Distributed & Embedded Real-Time Systems (0640751)

Lecture: 10

Real-Time Implementation of Digital Algorithms

Prof. Kasim M. Al-Aubidy
Philadelphia University

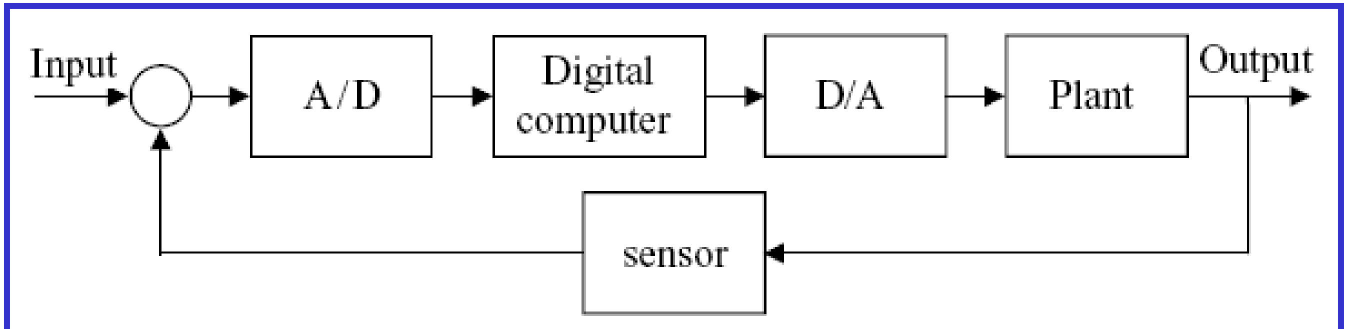
Course Objectives:

The main objective of this unit is to:

- Consider the methods used to implement simple digital algorithms.
- Study, analyze and implement PID control algorithm.
- Solve problems that arise in implementing such algorithms in real-time applications.
- Choice suitable sampling rates.
- Study different realization methods.

Sampled Data Systems:

- A sampled data system operates on discrete-time rather than continuous-time signals. A digital computer is used as the controller in such a system. A D/A converter is usually connected to the output of the computer to drive the plant. We will assume that all the signals enter and leave the computer at the same fixed times, known as the sampling times.
- The digital computer performs the controller or the compensation function within the system. The A/D converter converts the error signal, which is a continuous signal, into digital form so that it can be processed by the computer.
- At the computer output the D/A converter converts the digital output of the computer into a form which can be used to drive the plant.



Sampled data control system

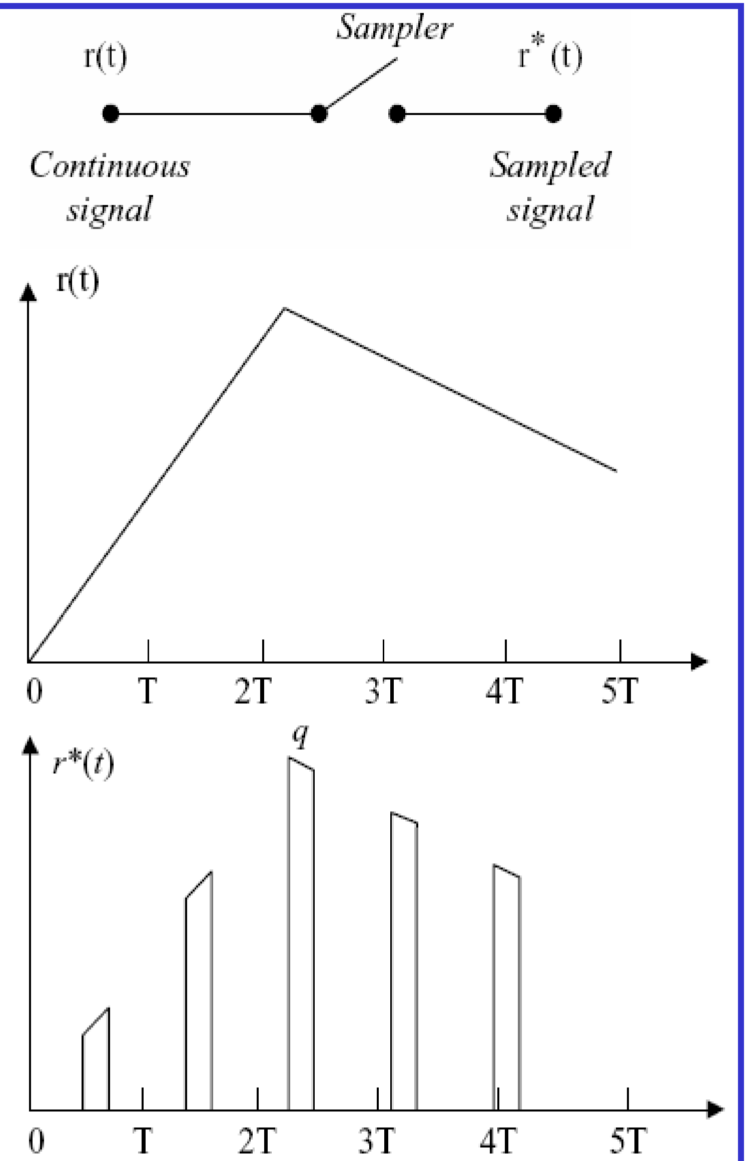
The Sampling Process:

- A sampler is basically a switch that closes every T seconds.
- When a continuous signal $r(t)$ is sampled at regular intervals T, the resulting discrete-time signal is shown, where q represents the amount of time the switch is closed.
- In practice the closure time q is much smaller than the sampling time T, and the pulses can be approximated by flat-topped rectangles.
- In control applications the switch closure time q is much smaller than the sampling time T and can be neglected.
- The ideal sampling process can be considered as the multiplication of a pulse train with a continuous signal, i.e.

$$r^*(t) = P(t)r(t),$$

$$P(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT);$$

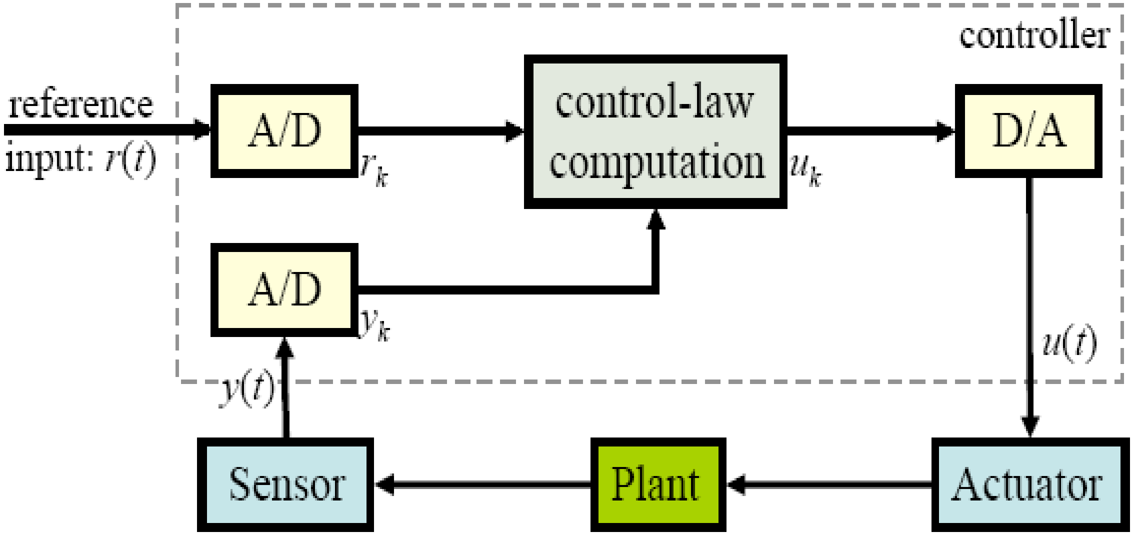
$$r^*(t) = r(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$$



