ELEC 6061 **Winter 2006**

Lecture 6: Discrete Equivalents

Lecturer: Peyman Gohari

Finding the discrete equivalent of a continuous system is desired when:

1. A discrete equivalent of a continuous plant is needed to simulate a sampled-data system. In a sampled-data system controller is discrete while plant is continuous. To simulate the entire system as a discrete-time system one needs to find the discrete equivalent of the plant (preceded by a hold circuit, and followed by a sampler).

2. To design a digital controller for a continuous-time plant one may start from scratch and directly design a digital controller (this will be discussed in future lectures). Alternatively, one may take advantage of the vast body of knowledge available on continuous-control design by taking the following two steps:

   (a) Design (or have designed) a continuous-time controller for the plant. In his calculations the designer should take into account the $T/2$ seconds delay introduced by sampling and hold.

   (b) Find the discrete equivalent of the above design, for implementation by a digital computer.

   This method of design is called **emulation**.

In this chapter three methods for finding discrete equivalents are discussed. The first method is to use *numerical integration* to approximate a system described by a set of linear differential equations. The second method uses the idea of *pole-zero mapping* to find the discrete-time counterpart of a continuous-time transfer function. Recall that a continuous-time pole located at $s = s_0$ corresponds to a discrete-time pole at $z = e^{s_0 T}$. Finally, we discuss *hold equivalents*, which uses different extrapolation techniques to convert samples into a continuous-time signal.

6.1 Design of discrete equivalents via numerical integration

To obtain the discrete equivalent of a transfer function via numerical integration, we first write the system differential equation, and then apply one of the following numerical integration techniques introduced before:

$$\begin{array}{lcll} \dot{x}(k) & \approx & \frac{x(k+1)-x(k)}{T} & \text{Forward rectangular rule} \\ \dot{x}(k+1) & \approx & \frac{x(k+1)-x(k)}{T} & \text{Backward rectangular rule} \\ \frac{\dot{x}(k)+\dot{x}(k+1)}{2} & \approx & \frac{x(k+1)-x(k)}{T} & \text{Trapezoid rule} \end{array}$$

The operation can be carried out directly on transfer function if one translates the above equations into frequency domain. In our translation, each discrete-time left-shift by n corresponds to a z^n multiplying factor in z-domain, and each $\frac{d^n}{dt^n}$ in continuous-time domain corresponds to an s^n multiplying factor in Laplace domain. Thus we have:

$$\begin{array}{lcll} s & \longleftarrow & \frac{z-1}{T} & \text{Forward rectangular rule} \\ sz & \longleftarrow & \frac{z-1}{T} & \text{Backward rectangular rule} \\ \frac{s+sz}{2} & \longleftarrow & \frac{z-1}{T} & \text{Trapezoid rule} \end{array}$$

which simplifies to:

$$\begin{array}{lcll} s & \longleftarrow & \frac{z-1}{T} & \text{Forward rectangular rule} \\ s & \longleftarrow & \frac{z-1}{Tz} & \text{Backward rectangular rule} \\ s & \longleftarrow & \frac{2}{T}\frac{z-1}{z+1} & \text{Trapezoid rule} \end{array}$$

The trapezoid rule is also called **Tustin's method** in digital control community.

Each of the above relations can be viewed as a map from z-plane to s-plane[1]. Since we are more familiar with s-plane characteristics, it would be useful to find the inverse of each transformation as a map from s-plane to z-plane:

$$\begin{array}{ll} z = sT + 1 & \text{Forward rectangular rule} \\ z = \frac{1}{1-Ts} & \text{Backward rectangular rule} \\ z = \frac{1+Ts/2}{1-Ts/2} & \text{Trapezoid rule} \end{array}$$

It is interesting to see how the stable region of s-plane (that is, the right-half plane described by $Re(s) > 0$) is mapped to z-plane. As illustrated in Figure 6.1, under trapezoid rule the stable region of s-plane is mapped exactly to the stable region of z-plane. This and other observations are summarized below.

