Chapter 5: Discrete Linear time invariant system controller design

1. Introduction

In the previous chapters we have discussed in details mainly the representation and modeling of discrete (sampled data) control systems, where the input-output or discrete transfer function representation is considered as the most important method to describe the behavior of such systems. This is because the knowledge of the transfer function is fundamental and basic for designing the appropriate controller which ensures the operation of the whole feedback control system with the required and desired performance.

This chapter will be devoted to discuss the discrete controller for the sampled data control system and we will be particularly interesting with the design of discrete proportional-integral-derivative (shortly named as PID) controller (regulator). We will focus on this type of controllers due to the fact that it is simple in structure and operation, familiar and extensively used everywhere.

2. Continuous time PID controller

Before tackling the subject of designing the discrete PID controller for a sampled data system, it is convenient to make brief review on continuous time PID controller.

The continuous time PID controller can be defined as an algorithm used to control or regulate a given physical quantity (variable) for a given process or system. It is the type of controller which is widely used in industrial processes, engineering and other fields. PID controller, also called three terms controller, is implemented using three actions:

- Proportional action (P).
- Integral action (I).
- Derivative action (D).

Using these three actions of control, different types and structures of control can be built such as: P, PI, PD and PID controllers.

In the subsequent sections, we consider the general input-output block diagram representation at the controller level as it is depicted in **Fig.5.1**.

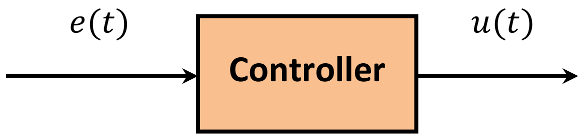


Fig.5.1 General block diagram of the controller input-output relation

With $e(t)$ and $u(t)$ are respectively the tracking error and the control signals.
We recall from the previous chapters that in time domain, the tracking error signal is defined in a unity feedback control system as the difference between the reference signal $r(t)$ and the actual output signal $y(t)$. Mathematically, this is defined by:

$$e(t) = r(t) - y(t) \tag{5.1}$$

2.1 Proportional (P) controller

The control law generated using the proportional (P) controller is defined by the following relation:

$$u(t) = K_p e(t) \tag{5.2}$$

With K_p is called the proportional constant (parameter) which is specific designation to the proportional-type controller.
In frequency domain, the transfer function representing P-type controller can be obtained by applying Laplace transform on both sides of the relation (5.2). The result is mentioned by:

$$G_P(s) = \frac{U(s)}{E(s)} = K_p \tag{5.3}$$

With :
$C_P(s)$: is the P-type controller transfer function.
$E(s)$ and $U(s)$ are respectively Laplace transform of the input and output of the controller. We can also represent the P-controller by the following general block diagram depicted in **Fig.5.2**.

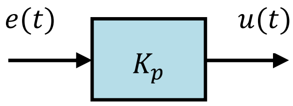


Fig.5.2 General block diagram representation of P-type controller

2.2 Proportional-Integral (PI) controller

The proportional-integral, which is abbreviated by PI and known as PI-type controller is a control algorithm based on two simultaneous actions; proportional action and integral action. Using this controller, the control signal is generated according to the following law (relation):

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\, d\tau \tag{5.4}$$

With:

K_p: is the proportional action constant (coefficient).

K_I: is the integral action constant (coefficient).

In some times (depending on the application), the PI control action is governed by:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau)\, d\tau \right] \tag{5.5}$$

With:

T_i: is called the integral time constant and is related to the integral constant by :

$$K_i = \frac{K_p}{T_i} \tag{5.6}$$

We can obtain the transfer function of PI controller by simply applying Laplace transform on both sides of either (5.4) or (5.5). In other words, we can have two transfer function configurations.

Using (5.4), the corresponding PI controller transfer function is given to be:

$$G_{PI}(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \tag{5.7}$$

Regarding the relation (5.6), another form of PI controller transfer function can also be obtained and given by:

$$G_{PI}(s) = \frac{U(s)}{E(s)} = K_p . \frac{1 + T_i s}{T_i s} \tag{5.8}$$

Accordingly, two block diagram representations can be given to the PI controller respective to the couple of the above relations. These are given respectively in **Fig.5.3 (a)** and **Fig.5.3 (b)**.

