Control Elements

Applicable Mathematical Actions

Basic Control Actions

Control Element Configuration

We see that **basic** mathematical **operations** e.g. scaling, (**K**), integration, (**1/s**) and differentiation, (**s**), are **able** to achieve the **desired** performance **attributes.**

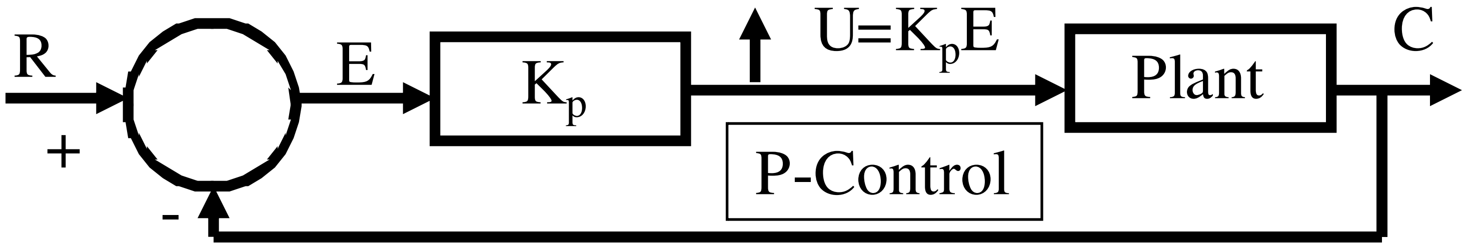
Therefore, we can set up **closed** loop control **systems** by incorporating these **mathematical** actions in the unity feedback control **structure.**

However, we need to **understand** their implications so that we can **link** these to complex control **objectives.**

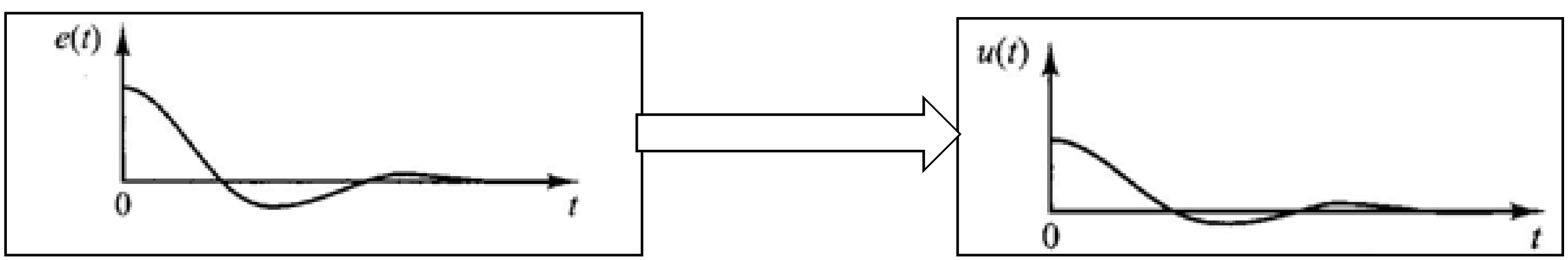
Basic Control Elements

Scaling or P- Control Action

P – control is defined by **H(s) = K_p**, a constant. The basic **structure** of the control **action** is as shown below.



This is termed **'Proportional'** control due to **input u(t)** being proportional to **error e(t),** as shown below.



P – Control Features

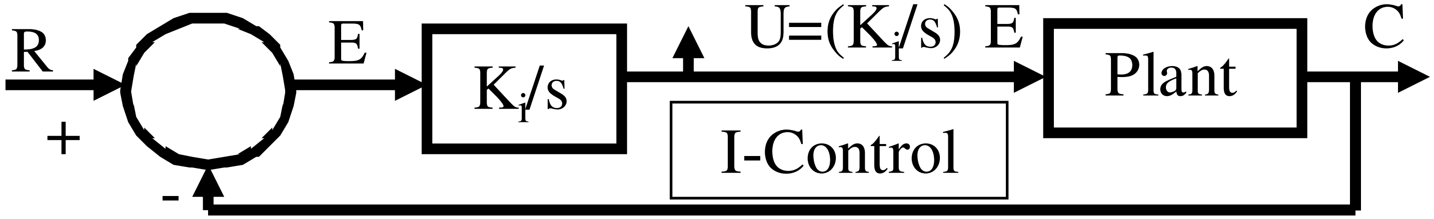
P – control remains **active** for non-zero values of **e(t)**. Further, **higher** error, leads to **larger** action.