The signal $r(t)$ after the sampling operation

PID Control Algorithm:

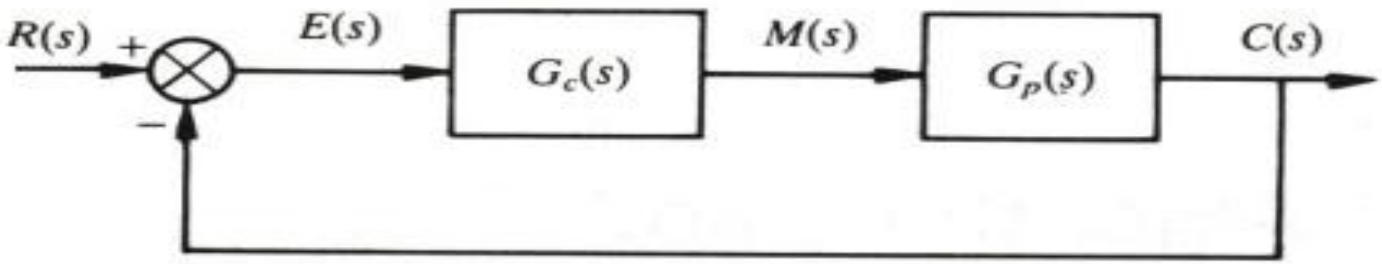
- The Proportional Integral Derivative (PID) control is a simple, three-term controller used in industry.
- The differential equation for a PID controller is :

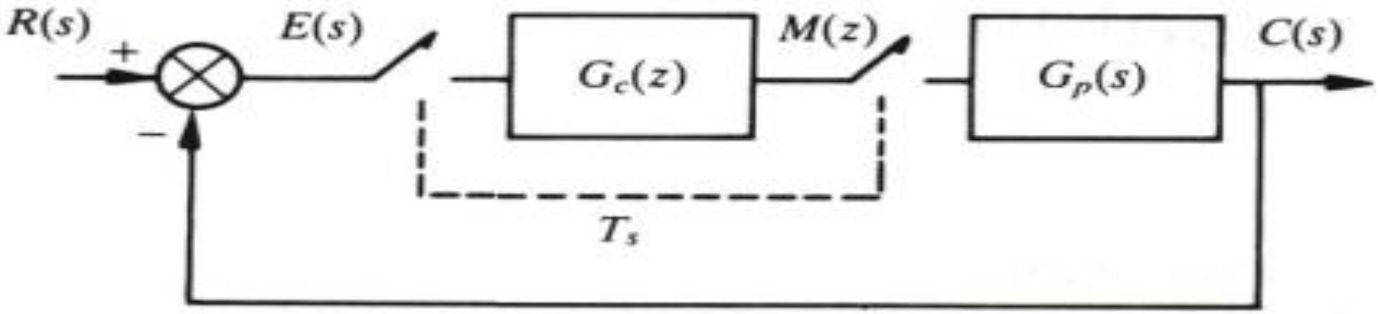


$$m(t) = K_p[e(t) + 1/T_i \int_0^t e(t)dt + T_d de(t)/dt] \qquad (1)$$

$$e(t) = r(t) - c(t)$$

$$G_c(s) = \frac{M(s)}{E(s)} = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \qquad (2)$$





- Both the time domain and frequency domain representation are continuous representation.
- To implement the controller using a a digital algorithm, it is required to convert from a continuous to discrete representation of the controller. There are several methods to doing this, such as;
  - First-order finite differences.
  - Z-transform.

Using 1st-order finite differences:

- Considering the time domain version of the PID controller, replace the differential and integral terms by their discrete equivalents by using:

$$\frac{df}{dt}\bigg|_k = \frac{f_k - f_{k-1}}{\Delta t}, \quad \int e(t)dt = \sum_{k=1}^{n} e_k \Delta t \qquad (3)$$

and hence equation 1 becomes

$$m(n) = K_p \left[T_d \left(\frac{e(n) - e(n-1)}{\Delta t} \right) + e(n) + \frac{1}{T_i} \sum_{k=1}^{n} e_k \Delta t \right] \qquad (4)$$

By introducing new parameters as follows:

$$K_i = K_p (T_s / T_i)$$
$$K_d = K_p (T_d / T_s)$$

$T_s = \Delta t =$ the sampling interval, equation 4 can be expressed as an algorithm of the form

$$s(n) = s(n-1) + e(n)$$
$$m(n) = K_p e(n) + K_i s(n) + K_d [e(n) - e(n-1)] \qquad (5)$$

$s(n)$= sum of the errors taken over the interval 0 to nT_s

Position and Velocity Algorithms:

- The digital control law given by equation 5 is referred to as the positional algorithm, because it is used to calculate the absolute value of the actuator position.
- The velocity algorithm is an alternative form of the PID control algorithm, and it is widely used to provide automatic bumpless transfer. This algorithm gives the change in the value of the manipulated variable at each sample time.

$$\frac{dm(t)}{dt} = K_p\left(\frac{de(t)}{dt} + \frac{1}{T_i}e(t) + T_d\frac{d^2e(t)}{dt^2}\right) \qquad (7)$$

$$\begin{aligned}\Delta m &= m(n) - m(n-1)\\ &= K_p\Big([e(n) - e(n-1)] + \frac{\Delta t}{T_i}e(n) +\\ &\quad \frac{T_d}{\Delta t}[e(n) - 2e(n-1) + e(n-2)]\Big)\end{aligned} \qquad (8)$$

$$\begin{aligned}\Delta m(n) = K_p\Big[&\left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s}\right)e(n) -\\ &\left(1 + 2\frac{T_d}{T_s}\right)e(n-1) + \frac{T_d}{T_s}e(n-2)\Big]\end{aligned} \qquad (9)$$

$$K_1 = K_p(1 + T_s/T_i + T_d/T_s)$$
$$K_2 = -K_p(1 + 2T_d/T_s)$$
$$K_3 = K_p T_d/T_s$$

$$\Delta m(n) = K_1e(n) + K_2e(n-1) + K_3e(n-2) \qquad (10)$$

Comparison of Position and Velocity Algorithms:

- Comparing the position algorithm (equ 5) and the velocity algorithm (equ 10) shows that:
- The velocity algorithm is simpler to program (PROVE THAT).
- The velocity algorithm is inherently safer in that large changes in demanded actuator position are unlikely to occur.

$$\Delta m(n) = K_p\left([c(n-1) - c(n)] + \frac{T_s}{T_i}(r - c(n)) + \frac{T_d}{T_s}[2c(n-1) - c(n-2) - c(n)]\right) \qquad (11)$$

$$\begin{aligned} s(n) &= s(n-1) + e(n) \\ m(n) &= K_p e(n) + K_i s(n) + K_d[e(n) - e(n-1)] \end{aligned} \qquad (12)$$

$$\begin{aligned} s(n) &= s(n-1) + e(n)(T_s/T_i) \\ m(n) &= K_p e(n) + K_p s(n) + K_d[e(n) - e(n-1)] \end{aligned} \qquad (13)$$

The PID Controller: Z-Transform:

- The PID controller can be expressed as a transfer function in z-transform:

let $d = T_d/T_s$ and $g = T_s/T_i$. Then

$$m(n) = K_p\left(e(n) + g\sum_{k=1}^{n} e(k) + d[e(n) - e(n-1)]\right) \quad (23)$$

$$D(z) = M(z)/E(z)$$

$$D(z) = K_p\left(1 + \frac{gz}{z-1} + d - dz^{-1}\right) \quad (24)$$

$$K_p gz/(z-1) = K_p gz/(1 - z^{-1})$$

$$x_1(i) = K_p(1+d)e(i)$$
$$x_2(i) = K_p ge(i) + x_2(i-1)$$
$$x_3(i) = -K_p de(i-1)$$
$$m(i) = x_1(i) + x_2(i) + x_3(i)$$