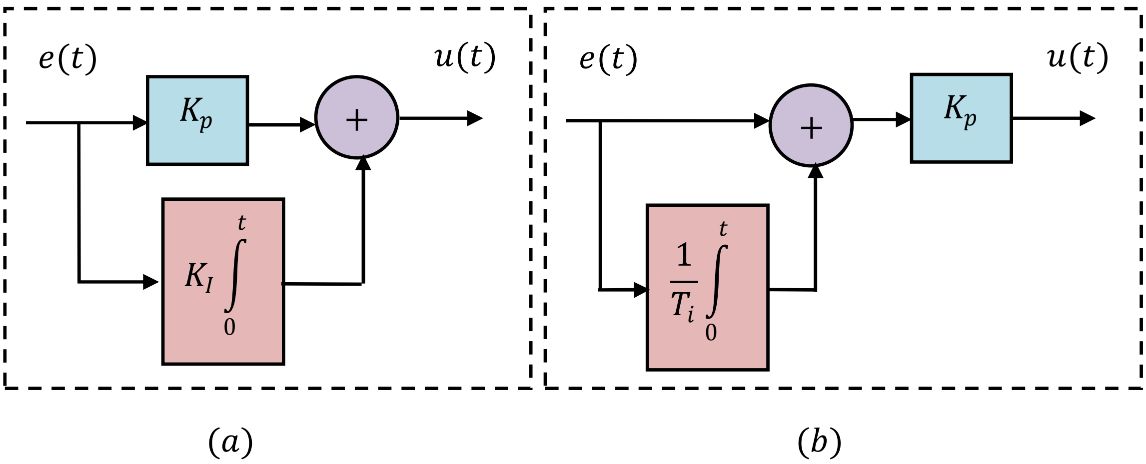


Fig.5.3 PI controller block diagram two possible representations

2.3 Proportional-Derivative (PD) controller

Among the possible combination between the three control actions P, I and D is that of combining the proportional action (P) with the derivative action (D) to form the Proportional-Derivative (PD) controller. For this controller type, the control law (signal) is generated using the following relationship between the tracking error, $e(t)$, as input to the controller and the control (actuating) signal, $u(t)$, as the output of the controller:

$$u(t) = K_p e(t) + K_d \frac{d[e(t)]}{dt} \tag{5.9}$$

with:

K_p and K_d are respectively the proportional action constant and the derivative action constant.

The expression (5.9) can also be rewritten under the following form:

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} \right] \tag{5.10}$$

With :

T_d : is called the derivative time constant, which is defined according to the following relation:

$$T_d = \frac{K_d}{K_p} \tag{5.11}$$

Using transfer function representation, PD controller can be input-output modeled by the following respective transfer functions:

$$G_{PD}(s) = \frac{U(s)}{E(s)} = (K_p + K_d s) \tag{5.12}$$

And:

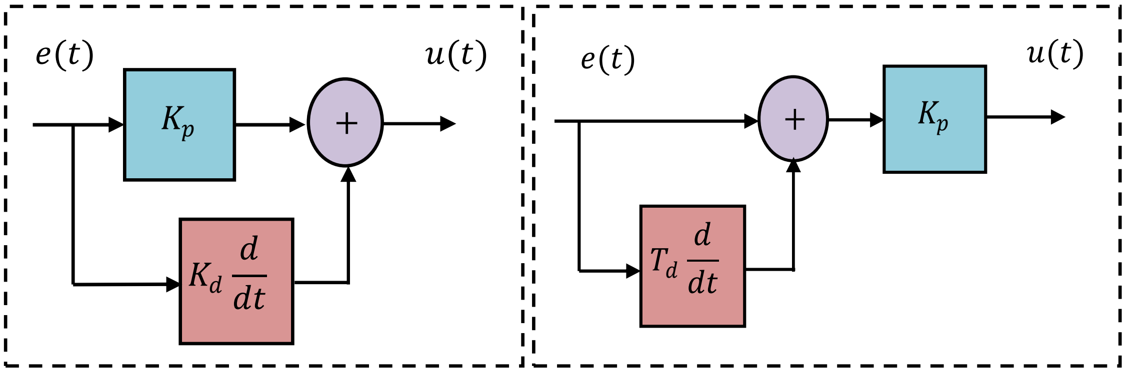
$$G_{PD}(s) = \frac{U(s)}{E(s)} = K_p(1 + T_d s) \tag{5.13}$$

With :

$$T_d = \frac{K_d}{K_p} \tag{5.14}$$

Is called the derivative time constant.

When using block diagram representation, PD controller operation is also illustrated as shown in **Fig.5.4.**



(a) (b)

Fig.5.4 PD controller block diagram two possible representations

2.4 Proportional-Integral-Derivative (PID) controller

Proportional-integral-Derivative, abbreviated by PID, controller is the control algorithm that involves the contribution of all three control actions, P, I and D. as for the previous types, PID controller can be implemented using several and different structures and architectures. In this section, we shall briefly discuss three of them, namely:

- Series PID configuration.
- Parallel PID configuration.
- Mixed PID configuration.