1 Under trapezoid rule, the discrete-time system is stable *if and only if* the continuous-time system is stable.

2 Under backward rectangular rule, the discrete-time system is stable *if* the continuous-time system is stable.

[1] The third mapping is called a **bilinear transformation**.

3 Under forward rectangular rule, the discrete-time system is stable *only if* the continuous-time system is stable.

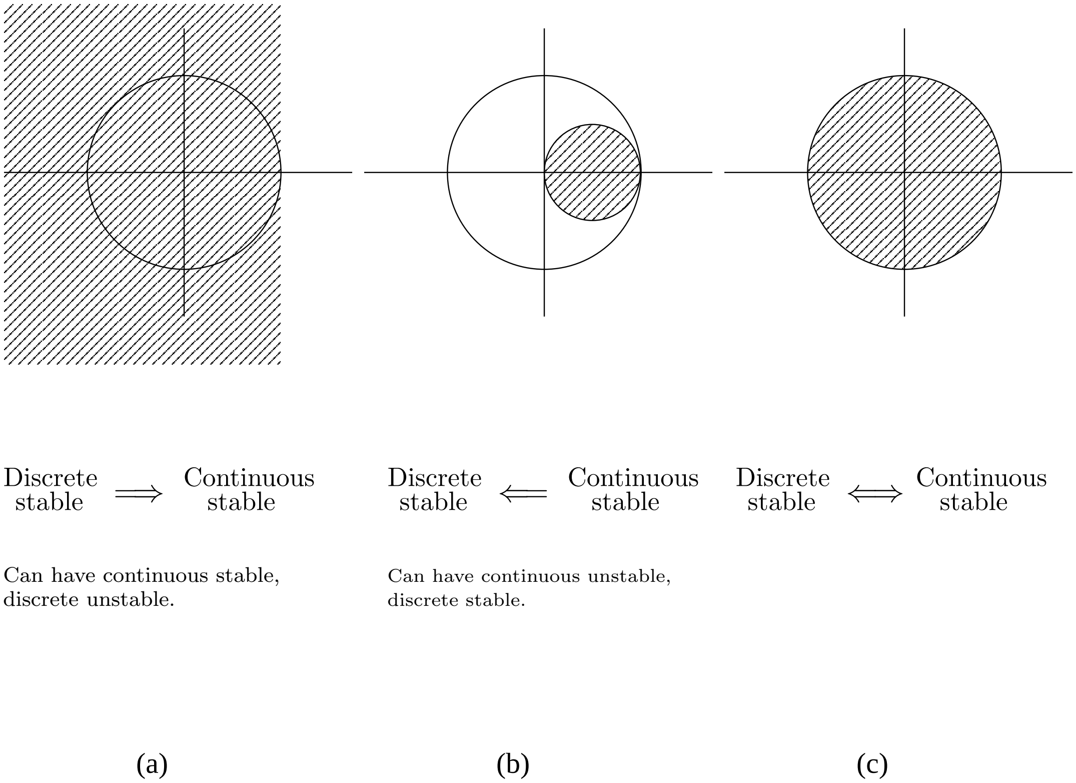

Discrete stable $\Longrightarrow$ Continuous stable
Can have continuous stable, discrete unstable.
Discrete stable $\Longleftarrow$ Continuous stable
Can have continuous unstable, discrete stable.
Discrete stable $\Longleftrightarrow$ Continuous stable
(a)
(b)
(c)

Figure 6.1: The image of the stable region of s-plane under three transformations: (a) forward rectangular, (b) backward rectangular, and (c) bilinear.