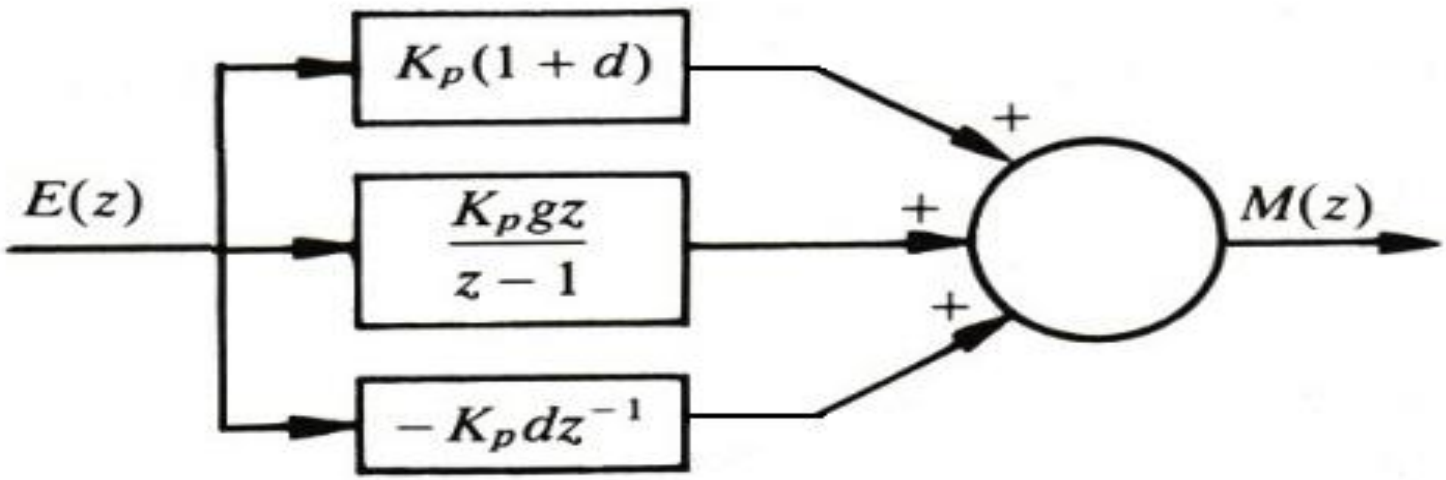
Substituting for d and g gives

$$x_1(i) = K_p(1 + T_d/T_s)e(i)$$
$$x_2(i) = K_p(T_s/T_i)e(i) + x_2(i-1) \quad (25)$$
$$x_3(i) = -K_p(T_d/T_s)e(i-1)$$

The algorithm from equation25 is

$$s(n) = K_1 e(n) + s(n-1)$$
$$m(n) = K_2 e(n) + K_3 e(n-1) + s(n)$$

$K_1 = K_p T_s / T_i$, $K_2 = K_p(1+T_d/T_s)$ and $K_3 = -K_p T_d / T_s$.



z-transform function form of PID controller

$$D(z) = K_p\left(\frac{(1+g+d)z^2 - (1+2d)z - d}{z(z-1)}\right) \qquad (26)$$

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1}}$$

where $a_0 = K_p(1+g+d)$
$a_1 = -K_p(1+2d)$
$a_2 = K_p d$
$b_1 = -1$

Direct implementation gives

$$m(i) = a_0 e(i) + a_1 e(i-1) + a_2 e(i-2) - b_1 m(i-1) \qquad (27)$$

$$m(i) = K_p\left[\left(1 + \frac{T_s}{T_i} + \frac{T_d}{T_s}\right)e(i) - \left(1 + 2\frac{T_d}{T_s}\right)e(i-1) + \frac{T_d}{T_s}e(i-2) + m(i-1)\right] \qquad (28)$$

Controller Realization:

1. Direct Structure:

1.1 Direct Noncanonical Structure:

The transfer function can be expressed as:

$$\frac{M(z)}{E(z)} = D(z) = \frac{\Sigma_{j=0}^{n} a_j z^{-j}}{1 + \Sigma_{j=1}^{n} b_j z^{-j}} \tag{29}$$

The transfer function in equation 29 is converted directly into the difference equation

$$m_i = \sum_{j=0}^{n} a_j e_{i-j} - \sum_{j=1}^{n} b_j m_{i-j} \tag{30}$$

EXAMPLE

Consider a system with the transfer function

$$\frac{M(z)}{E(z)} = D(z) = \frac{3 + 3.6z^{-1} + 0.6z^{-2}}{1 + 0.1z^{-1} - 0.2z^{-2}}$$

Then by direct method 1 the computer algorithm is simply

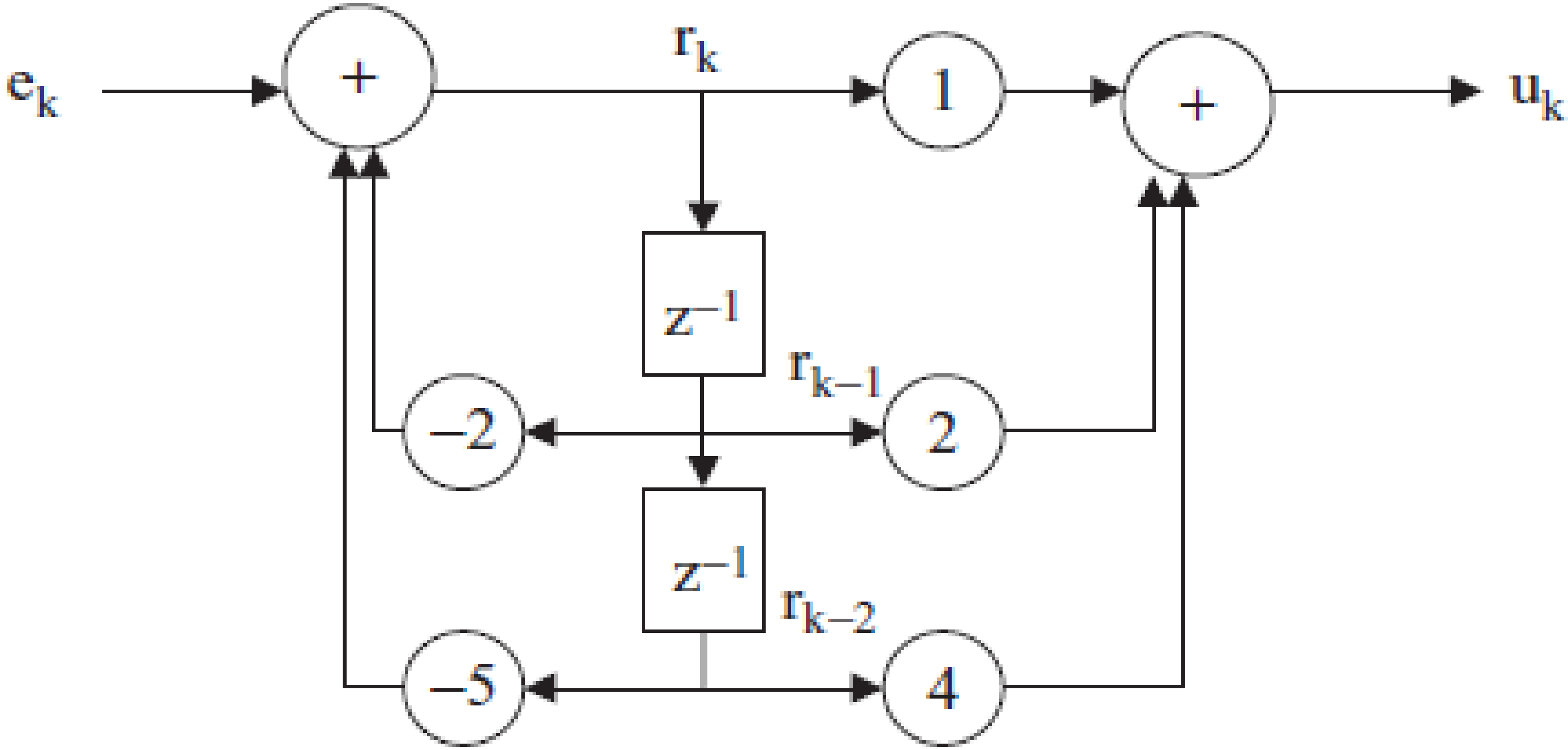
$$m_i = 3e_i + 3.6e_{i-1} + 0.6e_{i-2} - 0.1m_{i-1} + 0.2m_{i-2}$$

Example

The transfer function of a digital controller is found to be

$$D(z) = \frac{1 + 2z^{-1} + 4z^{-2}}{1 + 2z^{-1} + 5z^{-2}}.$$

Draw the block diagram of the direct canonical realization of this controller.