2.4.1 Series PID controller configuration

Series PID controller configuration is the structure of the controller where the proportional, integral and derivative actions are connected in series as it is mentioned and represented by the following block diagram of **Fig.5.5**.

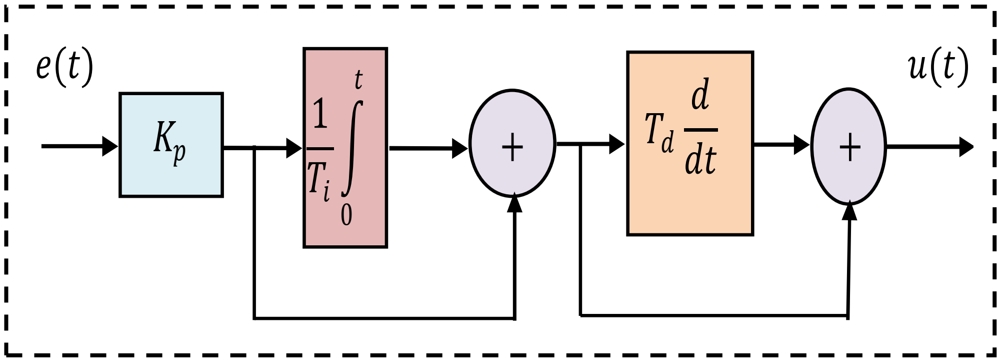


Fig.5.5 block diagram representation for series configuration of PID controller

The time domain relationship between the tracking error as input and the actuating (control) signal as the output of the series configuration of PID controller is given as follows:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

$$= K_p \left[\frac{T_i + T_d}{T_i} e(t) + \frac{1}{T_i} \int_0^t e(\tau)\, d\tau + T_d \frac{de(t)}{dt} \right] \tag{5.15}$$

By applying Laplace transform on both sides of (5.14), the frequency domain representation describing the behavior of the PID controller under its series structure implementation is given by the following continuous time transfer function:

$$G_{PID_s}(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s}\right)(1 + T_d s) \tag{5.16}$$

2.4.2 Parallel PID controller configuration

The PID controller can also be implemented using a parallel structure where the three actions Proportional, integral and derivative of the controller are connected in parallel structure as it is depicted in **Fig.5.6**.

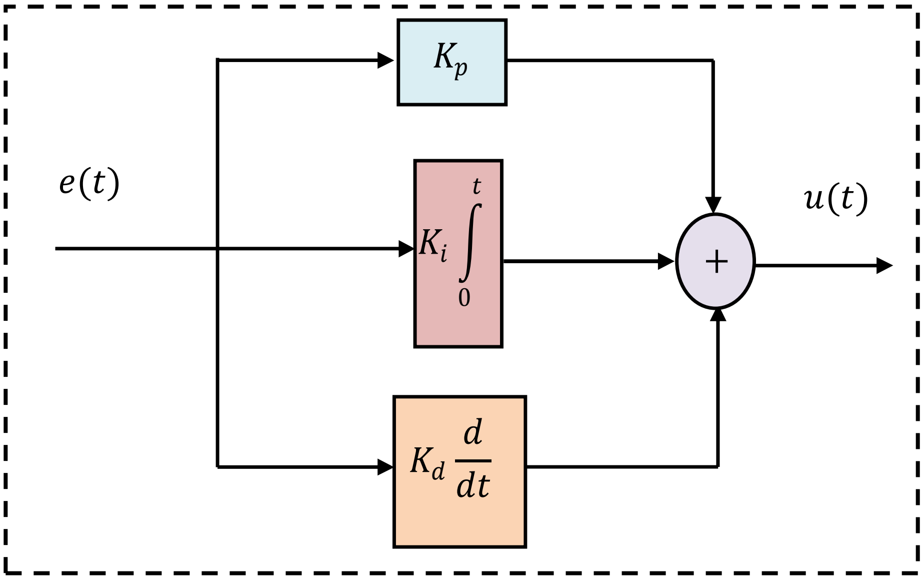


Fig.5.6 block diagram representation for parallel configuration of PID controller

The control law that governs the operation of parallel configuration of PID controller is expressed in the time domain by the following relationship between the input and the output of the controller:

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt} \tag{5.17}$$

From (5.16) and using Laplace transform, the corresponding transfer function representing and describing the behavior of the controller is obtained and given by:

$$G_{PID_p}(s) = \frac{U(s)}{E(s)} = K_p + K_i \frac{1}{s} + K_d s \tag{5.18}$$

Obviously, the parallel structure of PID controller is simpler and direct compared with the series configuration. Therefore, as long as the implementation of the controller is considered, the parallel structure is preferred in most of the industrial cases [7].