Despite the congruence of stability under the trapezoid rule, the scheme suffers from a serious drawback: the entire imaginary axis is mapped to only 2π-length of the unit circle! This is a great amount of distortion, which is overcome by introducing an extension of Tustin's rule. The approach is called bilinear with *prewarping*. The idea is to ensure that at a pre-specified frequency, discrete-time transfer function has exactly the same characteristics (gain and phase) as the corresponding continuous-time transfer function. The transformation is bilinear like Tustin's, however the constant coefficient (which used to be $\frac{2}{T}$) is different:

$$s = \frac{\omega_0}{\tan(\omega_0 T/2)} \frac{z-1}{z+1}$$

Thus, denoting the discrete equivalent of $H(s)$ by $H_{\text{BP}}(z)$ (for bilinear with prewarping), the relation between the continuous- and discrete-time transfer functions is:

$$H_{\text{BP}}(z) = H(s)\Big|_{s = \frac{\omega_0}{\tan(\omega_0 T/2)} \frac{z-1}{z+1}}$$

Now suppose a sinusoid at the continuous frequency of $s = j\omega_0$, which corresponds to the discrete frequency of $z = e^{j\omega_0 T}$, is applied to the systems. We have:

$$\begin{aligned}
H_{\mathrm{BP}}(e^{j\omega_0 T}) &= H(\frac{\omega_0}{\tan(\omega_0 T/2)} \frac{e^{j\omega_0 T} - 1}{e^{j\omega_0 T} + 1}) \\
&= H(\frac{\omega_0}{\tan(\omega_0 T/2)} \frac{e^{j\omega_0 T/2} - e^{-j\omega_0 T/2}}{e^{j\omega_0 T/2} + e^{j\omega_0 T/2}}) \\
&= H(\frac{\omega_0}{\tan(\omega_0 T/2)} j \frac{2\sin(\omega_0 T/2)}{2\cos(\omega_0 T/2)} \\
&= H(j\omega_0)
\end{aligned}$$

That is, a sinusoid at frequency ω_0 experience the same transfer function in both continuous and discrete domains.

Example 1 A third-order low-pass Butterworth filter is described by

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

The discrete equivalent of the filter is calculated using forward rectangular rule, backward rectangular rule, Tustin's bilinear rule, and bilinear rule with prewarping at $\omega = 1$, at sampling rates $f_s = 10,\ 1,\ 0.5$ Hertz (Figure 6.4). Observe that at the high sampling rate of $f_s = 10$ Hertz (60 times the bandwidth) all discrete approximations perform equally well. At the slow rate of $f_s = 1$ Hertz, the forward rule results in an unstable discrete-time system. (Recall that according to Figure 6.1 in this case it is possible for the discrete equivalent of a continuous-time, stable system to be unstable.) At the very slow rate of $f_s = 0.5Hz$, although all techniques are inaccurate, bilinear rule with prewarping has the advantage of generating the same magnitude and phase at the cutoff frequency, which was selected as the prewarping frequency.

□

6.2 Zero-pole matching equivalents

As we saw before a pole s_0 of the Laplace transform of a continuous-time signal is related to a pole z_0 of the z-transform of signal samples according to $z_0 = e^{s_0 T}$. The idea of zero-pole matching is to use this mapping to determine the location of zeros as well. The procedure is outlined below.

1. A continuous-time pole at $s = s_0$ is mapped to a discrete-time pole at $z = e^{s_0 T}$.

2. A continuous-time, finite, zero at $s = s_0$ is mapped to a discrete-time zero at $z = e^{s_0 T}$.

3. Let m and n be the degree of numerator and denominator of a continuous-time transfer function. If $m < n$, then the system will have $n - m$ zeros at infinity. Each continuous-time zero at $s = \infty$ is mapped to a discrete-time zero at $z = -1$. The rationale is that we would like the highest continuous-time frequency of $s = j\infty$ to correspond to the highest discrete-time frequency of[2] $z = e^{j\frac{\pi}{T}T} = -1$.

