Real-Time Systems

(0630581)

Lecture: 6

Real-Time Implementation of Digital Algorithms

Prof. Kasim M. Al-Aubidy
Computer Engineering Department
Philadelphia University
Summer Semester, 2011

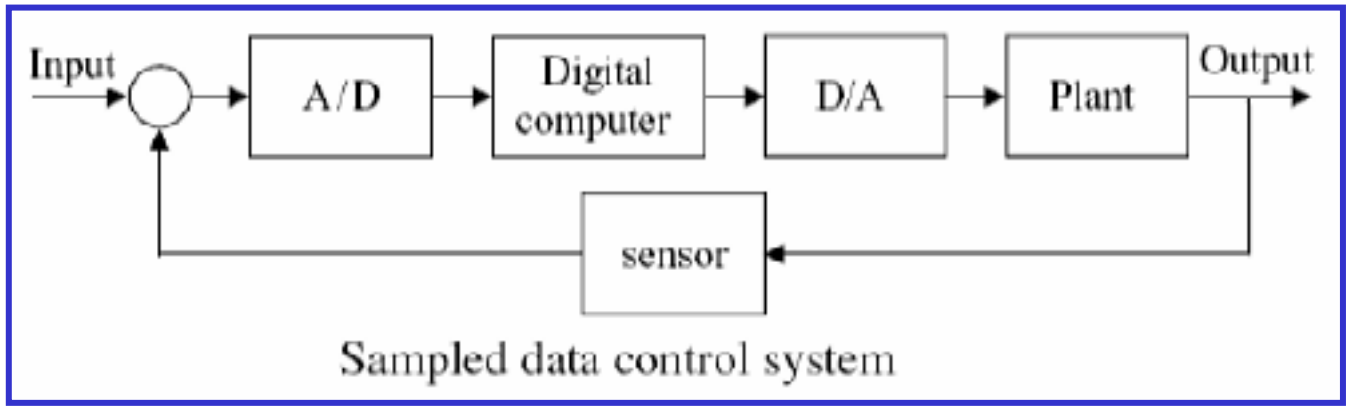
Course Objectives:

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

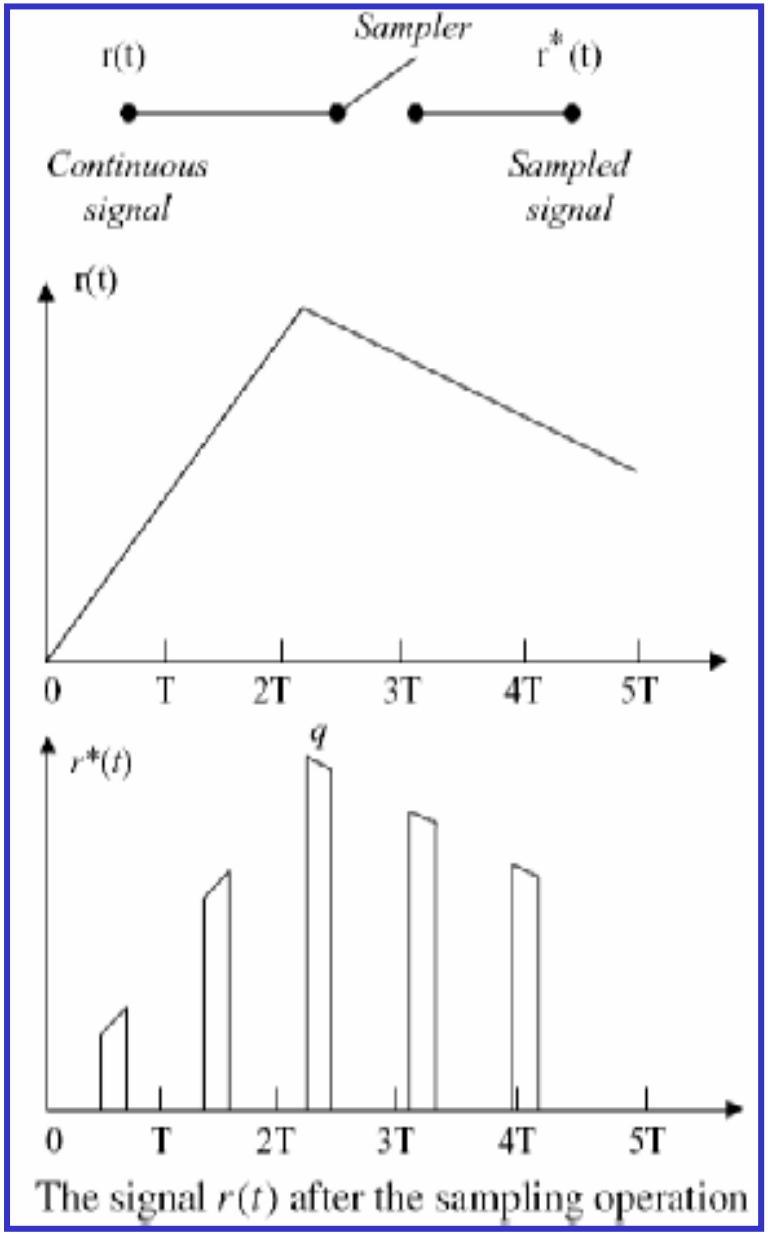
The Sampling Process:

- 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.