2.4.3 Mixed PID controller configuration

The mixed structure used to implement PID controller combines between both series and parallel connection of the proportional, integral and derivative control actions. A typical block diagram of this PID structure is shown in **Fig.5.7**.

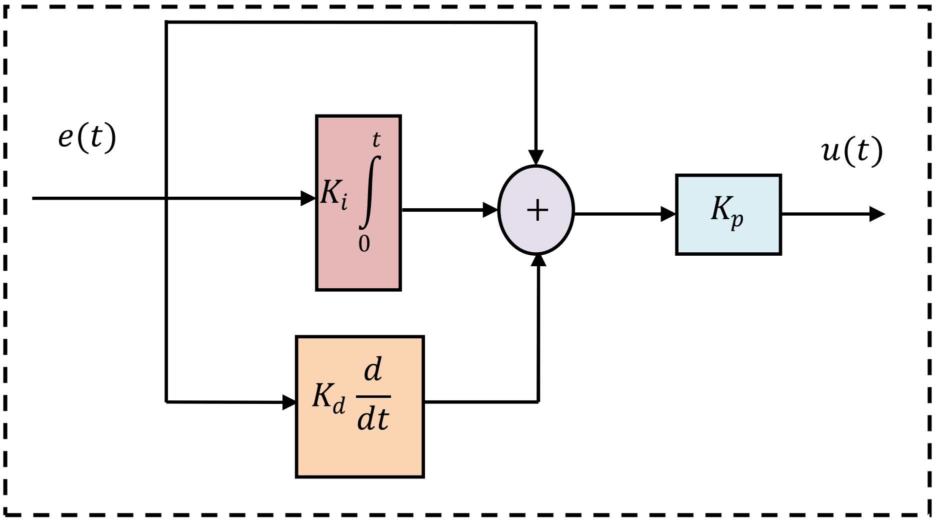


Fig.5.7 block diagram representation for mixed configuration of PID controller

The input-output law that describes the generation of the control signal by the mixed configuration of PID controller is expressed in the time domain by:

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau)\, d\tau + T_d \frac{de(t)}{dt} \right] \quad (5.19)$$

In the frequency domain, the continuous time transfer function which is representing the behavior of mixed structure PID controller is obtained by using Laplace transform mathematical tool to be written as:

$$G_{PID_{mix}}(s) = \frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right) \quad (5.20)$$

3. Discrete PID controller design

Among the direct and straightforward method of designing a discrete (digital) PID controller is to discretize the already implemented continuous time PID controller by using one the discretization methods discussed previously in chapter 3. The general mechanism can be explained and illustrated by the following diagram of **Fig.5.8**.

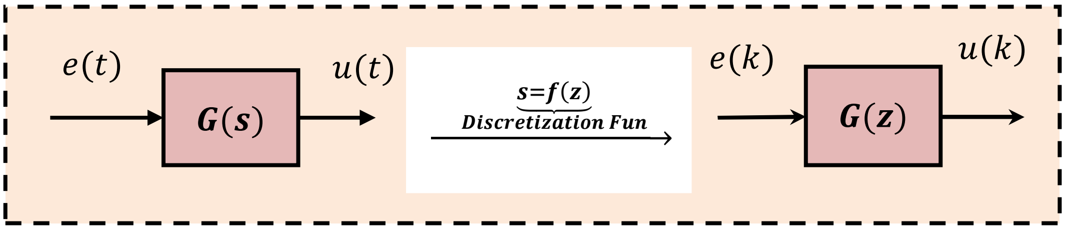


Fig.5.8 Block diagram representation showing discretization process

Where:

$G(s)$ and $G(z)$ represent respectively the continuous time and discrete transfer function of the corresponding controllers.

$s = f(z)$: is the used discretizing function, which corresponds to one of the discretization methods.

3.1 Discretization methods of continuous time derivative and integral

The basic principle employed to design the discrete (digital) PID controller by discretizing a continuous time controller consists of approximating the integral and derivative terms. These approximations consist, in fact, of converting the analog integral and derivative actions into discrete functions with respect to a predefined sampling period T_s.

In this section, we shall focus on using the more useful approximation (discretization) methods to obtain the discrete versions of the analog (I) and (D). Namely, this concern:

- Forward Euler's approximation method.
- Backward Euler's approximation method.
- Tustin approximation method.

In the following table (**Table 5.1**), we summarize the use of these approximation methods to obtain the equivalent z (discrete) transfer functions of integral (I) and derivative (D) parts of the PID controller.