1.2 Direct Canonical Structure:

the difference equation is formulated by introducing an auxiliary variable $P(z)$ such that

$$\frac{M(z)}{P(z)} = \sum_{j=1}^{n} a_j z^{-j} \qquad (32)$$

and

$$\frac{P(z)}{E(z)} = \frac{1}{1 + \Sigma_{j=1}^{n} b_j z^{-j}} \qquad (33)$$

From equations 32 and 33 two equations are obtained:

$$m_i = \sum_{j=0}^{n} a_j p_{i-j} \qquad (34)$$

and

$$p_i = e_i - \sum_{j=0}^{n} b_j p_{i-j} \qquad (35)$$

Using the example above the following algorithm is obtained:

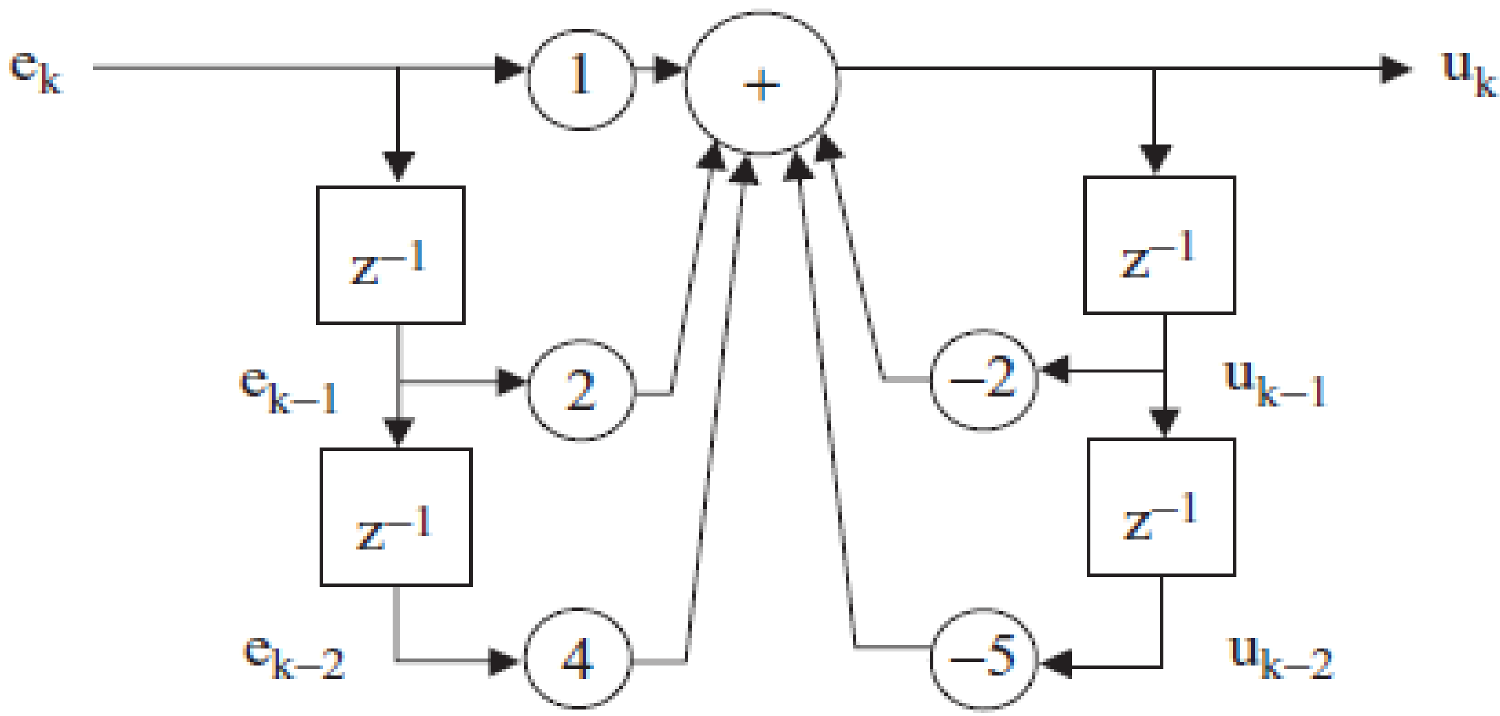
$$p_i = e_i - 0.1 p_{i-1} + 0.2 p_{i-2}$$
$$m_i = 3 p_i + 3.6 p_{i-1} + 0.6 p_{i-2}$$

Example

The transfer function of a digital controller is found to be

$$D(z) = \frac{1 + 2z^{-1} + 4z^{-2}}{1 + 2z^{-1} + 5z^{-2}}.$$

Draw the block diagram of the direct noncanonical realization of this controller.



2. Cascade Realization:

- The transfer function is expressed as the product of simple block elements of 1st and 2nd order, then each element can be converted to a difference equation using direct structure.

$$\frac{M(z)}{E(z)} = D(z) = \frac{3(1+z^{-1})(1+0.2z^{-1})}{(1+0.5z^{-1})(1-0.4z^{-1})} \quad (36)$$

Hence

$$D_1 = 3$$
$$D_2 = (1+z^{-1}) \qquad D_4 = 1/(1+0.5z^{-1})$$
$$D_3 = (1+0.2z^{-1}) \qquad D_5 = 1/(1-0.4z^{-1})$$

$$x_1(i) = 3e(i)$$
$$x_2(i) = x_1(i) + x_1(i-1)$$
$$x_3(i) = x_2(i) + 0.2x_2(i-1)$$
$$x_4(i) = x_3(i) - 0.5x_4(i-1)$$
$$x_5(i) = x_4(i) + 0.4x_5(i-1)$$

Example

The transfer function of a digital controller is given by

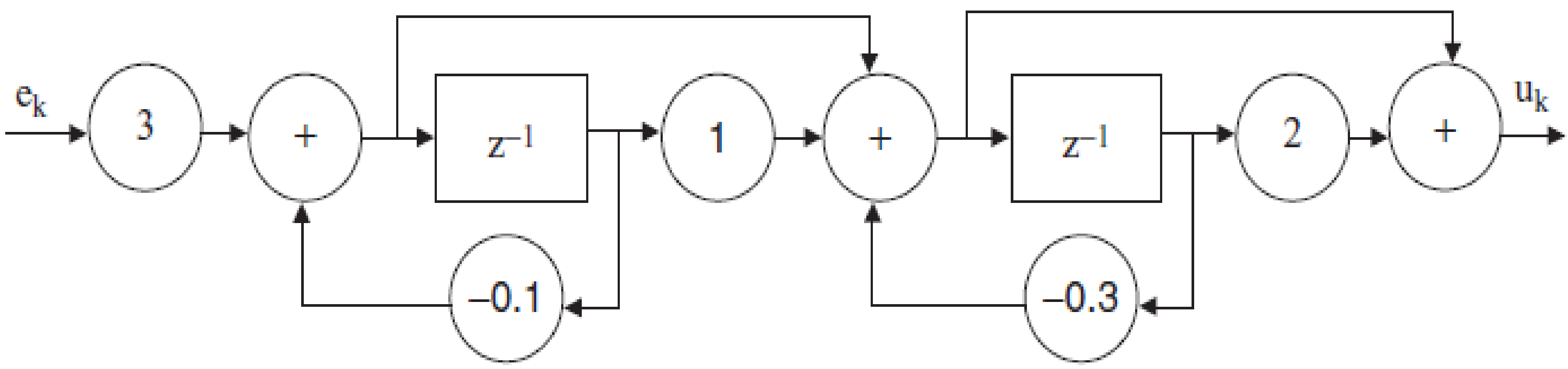
$$D(z) = \frac{3(z+1)(z+2)}{z^2 + 0.4z + 0.03}.$$

Use two-first order cascaded transfer functions to implement this controller.