It is also seen that if ‘$\mathbf{K_p}$’ is higher, **same** error results in **larger** control action, causing **tighter** control.

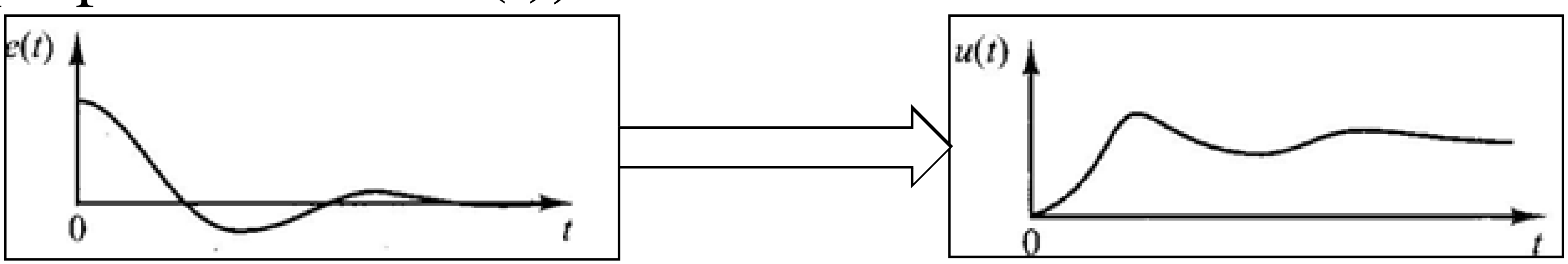
P – control is the **simplest** and hence, is **common** in most **situations** as it has the **ability** to achieve all **objectives.**

Integral or I- Control Action

I (or integral) **control** is defined by **H(s) = K_i/s.** Basic **structure** of such a control action is as shown below.



This is termed **'I' control**, due to **input u(t)** being proportional to $\int$**e(t),** as shown below.



I- Control Features

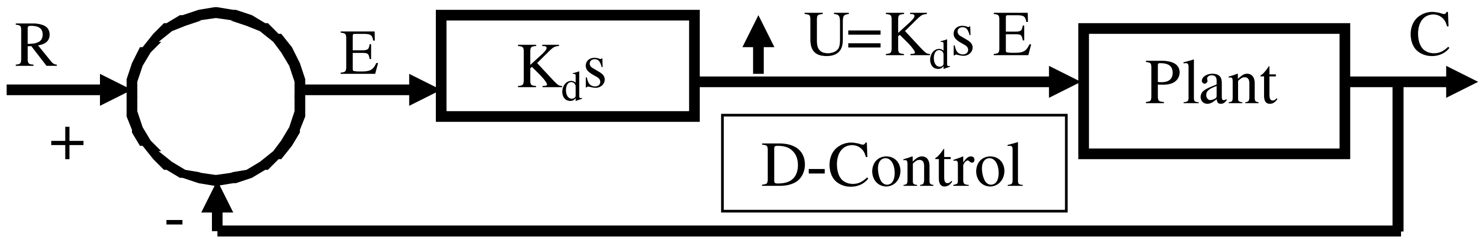
In this case, the **control** action **continues** till the **accumulated** error remains **non-zero.**

Thus, we find that control **action** continues long after the **instantaneous** error has gone to **zero** and hence, makes the **control** action more **appropriate** for tracking **task.**

However, it also takes a **long time** for u(t) to go to **zero,** thus **delaying** the steady-state.

Derivative or D- Control Action

D or derivative control action is defined by $\mathbf{H(s) = K_d\ s}$. Given below is the **schematic** of the above **action.**



In general, control **elemen**t with a **derivative** action results in a highly **sensitive** system because **u(t)** can be **non-zero** even if the **e(t)** is zero.