Table 5.1 Discretization of integral (I) and derivative (D) of PID controller [8]

	Integral term (I)		**Derivative term (D)**	
	Continuous	**discrete**	**Continuous**	**discrete**
Forward Euler's approximation method	$\frac{1}{s}$	$\frac{z}{(z-1)}$	s	$\frac{(z-1)}{z}$
Backward Euler's approximation method	$\frac{1}{s}$	$\frac{T_s z}{(z-1)}$	s	$\frac{(z-1)}{T_s z}$
Tustin approximation method	$\frac{1}{s}$	$\frac{T_s}{2}\frac{(z+1)}{(z-1)}$	s	$\frac{2}{T_s}\frac{(z-1)}{(z+1)}$
With : T_s represents the used sampling time period				

4. Discrete (Digital) PID controller

Now it is easier to design and implement any variant of the different discrete PID controller types by just choosing and applying one of the above discretization methods. In the subsequent sections, we will be interesting of using the approximation method of forward Euler.

4.1 Equivalent Digital Proportional (P) controller

The digital P-type controller is fully defined and described by its discrete transfer function. Since the behavior of continuous time P-type controller is represented by the transfer function:

$$G_P(s) = \frac{U(s)}{E(s)} = K_p \tag{5.21}$$

The equivalent z transfer function is obtained by taking Z transform of $G(s)$, therefore:

$$G_P(z) = Z\{G_P(s)\} = \frac{Z\{U(s)\}}{Z\{E(s)\}} = \frac{U(z)}{E(z)} = Z\{K_p\} = K_p \tag{5.22}$$

Obviously, analog and digital P-type controllers have the same transfer function as no approximation can be applied to discretize the proportional (P) term of the controller.

In time domain representation, the digital proportional controller algorithm is written according to the following relationship between the input and output samples:

$$u(k) = K_p e(k) \tag{5.23}$$

If the discrete transfer function of the controller is known, the digital P-type controller can also be represented by the following functional block diagram shown in **Fig.5.9**.

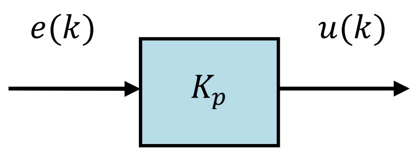


Fig.5.9 General block diagram representation of digital P-type controller

4.2 Equivalent Digital Proportional-Integral (PI) controller

If we consider that the continuous time PI controller is described by its transfer function mentioned in (5.8), which is rewritten for convenience as:

$$G_{PI}(s) = \frac{U(s)}{E(s)} = K_p.\frac{1+T_i s}{T_i s} \tag{5.24}$$

By applying Euler's forward discretization formula, we can get:

$$G_{PI}(z) = G_{PI}(s)\Big|_{s=\frac{(z-1)}{z}} = \frac{U(z)}{E(z)} = K_p.\left(1 + \frac{1}{T_i\left[\frac{(z-1)}{z}\right]}\right)$$

Finally, the equivalent z transfer function of digital PI-type controller is expressed as:

$$G_{PI}(z) = K_p.\left(1 + \frac{1}{T_i}\frac{z}{z-1}\right) \tag{5.25}$$

Accordingly, the control law (signal) generated by the controller at the time instant 'k' is expressed by the following time domain relationship.

$$u(k) = u(k-1) + K_p.\left(1+\frac{1}{T_i}\right)e(k) - K_p e(k-1) \tag{5.26}$$

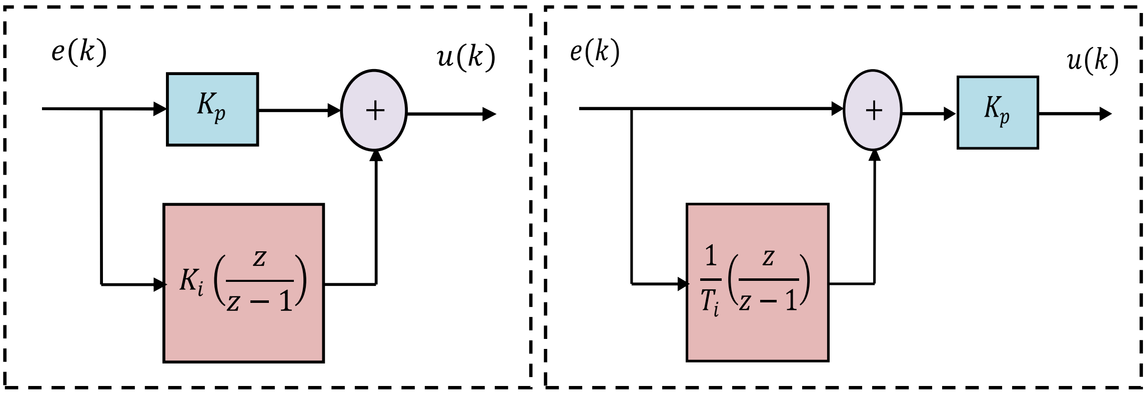
On the other hand, if the analog PI controller is represented by the second variant of transfer function given as :

$$G_{PI}(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_i}{s} \tag{5.27}$$

In this case, when discretizing using forward Euler' formula, the digital PI controller is obtained to be represented by the z transfer function given by:

$$G_{PI}(z) = G_{PI}(s)\big|_{s=\frac{(z-1)}{z}} = K_p + K_i\left(\frac{z}{z-1}\right) \tag{5.28}$$

The corresponding two possible block diagram representations of the above z transfer functions of a typical digital PI controller are shown in **Fig.5.10**.