Solution

The transfer function can be factorized as

$$D(z) = \frac{3(z+1)(z+2)}{(z+0.1)(z+0.3)} = \frac{3(1+z^{-1})(1+2z^{-1})}{(1+0.1z^{-1})(1+0.3z^{-1})}.$$

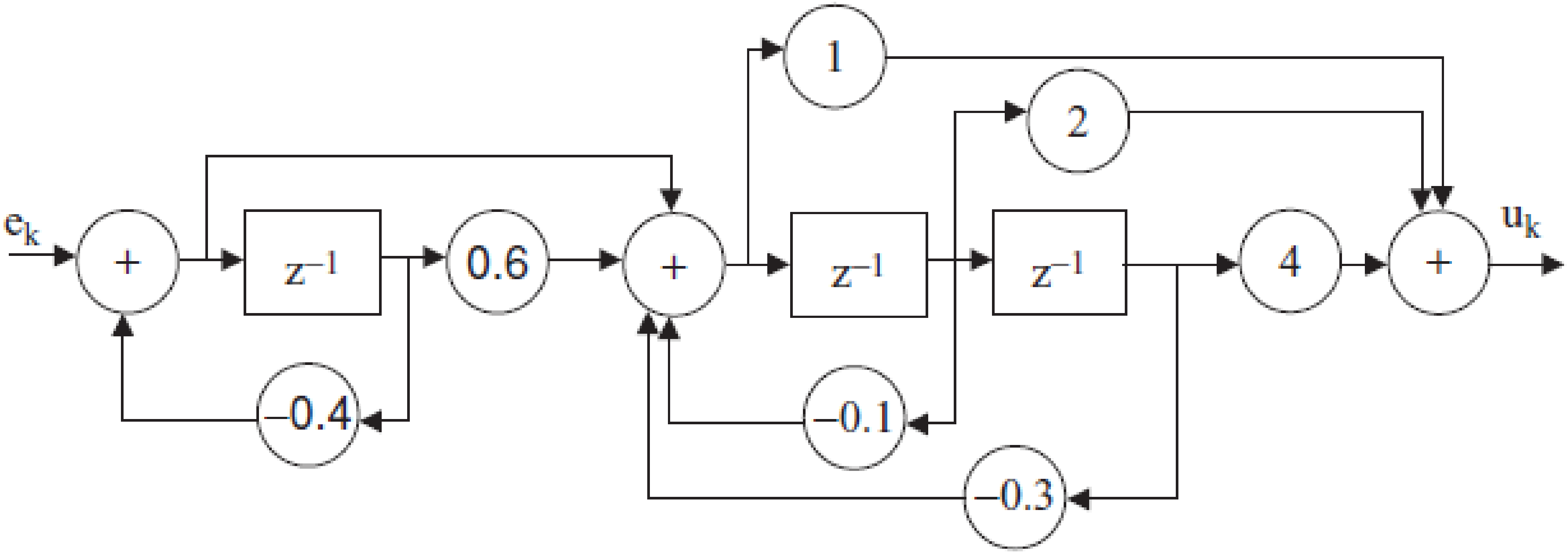


Example

The transfer function of a digital controller is given by

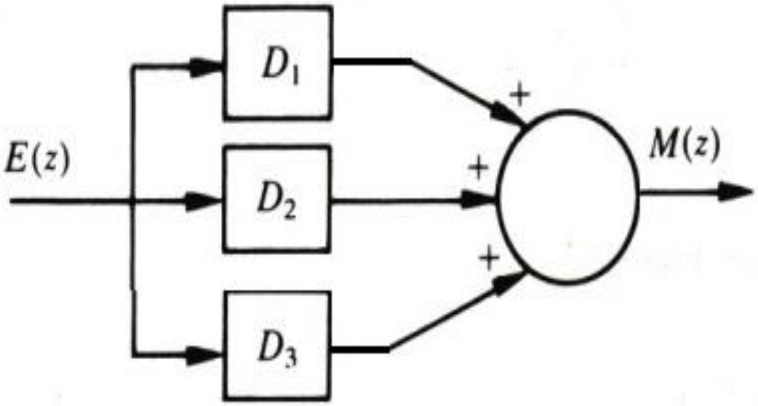
$$D(z) = \frac{(1 + 0.6z^{-1})(1 + 2z^{-1} + 4z^{-2})}{(1 + 0.4z^{-1})(1 + 0.1z^{-1} + 0.3z^{-2})}.$$

Use a first-order and a second-order cascaded transfer function to implement this controller.



3. Parallel Realization:

- The transfer function is expressed in fractional form or is expanded into partial fractions, then it can be expressed as given bellow.
- Each element is expressed in difference equation form using direct structure.



Consider a system with the transfer function

$$\frac{M(z)}{E(z)} = D(z) = \frac{3 + 3.6z^{-1} + 0.6z^{-2}}{1 + 0.1z^{-1} - 0.2z^{-2}}$$

The partial fraction expansion is

$$\frac{M(z)}{E(z)} = D(z) = -3 - \frac{1}{1 + 0.5z^{-1}} + \frac{7}{1 - 0.4z^{-1}}$$

Hence $D_1 = -3$, $D_2 = -1/(1 + 0.5z^{-1})$, $D_3 = 7/(1 - 0.4z^{-1})$

and the algorithm is

$$x_1(i) = -3e(i)$$
$$x_2(i) = -e(i) - 0.5x_2(i-1)$$
$$x_3(i) = 7e(i) + 0.4x_3(i-1)$$
$$m(i) = x_1(i) + x_2(i) + x_3(i)$$

