Dr. Norbert Cheung's Series
in
Electrical Engineering

Level 4 Topic no: 13

Digital Filters

Contents

Reference:

"Modern Digital Control Systems, 2nd edition" Raymond G. Jacquot, Longman.

Email: norbert.cheung@polyu.edu.hk **Web Site:** www.ncheung.com

1. Introduction

Any z-domain transfer function could then be thought of as a digital filter.

In the design of digital compensators, two paths are generally taken. The first is to ignore any zero-order holds and samplers in the control loop and do a preliminary design in the s-domain as if one were building a continuous-time control system. This design must then be converted to a discrete-time design by some approximate technique to yield a discrete-time compensator. Once these steps have been accomplished, a z-domain analysis can be employed to verify whether the original design goals have been met. The second method is to design the compensator directly in the z-domain using z-domain frequency response methods or the z-domain root-locus method; this technique was the topic of Chapters 3 and 4.

The first method has an advantage since engineers are more accustomed to thinking clearly in the s-plane than in the z-plane. It has a disadvantage since in the process of conversion to a discrete-time design, by whatever method, the z-plane poles are distorted from where they are needed, and hence a trial-and-error design procedure may be in order.

The second method has the advantage that the poles and zeros of the compensator are located directly, and the designer can pick these locations *a priori*. A disadvantage is that it is difficult for the designer to visualize exactly where he might want the z-domain poles and zeros to improve system performance.

The author's experience indicates that the first method is usually best for all but experienced z-domain designers. We shall assume for the remainder of this chapter that the design has been done in the s-domain and then is to be implemented in discrete form on the digital computer. In this chapter we discuss conversion of an s-domain transfer function $H(s)$ into a discrete-time transfer function $H(z)$ or an equivalent difference equation.

Filter design in s-domain *H(s)* >> convert into discrete form *H(z)*

2. Using prototypes to design filters

One good prototype filter is the Butterworth filter, which has minimum ripple in the passband and stopband, although it does not make the transition from passband to stopband as crisply as other possible filters. The best way to start the process is to begin with a unity-gain, unity-bandwidth lowpass Butterworth filter of a fixed order. Typical low-order unity bandwidth (1 rad/sec) Butterworth transfer functions are given in Table 5.2.

Table 5.2. Butterworth Unity-Gain, Unity-Cutoff Frequency Transfer

Order	$H_B(s)$
2	$\frac{1}{s^2 + 1.414s + 1}$
3	$\frac{1}{s^3 + 2s^2 + 2s + 1}$
4	$\frac{1}{s^4 + 2.6133s^3 + 3.414s^2 + 2.6133s + 1}$

For a lowpass filter with arbitrary cutoff frequency ω_0, we simply employ the lowpass transformation whereby s is replaced by s/ω_0, which essentially frequency scales the lowpass filter:

$$s \rightarrow \frac{s}{\omega_0} \quad (5.9.1)$$

For a bandpass filter with the passband centered at ω_0 and a bandwidth of BW rad/s we employ the bandpass transformation, which is

$$s \rightarrow \frac{1}{\mathrm{BW}}\left(\frac{s^2 + \omega_0^2}{s}\right) \quad (5.9.2)$$