(a) (b)

Fig.5.10 Digital PI controller block diagram two possible representations

4.3 Equivalent Digital Proportional-Derivative (PD) controller

We have described earlier the continuous time PD controller by two forms of transfer function which we rewrite here for convenience as:

$$G_{PD}(s) = \frac{U(s)}{E(s)} = (K_p + K_d s) \tag{5.29}$$

And:

$$G_{PD}(s) = \frac{U(s)}{E(s)} = K_p(1 + T_d s) \tag{5.30}$$

By discretizing using forward Euler's formula, the corresponding equivalent two possible discrete transfer function of a typical digital PD controller can be obtained to be:

$$G_{PD}(z) = \frac{U(z)}{E(z)} = G_{PD}(s)\Big|_{s=\frac{(z-1)}{z}} = \left(K_p + K_d\frac{(z-1)}{z}\right) \qquad (5.31)$$

And:

$$G_{PD}(z) = \frac{U(z)}{E(z)} = G_{PD}(s)\Big|_{s=\frac{(z-1)}{z}} = K_p\left(1 + T_d\frac{(z-1)}{z}\right) \qquad (5.32)$$

We can express the control law used to generate the control signal of the above two structures of the digital PD controller by just applying inverse z transform on the respective transfer functions (5.31) and (5.32). The result will be respectively as below:

$$u(k) = (K_p + K_d)e(k) - K_d e(k-1) \qquad (5.33)$$

And :

$$u(k) = (K_p + K_pT_d)e(k) - K_pT_d e(k-1) \qquad (5.34)$$

Using block diagram representation, these two configurations of digital PD controller can represented as it is depicted in **Fig.5.11.**

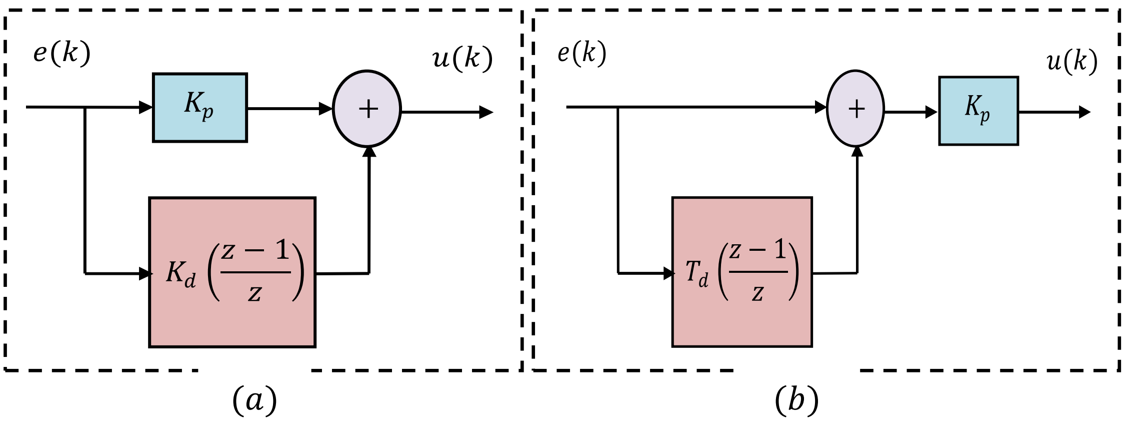


Fig.5.11 Digital PD controller block diagram two possible representations

4.4 Equivalent Digital Proportional-Integral-Derivative (PID) controller

As it was previously discussed in section 2.4, continuous time PID controller can be implemented using three different structures. Similarly, equivalent digital PID controller is also being designed under the same of these configurations; namely:

- Series digital PID configuration.
- Parallel digital PID configuration.
- Mixed digital PID configuration.

In the following subsections, we shall interest to deriving the different time domain, frequency domain and block diagram representations corresponding to each configuration.