Example

The transfer function of a digital controller is given by

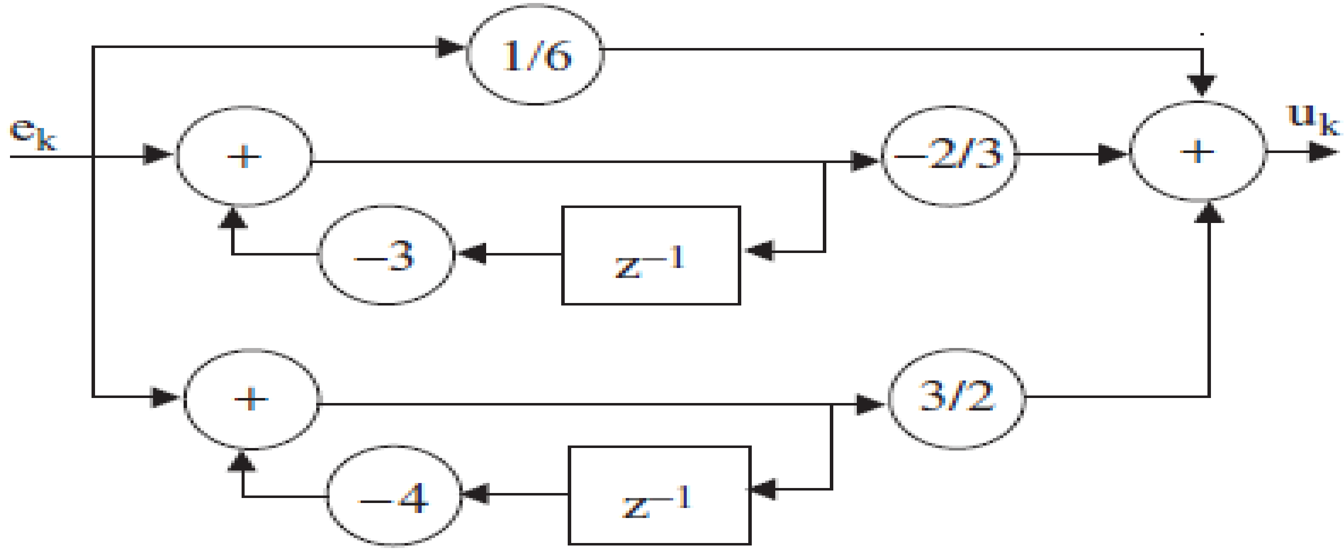
$$D(z) = \frac{(1+z^{-1})(1+2z^{-1})}{(1+3z^{-1})(1+4z^{-1})}.$$

Realize this transfer function using first-order parallel transfer functions.

Solution

The controller transfer function can be factorized as follows:

$$D(z) = \frac{(1+z^{-1})(1+2z^{-1})}{(1+3z^{-1})(1+4z^{-1})} = \frac{-2}{3(1+3z^{-1})} + \frac{3}{2(1+4z^{-1})} + \frac{1}{6}.$$



PID Controller Implementation:

Using Parallel Realization:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} + u_0,$$

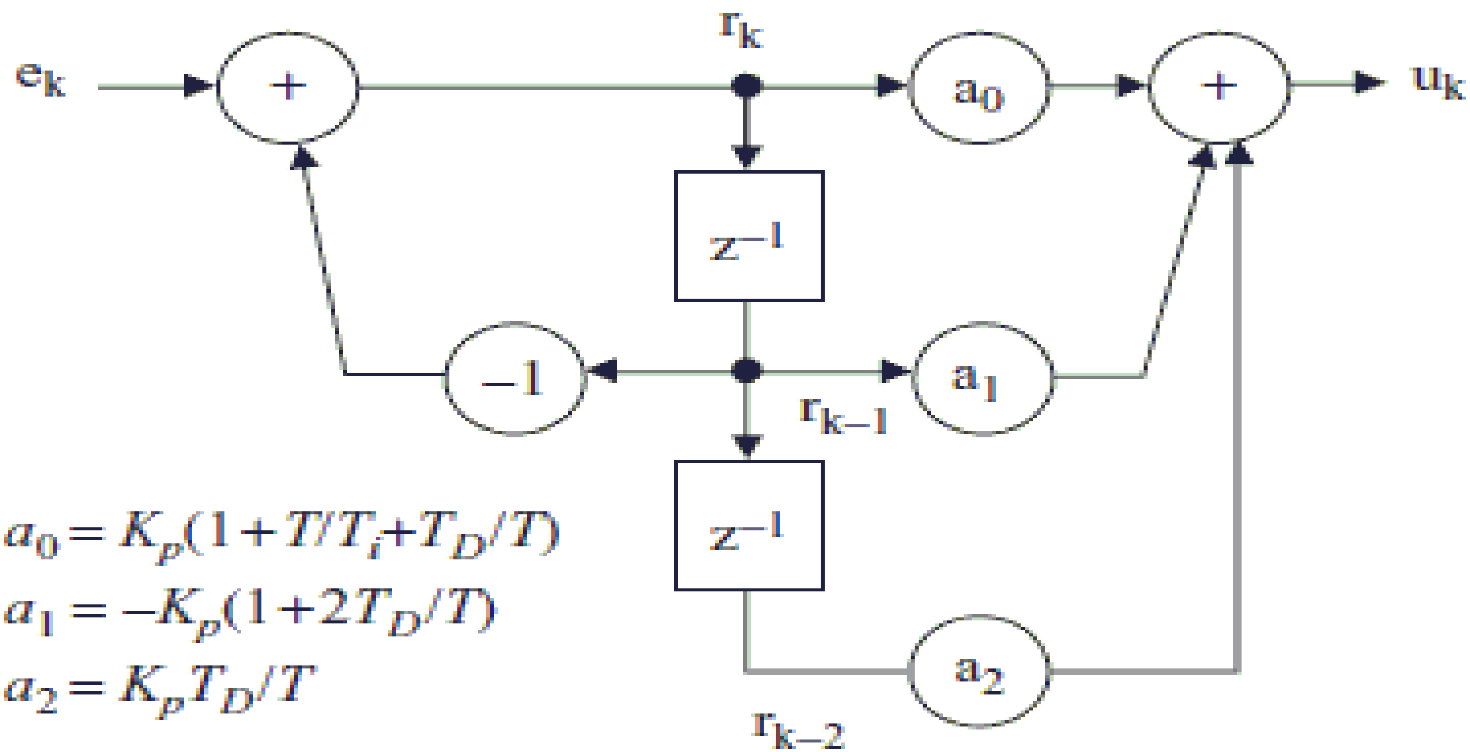
$$\frac{U(s)}{E(s)} = K_p + \frac{K_p}{T_i s} + K_p T_d s.$$

where $K_i = \frac{K_p}{T_i}$ and $K_d = K_p T_d$.

$$u(kT) = K_p \left[e(kT) + T_d \frac{e(kT) - e(kT - T)}{T} + \frac{T}{T_i} \sum_{k=1}^{n} e(kT) \right] + u_0.$$

$$\frac{U(z)}{E(z)} = K_p \left[1 + \frac{T}{T_i(1 - z^{-1})} + T_d \frac{(1 - z^{-1})}{T} \right].$$

PID Controller Implementation Using Parallel Realization:



$$a_0 = K_p(1 + T/T_i + T_D/T)$$
$$a_1 = -K_p(1 + 2T_D/T)$$
$$a_2 = K_p T_D/T$$

Homework: Draw the diagram of the PID implementation using direct canonical realization?

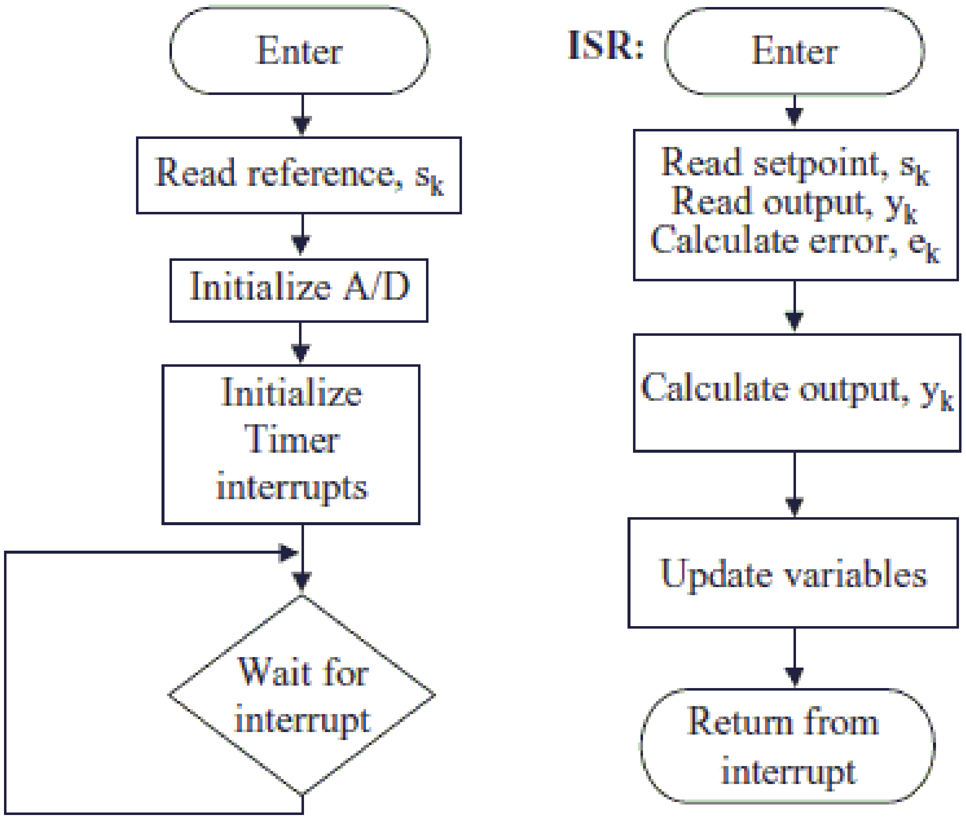
Real-Time Algorithms:

Examples:

1. Realization using single processor.
2. Realization using more than one processors.
3. Online Tuning of PID Controller.
4. Intelligent Tuning of PID Controller.

MICROCONTROLLER IMPLEMENTATIONS:

- The final stage of a system design is the implementation of the required algorithm (set of difference equations) on a single MC or more.
- Microcontrollers have traditionally been programmed using the assembly language of the MC. Assembly language has several important disadvantages and is currently less popular than it used to be.
- For accurate implementation, use the timer/counter interrupt to generate the required sampling interval. In this case, the software consists of two parts: the main program and the interrupt service routine.

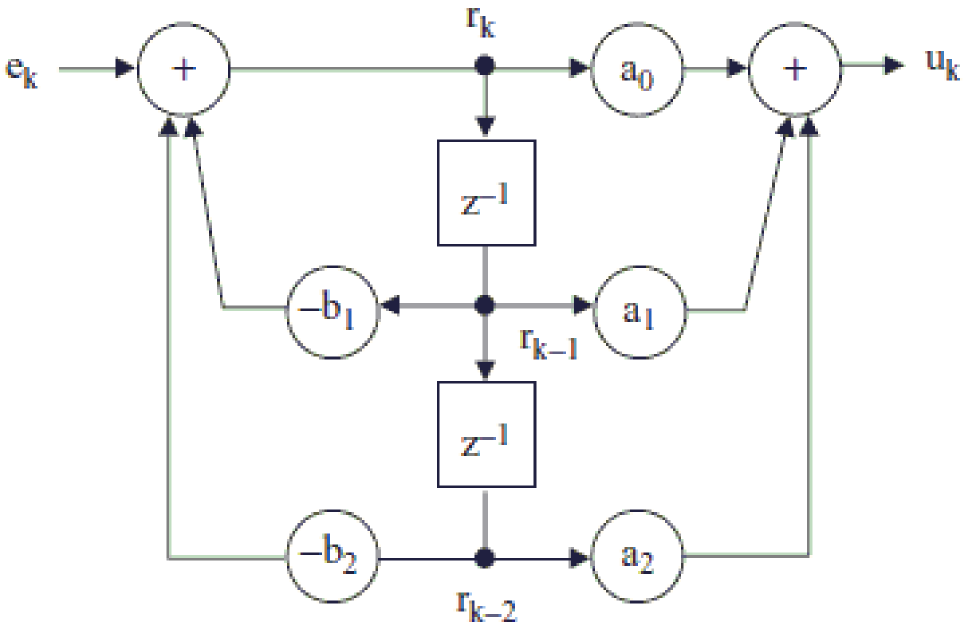


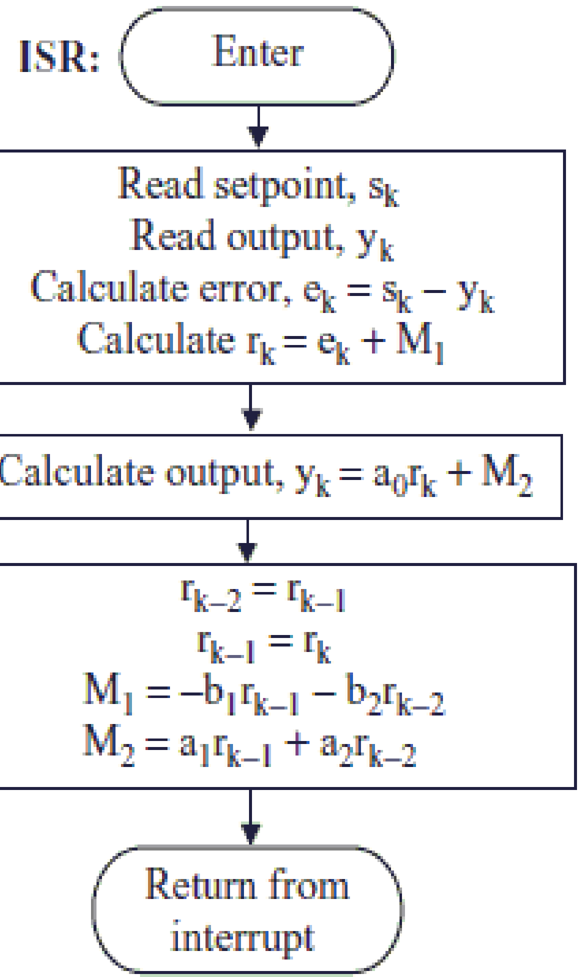
Implementing Second-Order Modules

A 2nd-order module can be realized using adders, multipliers and delay elements. The difference equations describing such a module are;

$$M_1 = -b_1 r_{k-1} - b_2 r_{k-2}$$
$$M_2 = a_1 r_{k-1} + a_2 r_{k-2}$$
$$r_k = e_k + M_1,$$

$$u_k = a_0 r_k + M_2.$$





Example:

Assume that the digital controller to be implemented is in the form of a second-order module. The controller parameters are assumed to be