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

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

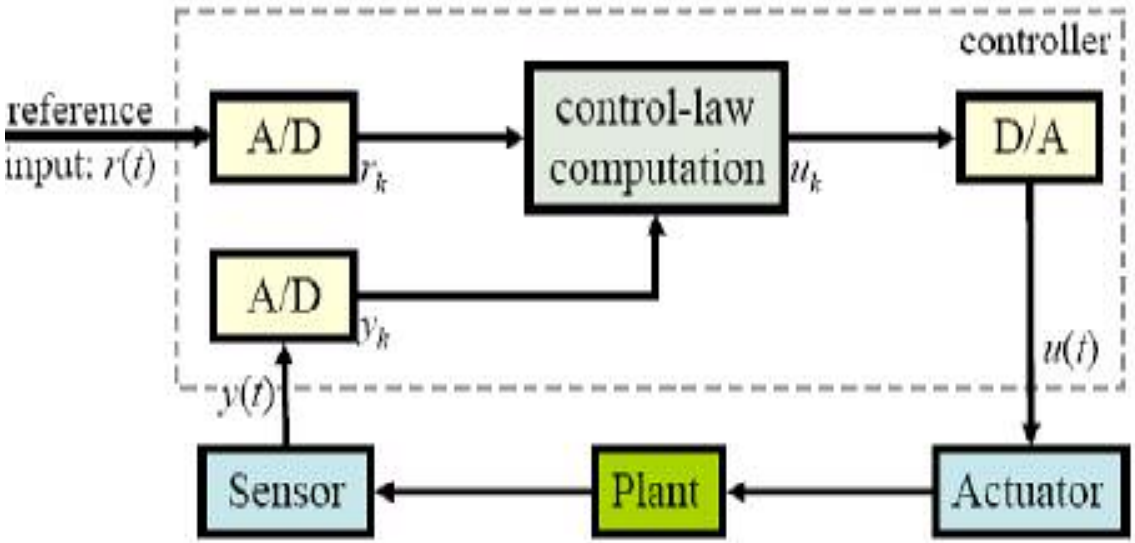
$$r^*(t) = r(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$$



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

PID Control Algorithm:

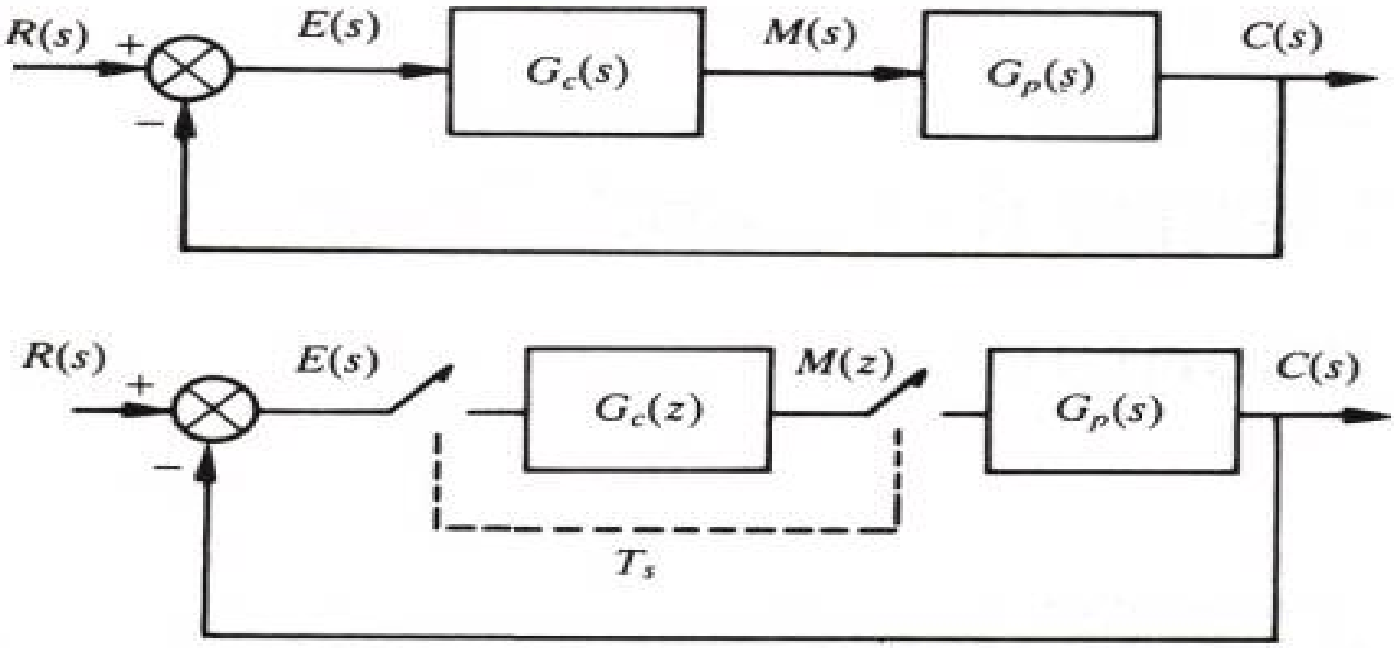
- The Proportional Integral Derivative (PID) control is a simple, three-term controller used in industry.
- The differential equation for a PID controller is :



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

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

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



- 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:

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

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.