4.4.1 Equivalent Series configuration of digital PID controller

For convenience, we rewrite the transfer function of the series structure of the analog PID controller as:

$$G_{PID_s}(s) = \frac{U(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s}\right)(1 + T_d s) \qquad (5.35)$$

Applying forward Euler's discretization formula, the discrete series configuration PID controller transfer function is obtained as:

$$G_{PID_s}(z) = \frac{U(z)}{E(z)} = G_{PID_s}(s)\Big|_{s=\frac{(z-1)}{z}}$$

$$G_{PID_s}(z) = K_p\left[1 + \frac{1}{T_i\frac{(z-1)}{z}}\right]\left[1 + T_d\frac{(z-1)}{z}\right] \qquad (5.36)$$

As long as the control law is concerned, the application of inverse z transform tool allows us to express the control signal generated at the discrete time instant 'k' by the following relationship between the input and the output samples of the controller.

$$u(k) = u(k-1) + C_0 e(k) + C_1 e(k-1) + C_2 e(k-2) \qquad (5.37)$$

Where :

C_0, C_1, C_2: are known constant coefficients and are respectively defined by :

$$\begin{cases} C_0 = \dfrac{K_p}{T_i}[(T_i + T_d) + 1 + T_i T_d] \\ C_1 = -\dfrac{K_p}{T_i}[(T_i + T_d) + 2T_i T_d] \\ C_2 = K_p T_d \end{cases}$$

We can also give the block diagram representation of this digital PID structure by direct implementation of the respective z transfer function expression (5.36). this is shown in **Fig.5.12**.

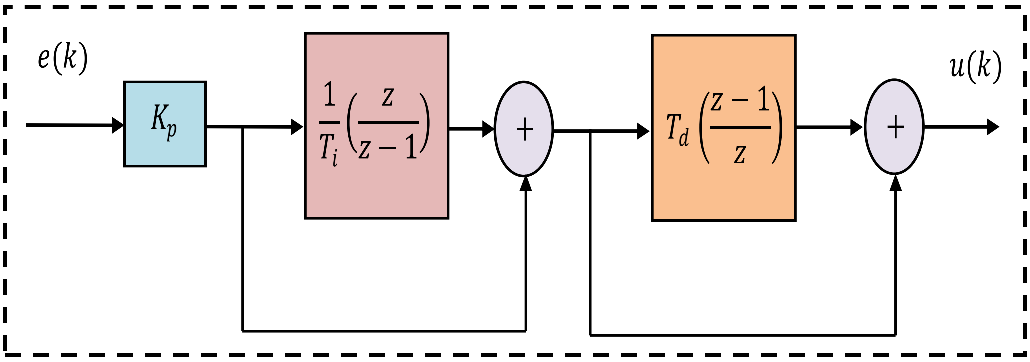


Fig.5.12 Block diagram representation for series configuration digital PID controller

4.4.2 Equivalent parallel configuration of digital PID controller

By referring to transfer function expression derived for parallel structure analog PID controller in section 2.4.2, and using the discretization formula of forward Euler's method, the equivalent z transfer function of this PID structure can be obtained as follows:

$$G_{PID_p}(z) = \frac{U(z)}{E(z)} = G_{PID_p}(s)\Big|_{s=\frac{(z-1)}{z}}$$

$$G_{PID_p}(z) = K_p + K_i\left(\frac{z}{z-1}\right) + K_d\left(\frac{z-1}{z}\right) \qquad (5.38)$$

The corresponding control law responsible of generating the control signal can be derived from the expression of the discrete transfer function (5.38) using the inverse z transform. After some algebraic manipulations, this control law is described by the following relationship.

$$u(k) = u(k-1) + [K_p + K_i + K_d]e(k)$$

$$-[K_p + 2K_d]e(k-1) + K_d e(k-2) \qquad (5.39)$$

The control law described by (5.39) is also called the difference equation representation of the parallel structure digital PID controller.

The block diagram representing the operation and behavior of this PID controller structure can be drawn from the controller's z transfer function (5.38) and it is shown in **Fig.5.13**.