If a highpass filter is desired, we use the highpass transformation, which is

$$s \rightarrow \frac{\omega_0}{s} \quad (5.9.3)$$

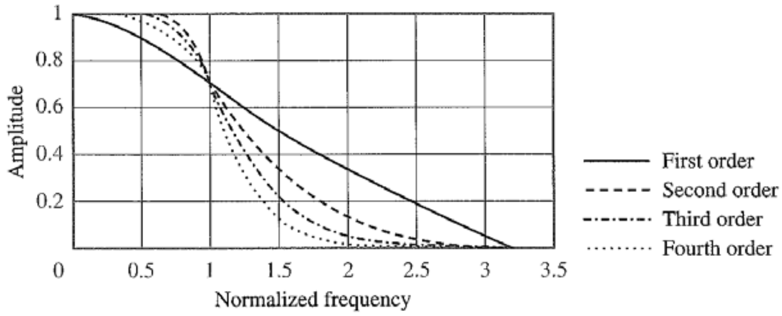

Figure 15.20 Butterworth low-pass filter frequency response

Table 15.3 **Butterworth polynomials in quadratic form**

Order n	Quadratic factors
1	$s+1$
2	$s^2+\sqrt{2}s+1$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.7654s+1)(s^2+1.8478s+1)$
5	$(s+1)(s^2+0.6180s+1)(s^2+1.6180s+1)$

Example 5.1. Consider a second-order bandpass filter with a bandpass frequency of 10 rad/s and a Q of 5 with the continuous-time transfer function

$$H(s) = \frac{Y(s)}{U(s)} = \frac{2s}{s^2 + 2s + 100} \qquad \text{(a)}$$

If we cross-multiply the transfer function, we get

$$(s^2 + 2s + 10)Y(s) = 2sU(s)$$

Inversion of the indicated transform yields a differential equation description

$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 100y = 2\frac{du}{dt} \qquad \text{(b)}$$

An alternative description is that of the impulse response function, which is the inverse Laplace transform of the transfer function of (a). This inversion can be accomplished by adding a constant term to the numerator of (a) and then subtracting a term of the same size upon completing the square in the denominator

$$H(s) = \frac{2(s+1)}{(s+1)^2 + 99} - \frac{2}{\sqrt{99}}\frac{\sqrt{99}}{(s+1)^2 + 99}$$

Using the appropriate table entries in Appendix A, the impulse response function is

$$h(t) = \begin{cases} 2e^{-t}\cos\sqrt{99}t - \dfrac{2}{\sqrt{99}}e^{-t}\sin\sqrt{99}\,t & t \geq 0 \\ 0 & t < 0 \end{cases} \qquad \text{(c)}$$

The frequency response of this filter is

$$H(j\omega) = H(s)|_{s=j\omega} = \frac{2j\omega}{-\omega^2 + 2j\omega + 100} \qquad \text{(d)}$$

3. Conventional design techniques

General filter function in s-domain

If the filter described by relation (5.2.1) has numerator and denominator polynomials of equal order, long division must be carried out to give a constant plus a remainder proper polynomial ratio. The result is of the form

$$H(s) = b_n + \frac{c_{n-1}s^{n-1} + \cdots + c_1 s + c_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} = \frac{Y(s)}{U(s)} \qquad (5.2.4)$$

If we now define a new response variable $W(s)$ by the transfer function relation

$$\frac{W(s)}{U(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \qquad (5.2.5)$$

and thus the output relation is

$$Y(s) = b_n U(s) + (c_{n-1}s^{n-1} + \cdots + c_1 s + c_0)W(s) \qquad (5.2.6)$$

Relation (5.2.5) could be realized as a cascade of integrators with feedback to create the denominator dynamics as illustrated in the lower half of Fig. 5.1. Relation (5.2.6) can be thought of as a feedforward of the input $U(s)$ and the outputs of the integrators as illustrated in the upper portion of Fig. 5.1. One common way to convert the continuous-time filtering of Fig. 5.1 to a digital filter would be to change the signals in the block diagram to z-domain signals and to replace the integrator $(1/s)$ blocks by approximate discrete-time integrators. In the following section we explore several approximate integration algorithms.