a0 = 1, a1 = 0.8, a2 = 1.2, b1 = 1.85, b2 = 0.92,

i.e. the required controller transfer function is

$$D(z) = \frac{1 + 0.8z^{-1} + 1.2z^{-2}}{1 + 1.85z^{-1} + 0.92z^{-2}}$$

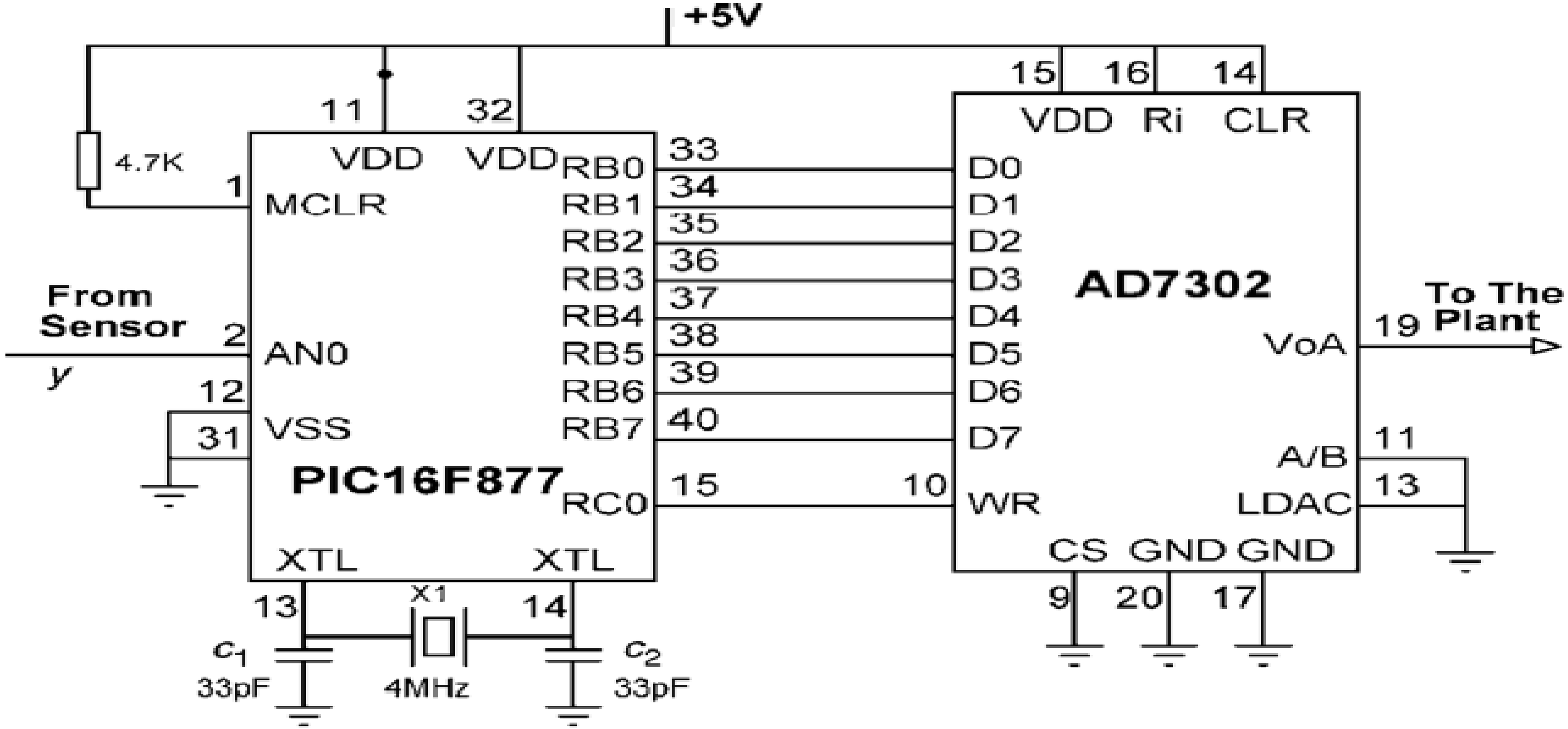
Also assume that the required sampling interval is *T = 10 ms?*

Solution:

The controller hardware is based on a PIC16F877 microcontroller. The microcontroller is operated from a 4MHz crystal, connected to OSC1 and OSC2 inputs. With a 4MHz crystal, the basic clock rate of the microcontroller is 1μs.

The program consists of the *main program (its functions: Initialize AD, Initialize Timer, Read AD Input) and ISR.*

Main program: *The coefficients of the controller are defined at the beginning of the main* program. Also, the A/D converter and the timer initialization functions are called here. The main program then enables global interrupts and enters an endless loop waiting for timer interrupts to occur.



ISR: *This is the interrupt service routine. The program jumps to this function every 10 ms.* The function reads a sample, and calculates the error term *ek . The output value yk is then* calculated and sent to the D/A converter. In the final part of the ISR, the variables are updated for the next sample, and the timer interrupt is re-enabled.

Initialize Timer: *This function initializes the timer TMR0 so that timer interrupts can be* generated at 10 ms intervals. The timing interval is given by:

Timing interval = internal clock period*prescaler*(256 − TMR0 value)

Then, the TMR0 value = 256 − 10ms*/(1us*64)= 100*

Choice of Sampling Interval (T):

•Choosing a large sampling time has several disadvantages;

- destabilizing effects on the system.
- information loss occurs.
- errors increase when a continuous system is sampled.

• Decreasing the sampling interval towards zero will make a discrete system converge towards an equivalent continuous system, and in practice this is not the case due to the hardware and software delay.

•Various empirical rules have been suggested by many researchers for the selection of the sampling interval. These rules are based on practical experience and simulation results. Among them are the following;

➢ If the plant has the dominant time constant *Tp, then the sampling interval T for the closed loop* system should be selected such that $T < Tp/10$.

➢If the closed-loop system is required to have a settling time *Tss or a natural frequency of* ωn then choose the sampling interval *T such that* $T<Tss/10$ *and* $\omega s>10\omega n$, *where* ωs *is the* sampling frequency, i.e. $\omega s = 2\pi/T$.

➢If the process has an open-loop model $G(s) = e^{-sT_1}/(1 + sT_2)$ then the sampling interval should be selected such that $T < T1/4$.

Homework (2):

1. The transfer function of a digital controller is given by

$$D(z) = \frac{1 + 2z^{-1} + 3z^{-2}}{1 + 4z^{-1} + 5z^{-2}}$$

a). Draw the block diagram of the direct canonical realization.

b). Draw the block diagram of the direct noncanocical realization.

c). Compare the realizations obtained above.

2. The transfer function of a digital controller is found to be

$$D(z) = \frac{1 + 2z^{-1} + 5z^{-2}}{1 + 3z^{-1} + 7z^{-2}}$$

a). Draw the block diagram of the direct canonical realization.

b). Draw the block diagram of the direct noncanocical realization.

c). Compare the realizations obtained above.

3. The transfer function of a digital controller is given by

$$D(z) = \frac{2(z+2)(z+3)}{z^2 + 0.4z + 0.03}$$

Use two 1st-order cascaded transfer functions to implement this controller.

4. The transfer function of a digital controller is given by

$$D(z) = \frac{(1 + 0.2z^{-1})(1 + 2z^{-1} + 4z^{-2})}{(1 + 0.3z^{-1})(1 + 0.2z^{-1} + 0.4z^{-2})}$$

Use a 1st-order and a 2nd -order cascaded transfer function to implement this controller?

5. The transfer function of a digital controller is given by

$$D(z) = \frac{(1 + 2z^{-1})(1 + 3z^{-1})}{(1 + z^{-1})(1 + 5z^{-1})}$$

Realize this transfer function using 1st-order parallel transfer functions?

6. Draw the block diagram of the PID implementation using;

a). Parallel realization? b). Direct canonical realization?

7. Describe how a given realization can be implemented on a microcontroller?

8. Draw a flow diagram to show how the PID algorithm can be implemented on a microcontroller?

9. The transfer function of a digital controller is given by

$$D(z) = \frac{1 + 2z^{-1} + 5z^{-2}}{1 + 3z^{-1} + 4z^{-2}}$$

a). Draw a flow diagram to show how this controller can be implemented on a MC?

b). Write a program to implement this algorithm on a microcontroller.

10. Draw a flow diagram to show and explain how a 2nd -order transfer function can be implemented on a microcontroller?
11. Explain howa 2nd -order direct canonical functions can be cascaded to obtain higher order transfer functions?
12. Explain how the sampling time can be selected in a 1st-order system and a 2nd-order system?.
13. Describe the problems that may occur when very large or very small sampling times are selected?
14. Explain how the system stability is affected when the sampling time is increased?

References:

1. Houpis, C.H. and Lamont, B. Digital Control Systems: Theory, Hardware, Software, 2nd edn., McGraw-Hill, New York, 1992.
2. Dogan Ibrahim, "Microcontroller Based Applied Digital Control", J.Wiley, England, 2006. ISBN 0-470-86335-8.
3. Katz, P. Digital Control Using Microprocessors. Prentice Hall, Englewood Cliffs, NJ, 1981.
4. Nagle, H.T. Jr. and Nelson, V.P. Digital filter implementation on 16-bit microcomputers. *IEEE MICRO, 1, 1, February 1981, pp. 23–41.*
5. Phillips, C.L. and Nagle, H.T. Jr. Digital Control Systems: Analysis and Design, 2nd edn., Prentice Hall, Englewood Cliffs, NJ, 1990.