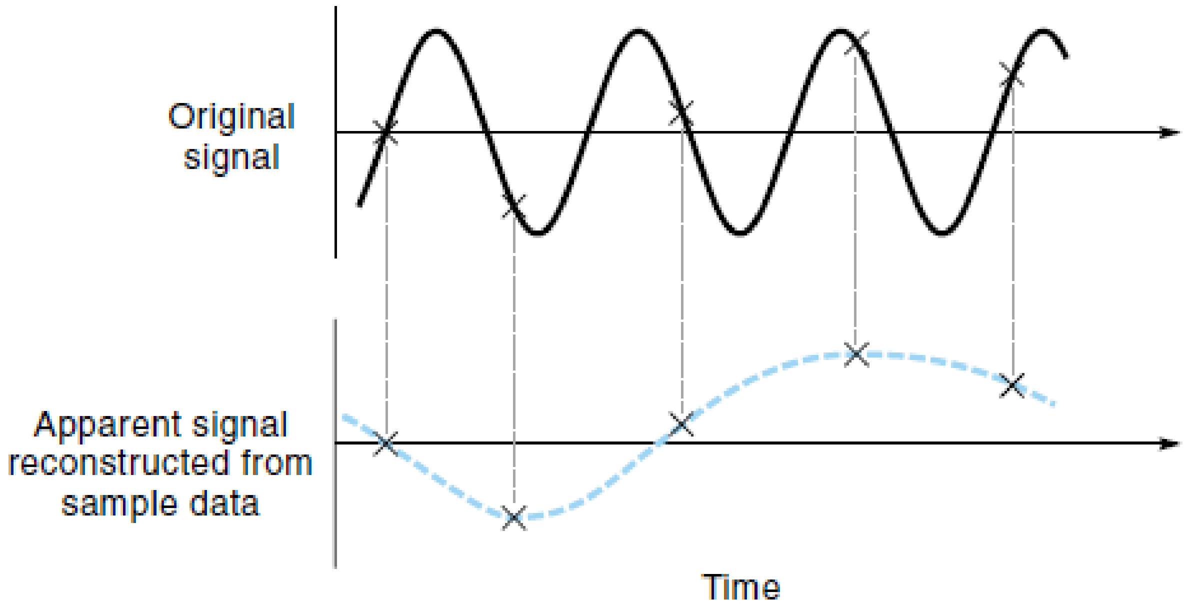
- the slower the sample rate, the less responsive the system is going to be because the controller would be always working with “old” data

- Shannon’s sampling theorem states that the sampling rate must be at least twice the highest frequency being monitored

Sampling Rate: important Parameter

Unsuitable sample time:

the sample rate was slightly less than twice the frequency



The End of this part