4. Finally, we adjust gain of the discrete equivalent by making continuous-time and discrete-time systems gains equal at a pre-specified frequency, in other words, we demand the following equality:

$$H_{zp}(e^{j\omega_0 T}) = H(j\omega_0)$$

Often in practice we require the DC gains to be equal, that is, $\omega_0 = 0$, in which case we must have:

$$H_{zp}(1) = H(0)$$

Exercise 1 Compute the discrete equivalent to

$$H(s) = \frac{a}{s+a}$$

by zero-pole matching such that the DC gains of discrete equivalent and original system are equal.

Answer:

$$H_{zp}(z) = \frac{1 - e^{-aT}}{z - e^{-aT}}$$

□

6.3 Hold equivalents

In the final method of computing a discrete equivalent for a continuous-time system, we assume that the system is part of a digital loop, and as such is preceded by a hold circuit and is followed by a sampler (Figure 6.2). It is then desired to find the discrete-time transfer function from input samples $e(k)$ to output samples $u(k)$, which will be a discrete approximation to the continuous-time transfer function from continuous-time input $e(t)$ to continuous-time output $u(t)$.

Naturally the quality of approximation depends on the method used for extrapolating samples.

[2] If it is desired to have a delay of one sampling period (for example, to allow enough time for the computation of output) then one of the continuous-time zeros at $s = \infty$ should be mapped to a discrete-time zero at $z = \infty$. Note that in this case after dividing numerator by denominator we would have $H(z) = Az^{-1} + Bz^{-2} + \cdots$, which gives us the desired unit delay of z^{-1}.

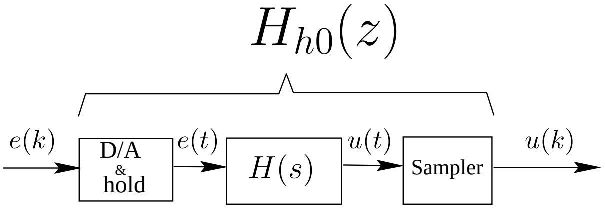


Figure 6.2: Hold equivalent.

6.3.1 Zero-order hold equivalent

In the simplest extrapolation method introduced earlier, samples are held constant until the next sampling instant. We found that in this case the discrete equivalent can be calculated using the following formula:

$$H_{h0}(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{H(s)}{s}\right\}$$

Exercise 2 Compute zero-order-hold equivalent to

$$H(s) = \frac{a}{s+a}$$

Answer:

$$H_{zp}(z) = \frac{1 - e^{-aT}}{z - e^{-aT}}$$

Note that incidentally the answer is the same as previous exercise where pole-zero matching was used; however, this is not true in general. □

6.3.2 Triangle hold equivalent

We now consider a first-order hold, which extrapolates samples by connecting them using straight lines. The impulse response of a first-order hold is shown in Figure 6.3. As evident from the figure, a first-order hold is noncausal, but this does not cause any problem as the resulting discrete-equivalent is in fact causal.

The derivation of first-order and zero-order equivalents are similar (the latter was carried out in Section 4, Lecture 3; the former is left as an exercise). The first-order equivalent $H_{tri}(z)$ of a continuous-time transfer function $H(s)$ is:

$$H_{tri}(z) = \frac{(z-1)^2}{Tz}\mathcal{Z}\left\{\frac{H(s)}{s^2}\right\}$$

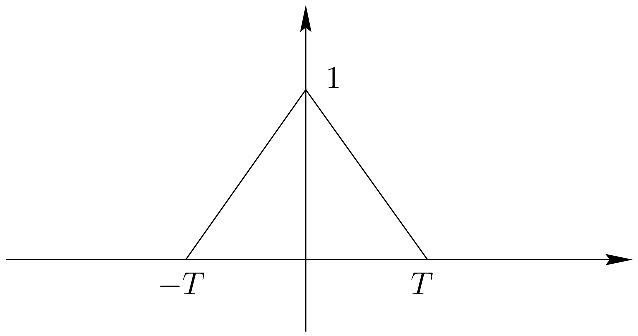


Figure 6.3: The impulse response of a triangle (first-order) hold.