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

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

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

$$\begin{aligned}K_1 &= K_p(1 + T_s/T_i + T_d/T_s)\\ K_2 &= -K_p\,(1 + 2T_d/T_s)\\ K_3 &= K_p\,T_d/T_s\end{aligned}$$

$$\Delta m(n) = K_1 e(n) + K_2 e(n-1) + K_3 e(n-2) \qquad (10)$$

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.

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

$$s(n) = s(n-1) + e(n)$$
$$m(n) = K_p e(n) + K_i s(n) + K_d[e(n) - e(n-1)] \quad (12)$$

$$s(n) = s(n-1) + e(n)(T_s/T_i)$$
$$m(n) = K_p e(n) + K_p s(n) + K_d[e(n) - e(n-1)] \quad (13)$$

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

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

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

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

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

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

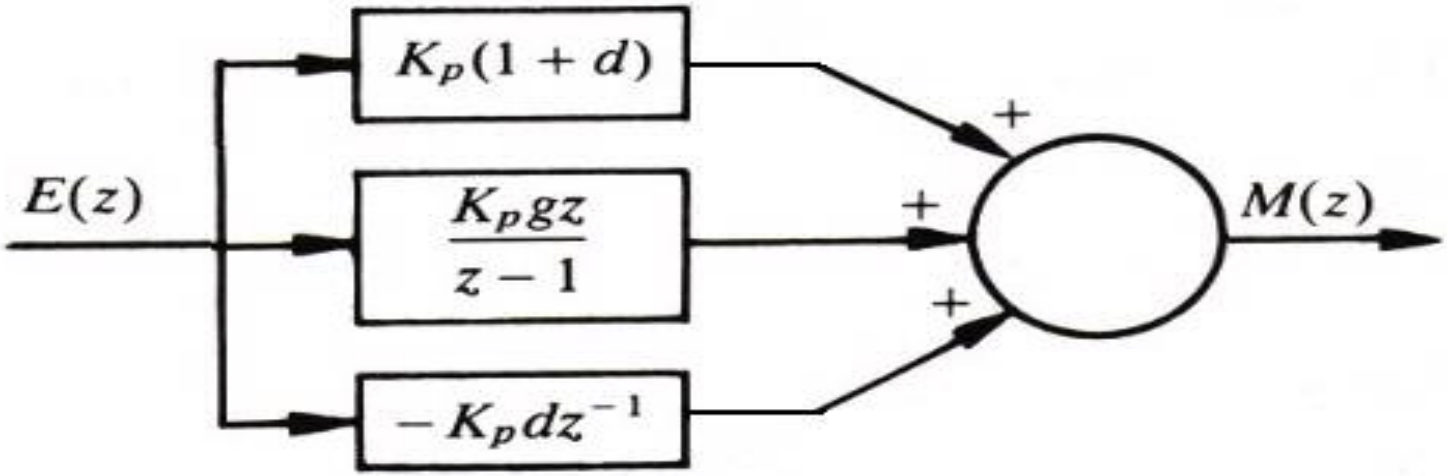
Substituting for d and g gives

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

The algorithm from equation 25 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:

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

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

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

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

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

1.2 Direct Canonical Structure:

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

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

and

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

From equations 32 and 33 two equations are obtained:

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

and

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

Using the example above the following algorithm is obtained:

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

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.

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

Hence $D_1 = 3$

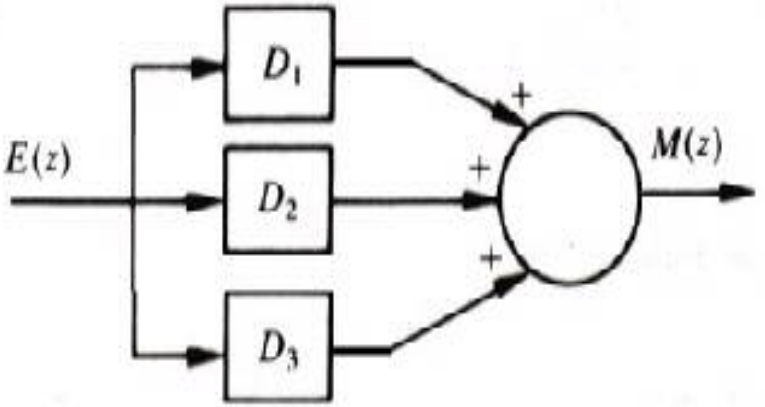
$D_2 = (1+z^{-1})$ $\qquad D_4 = 1/(1+0.5z^{-1})$

$D_3 = (1+0.2z^{-1})$ $\qquad D_5 = 1/(1-0.4z^{-1})$

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

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

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