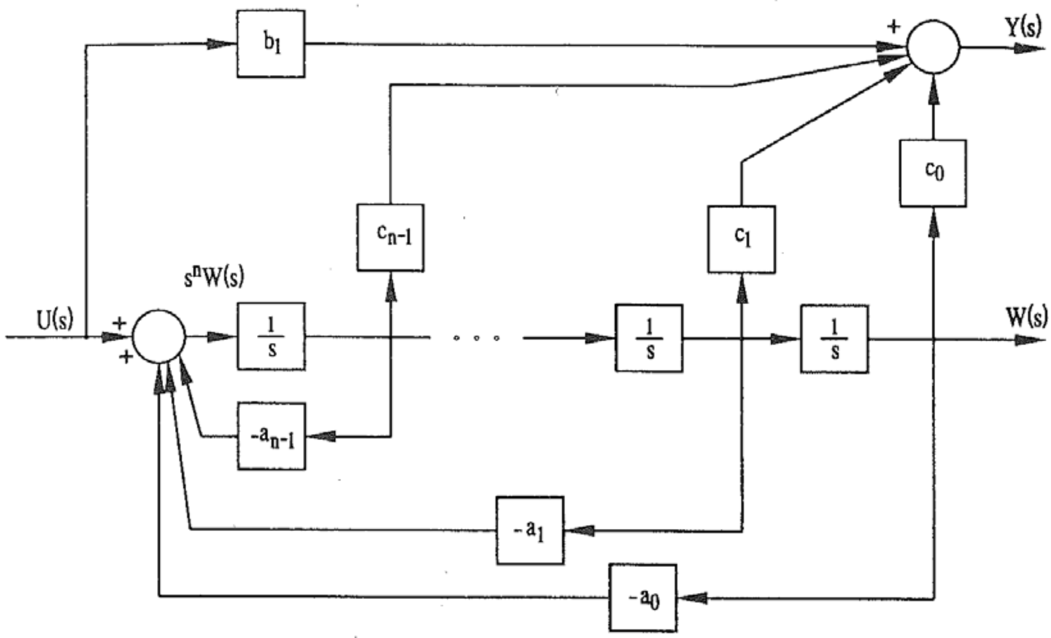

Figure 5.1. Phase-variable filter realization.

Example 5.2. The filter examined in Example 5.1 had a transfer function

$$H(s) = \frac{2s}{s^2 + 2s + 100} = \frac{Y(s)}{U(s)}$$

From relation (5.2.5) the forward dynamics are described by

$$\frac{W(s)}{U(s)} = \frac{1}{s^2 + 2s + 100}$$

and the output equation from (5.2.6) is $Y(s) = 2sW(s)$. These dynamics are represented in Fig. 5.2.

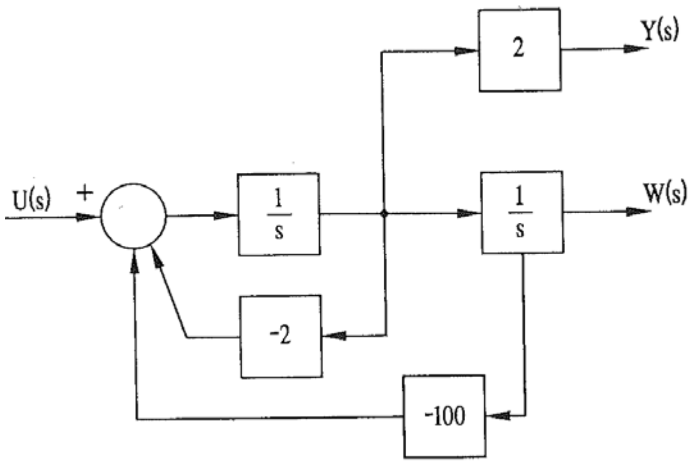

Figure 5.2. Bandpass filter realization.

4. Approximate numerical integration techniques

In this section we explore several techniques for numerical integration with the purpose in mind of approximation of filter transfer functions. The function we wish to approximate is represented in Fig. 5.3. The input–output relation is

$$y(t) = \int_0^t x(\tau)\, d\tau \qquad (5.3.1)$$

which can be thought of the area under the $x(\tau)$ curve between $\tau = 0$ and $\tau = t$.

Forward Rectangular Integration

In Fig. 5.4 we note that the integral at time kT will be denoted as $y(kT)$ and that at time $(k - 1)T$ will be denoted as $y((k - 1)T)$.