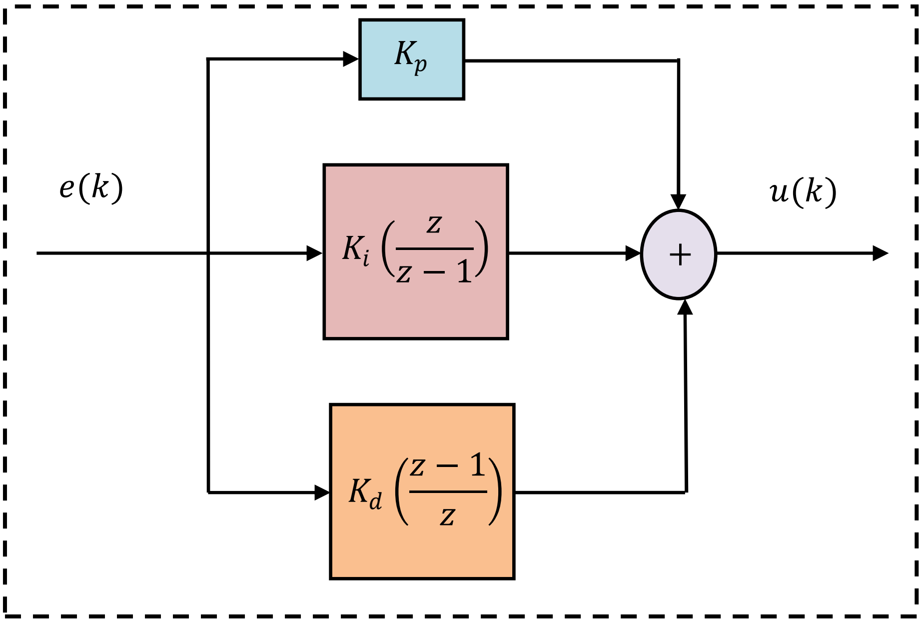


Fig.5.13 Block diagram representation for parallel configuration digital PID controller

4.4.3 Equivalent mixed configuration of digital PID controller

The mixed structure of digital PID controller combines the series and parallel structures exactly as it is done for the case of analog PID controller. Consequently, the equivalent discrete transfer function representing the implementation of this controller structure is directly obtained by discretization. Using the forward Euler's formula of discretization, the equivalent z transfer function of mixed structure digital PID controller is given as:

$$G_{PID_{mix}}(z) = \frac{U(z)}{E(z)} = G_{PID_{mix}}(s)\Big|_{s=\frac{(z-1)}{z}}$$

$$G_{PID_{mix}}(z) = K_p\left[1 + \frac{1}{T_i}\left(\frac{z}{z-1}\right) + T_d\left(\frac{z-1}{z}\right)\right] \qquad (5.40)$$

Using inverse z transform applied on transfer function described by (5.40), the corresponding difference equation representing the controller in the time domain can be obtained and written as:

$$u(k) = u(k-1) + C_0 e(k) + C_1 e(k-1) + C_2 e(k-2) \qquad (5.41)$$

Where C_0, C_1, C_2: are known constant coefficients and are respectively defined by :

$$\begin{cases} C_0 = K_p\left(1 + \frac{1}{T_i} + T_d\right) \\ C_1 = -K_p(1 + 2T_d) \\ C_2 = K_p T_d \end{cases}$$

5. Effect PID controller constants on the control system's performance

We end up this discussion about the design of the equivalent discrete (digital) PID controller by stating how the performance of the discrete (sampled data) feedback control system is affected by the choice of the parameters (constants), K_p, K_i and K_d, which characterize respectively the proportional, integral and derivative actions of PID controller. The effect of these constants is individually summarized in the following table (**Table 5.2**).

Table 5.2 Effect of Changing Independently PID Parameters on System Response

Closed Loop Response	***Rise Time***	***Overshoot***	***Settling Time***	***Steady State Error***	***Stability***
Increasing K_P	decrease	Increase	Small Increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large Decrease	Degrade
Increasing K_D	Small Decrease	decrease	Decrease	Minor Change	Improve

References

1. John J. D'Azzo, Constantine H. Houpis, Stuart N. Sheldon, « ***Linear control System analysis and design with MATLAB***", Fifth Edition, Marcel Dekker, Inc., NewYork, U.S.A., 2003.

2. Charles L. Phillips, H. Troy Nagle, "***Digital Control System Analysis and Design***", 3rd edition, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

3. Rao V. Dukkipati "***Analysis and Design of Control Systems using MATLAB***", New Age International (P) Ltd., Publishers, New Delhi, 2006.

4. M. Gopal, "***Digital Control and State Variable Methods- Conventional and Neuro-Fuzzy Control Systems***", 2nd edition, Tata McGraw Hill Publishing Company Limited, New Delhi, 2006.

5. Gregory P. Starr, « ***Introduction to Applied Digital Control*** », Second Edition, Department of Mechanical Engineering, The University of New Mexico, November 2006.

6. M. Sami Fadali, Antonio Visioli, « ***Digital Control Engineering, Analysis and Design***", Second Edition, Elsevier Inc., U.S.A., 2013.

7. David I. Wilson, **« *Advanced Control using* *MATLAB: or* *Stabilizing the Unstabilisable* »,** Auckland University of Technology, New Zealand, July 27, 2016.

8. Katsuhiko Ogata, "***Discrete time control systems***", 2nd Edition, Prentice Hall, Englewood Cliffs, New Jersey, 1995.