D- Control Features

In the context of **D-control**, the control action starts **even before** the error has time to **build-up.**

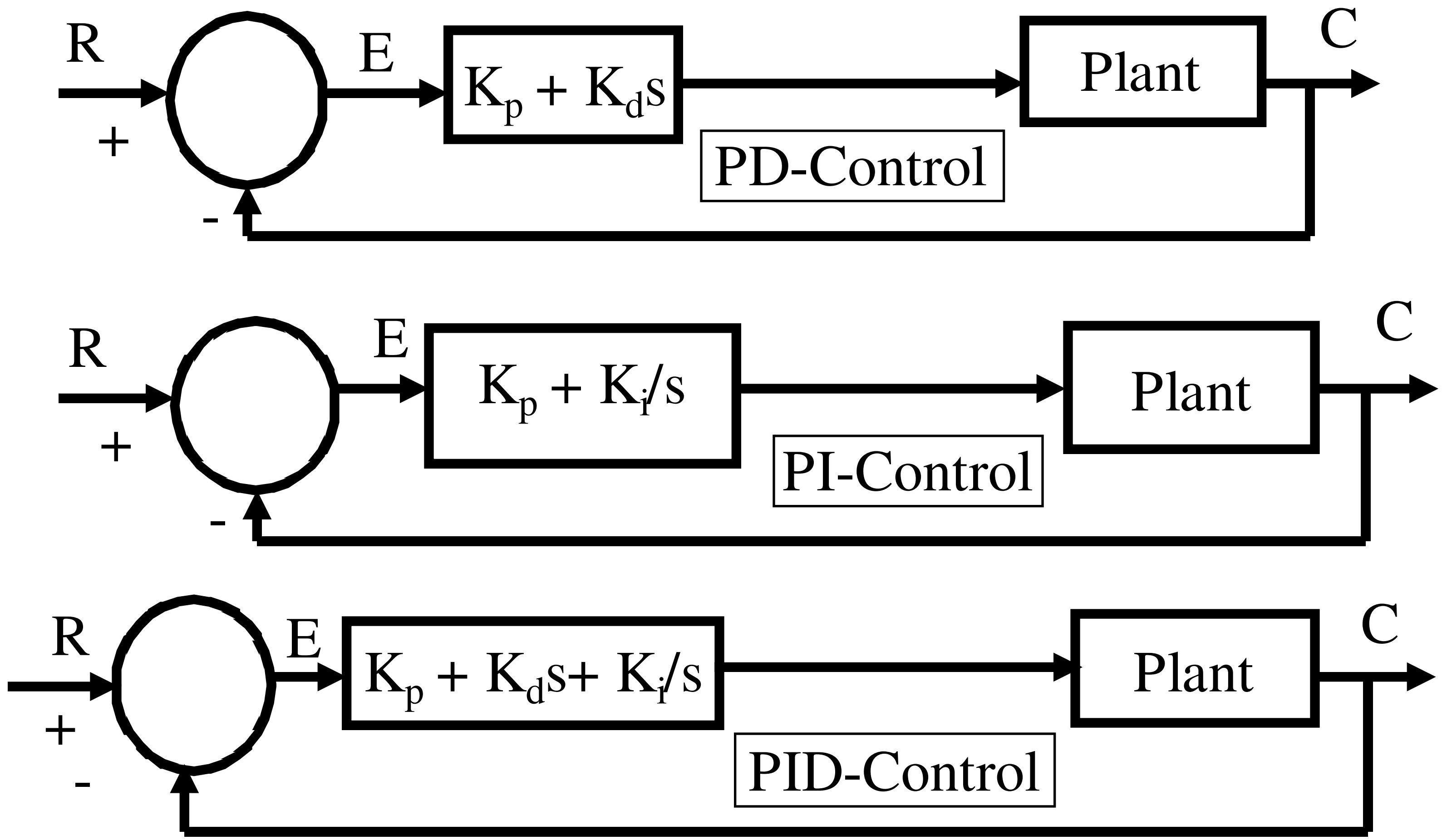
This is some form of **anticipation** that the system **acquires**, which does not **allow** error to **build-up** and, therefore, has the ability to reach the steady-state **faster.**

‘**D**’ control is, therefore, **ideal** for disturbance **rejection.**

However, there can be significantly **larger** control input at the **start**, which can also become **unbounded** in some situations and, hence, is generally **not desirable.**

Additional Control Actions

As **P, I & D** actions are **linear**, their **combinations** also result in **additional** linear control **actions**, as below.



Summary

P, D & I are the fundamental **building blocks** of control **elements**, which provide **adequate DOFs** for varied control **objectives.**