The area accumulated on the interval $((k - 1)T, kT)$ is then $x((k - 1)T) \cdot T$. Then the area at time kT can be thought of as the area at $(k - 1)T$ plus the rectangular area shown in Fig. 5.4. The integration algorithm is then

$$y(kT) = y((k - 1)T) + x((k - 1)T) \cdot T \qquad (5.3.2)$$

This difference equation algorithm for numerical integration can be transformed to give a z-domain transfer function

$$H(z) = \frac{Y(z)}{X(z)} = \frac{T}{z - 1} \qquad (5.3.3)$$

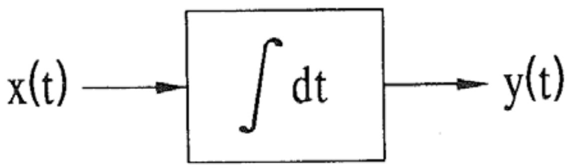

Figure 5.3. Continuous time-domain integrator.

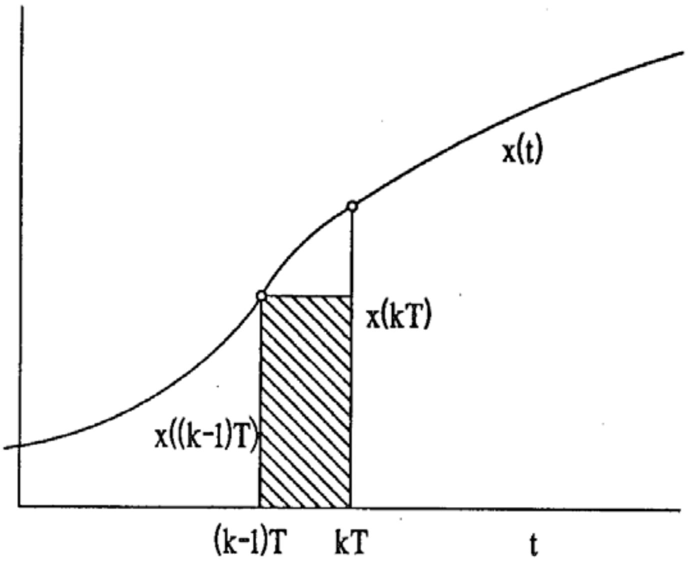

Figure 5.4. Forward rectangular integration.

Backward Rectangular Integration

Consider now an alternative definition for the integral approximation illustrated in Fig. 5.5. With this definition the integration algorithm becomes

$$y(kT) = y((k - 1)T) + x(kT) \cdot T \tag{5.3.4}$$

Upon z-transformation the transfer function for this discrete-time integra-

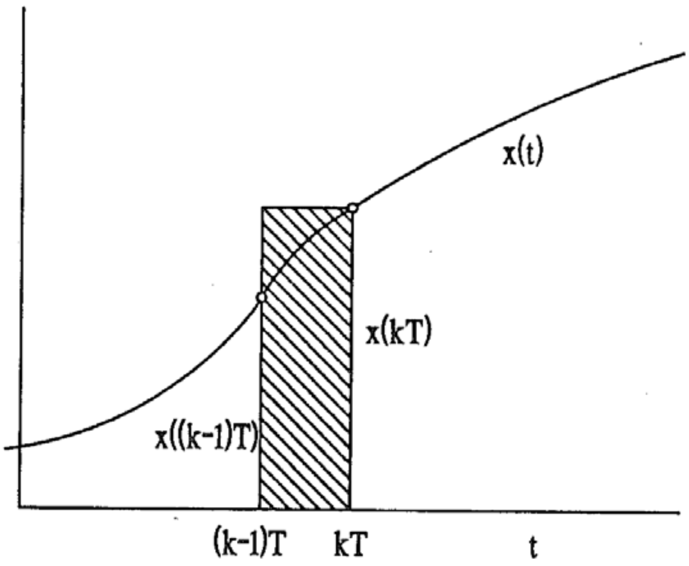

Figure 5.5. Backward rectangular integration.

tor is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{Tz}{z - 1} \qquad (5.3.5)$$

which differs in the numerator from that given by forward rectangular integration.

Trapezoidal Integration

Perhaps a better approximation to the integral can be obtained by using both samples of the $x(t)$ function in computation of the additional accumulated area by using the trapezoidal area illustrated in Fig. 5.6.

For this approximation the additional accumulated area is the area of the trapezoid or the algorithm becomes

$$y(kT) = y((k - 1)T) + \frac{T}{2}[x((k - 1)T) + x(kT)] \qquad (5.3.6)$$

and the associated z-domain transfer function is

$$H(z) = \frac{T}{2}\left(\frac{z + 1}{z - 1}\right) \qquad (5.3.7)$$

This is commonly referred to as the bilinear transformation, and the method of filter synthesis using this method is sometimes called Tustin's method

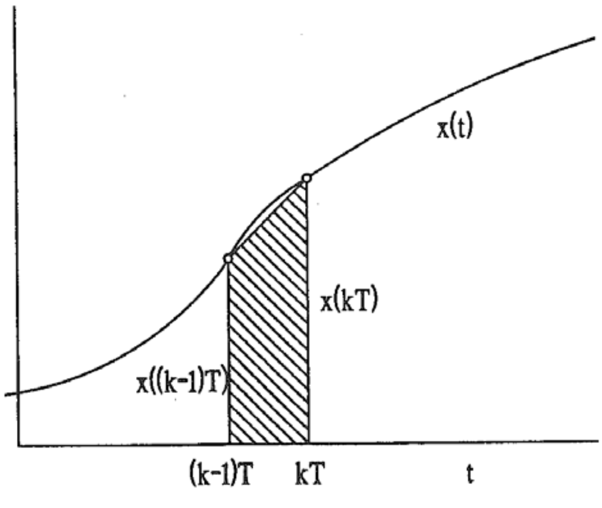

Figure 5.6. Trapezoidal integration.

Table 5.1. Substitutions for Various Integration Methods

Method	
Forward rectangular	$s = (z - 1)/T$
Backward rectangular	$s = (z - 1)/Tz$
Trapezoidal (bilinear transformation)	$s = 2(z - 1)/T(z + 1)$

Example 5.3. Consider the second-order bandpass filter with the passband centered at 10 rad/s and a quality factor of $Q = 5$, which was illustrated in Example 5.1. The s-domain transfer function is

$$H(s) = \frac{Y(s)}{U(s)} = \frac{2s}{s^2 + 2s + 100} \qquad (*)$$

Synthesize an "equivalent" discrete-time filter using the backward rectangular integration method. Since the critical frequency is at 10 rad/s (1.59 Hz), a reasonable sampling frequency would be 10 Hz or the sampling period T will be 0.1 s. The appropriate substitution from Table 5.1 would be

$$s = \frac{10(z - 1)}{z}$$

Substitution of this relation into relation (*) yields

$$H(z) = \frac{2\left[\frac{10(z-1)}{z}\right]}{\left[\frac{10(z-1)}{z}\right]^2 + 2\left[\frac{10(z-1)}{z}\right] + 100}$$

Cleaning up the algebra yields the transfer function

$$H(z) = \frac{0.0909z(z - 1)}{z^2 - 1.0z + 0.4545}$$

This filter has zeros at $z = 0$ and $z = 1$, while there is a complex pair of poles at $z = 0.5 \pm j0.4522$.

The difference equation, which is equivalent to the transfer function, is

$$y(k + 2) = y(k + 1) - 0.4545y(h) + 0.0909u(k + 2) - 0.0909u(k + 1)$$

Example 5.4. Consider the bandpass filter considered in Example 5.1 with transfer function

$$H(s) = \frac{2s}{s^2 + 2s + 100}$$

Synthesize an "equivalent" digital filter employing the bilinear transformation method and a sampling integral of $T = 0.1$ s.

The appropriate bilinear transformation is

$$s = \frac{20(z - 1)}{z + 1}$$

Substitution of this relation into the original s-domain transfer function yields

$$H(z) = \frac{\dfrac{40(z-1)}{z+1}}{400\left(\dfrac{z-1}{z+1}\right)^2 + (2)(20)\left(\dfrac{z-1}{z+1}\right) + 100}$$

Upon cleaning up the algebra yields a z-domain transfer function is

$$H(z) = \frac{0.0740(z - 1)(z + 1)}{z^2 - 1.111z + 0.8519}$$

This filter has zeros at $z = 1$ and $z = -1$ and poles at $z = 0.5556 \pm j0.737$. The difference equation algorithm for real-time filtering is

$$y(k + 2) = 1.111y(k + 1) - 0.8519y(k) + 0.0740u(k + 2) - 0.0740u(k)$$

4. Zero order hold approximation

The zero-order-hold approximation is one of the most popular techniques used by designers to get from a continuous design to a discrete design. This technique simply assumes that the continuous filter we want to approximate is preceded by a zero-order hold (ZOH) and followed by a sampler such that the input and output are sequences of numbers, as shown in Fig. 5.7.

In Section 3.3 we discussed a situation that involved the derivation of a z-domain transfer function for a continuous-time plant driven by a zero-order hold and followed by a sampler. The result of that derivation indicated that the z-domain transfer function from the input of the zero-order hold to the output of the sampler was given by

$$H(z) = \frac{Y(z)}{U(z)} = (1 - z^{-1})\mathscr{Z}\mathscr{L}^{-1}\left[\frac{H(s)}{s}\right] \tag{5.7.1}$$

This is sometimes referred to as the step-invariant method because the step response samples will be identical.

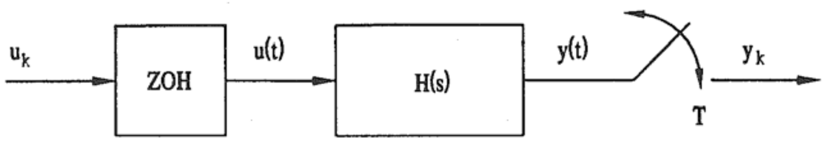

Figure 5.7. Zero-order-hold approximation.

Example 5.7. Given the bandpass transfer function of Example 5.1, let us generate an approximately equivalent digital filter transfer function by the zero-order-hold method for a sampling interval of $T = 0.1$ s.

$$\frac{H(s)}{s} = \frac{1}{s}\frac{2s}{s^2 + 2s + 100}$$

or

$$\frac{H(s)}{s} = \frac{2}{s^2 + 2s + 100} = \frac{2}{(s + 1)^2 + 9.95^2}$$

the z-transform of the samples of the inverse can be established by noting that $b = 9.95$ and thus $bT = 0.995$ and $a = 1$ and thus $aT = 0.1$, so

$$e^{-aT} = 0.9048$$
$$\sin bT = 0.8388$$
$$\cos bT = 0.5445$$

Using entry 12 from Table A.2, the resulting z-transform is

$$\mathfrak{Z}\mathcal{L}^{-1}\left[\frac{H(s)}{s}\right] = \frac{2(0.9048)(0.8388)z}{9.95(z^2 - 0.9853z + 0.8186)}$$

Employing relation (5.7.1) the z-domain transfer function is

$$H(z) = \frac{0.1526(z - 1)}{z^2 - 0.9853z + 0.8186} = \frac{Y(z)}{U(z)}$$

The z-domain poles are at $z = 0.4927 \pm j0.7588$ while the zero is at $z = 1$. The filtering algorithm difference equation is

$$y(k + 2) = 0.9853y(k + 1) - 0.8186y(k) + 0.1526u(k + 1) - 0.1526u(k)$$

----- END -----