In MATLAB, the statement `sysD=c2d(sys,T,`*parameter*`)` computes the discrete equivalent `sysD` of a continuous-time system `sys` when the sampling period is `T`, using the method specified by *parameter*: `'tustin'` for Tustin's, `'prewarp'` for bilinear with prewarping, `'matched'` for pole-zero matching, `'zoh'` for zero-order hold, and finally `'foh'` for first-order (triangular) hold.

Example 2 Figure 6.5 compares discrete equivalents of a Butterworth filter calculated using pole-zero matching, zero-order hold and first-order (triangular) hold, when sampling rates are $f_s = 1$ and $f_s = 0.5$ Hertz. Observe that even at the very low sampling rate of $f_s = 0.5$ Hz the triangular equivalent shows a very good phase response. □

Fig. 6.4a Magnitude and phase of discrete equivalents

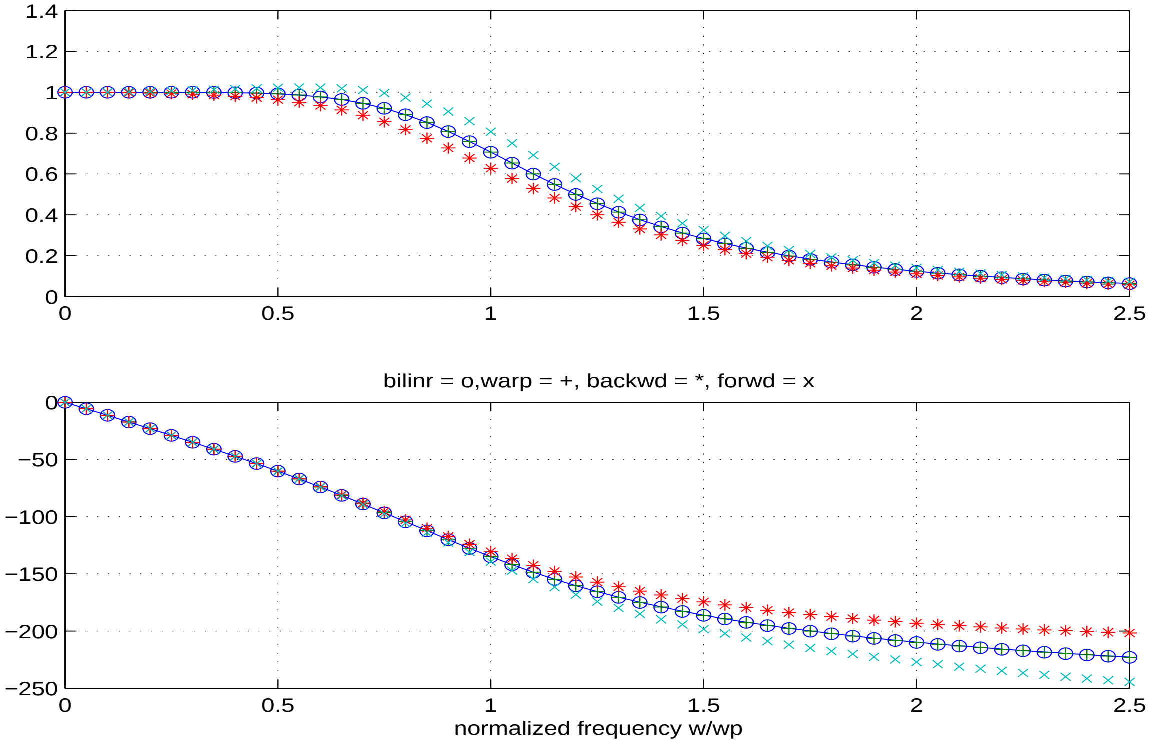
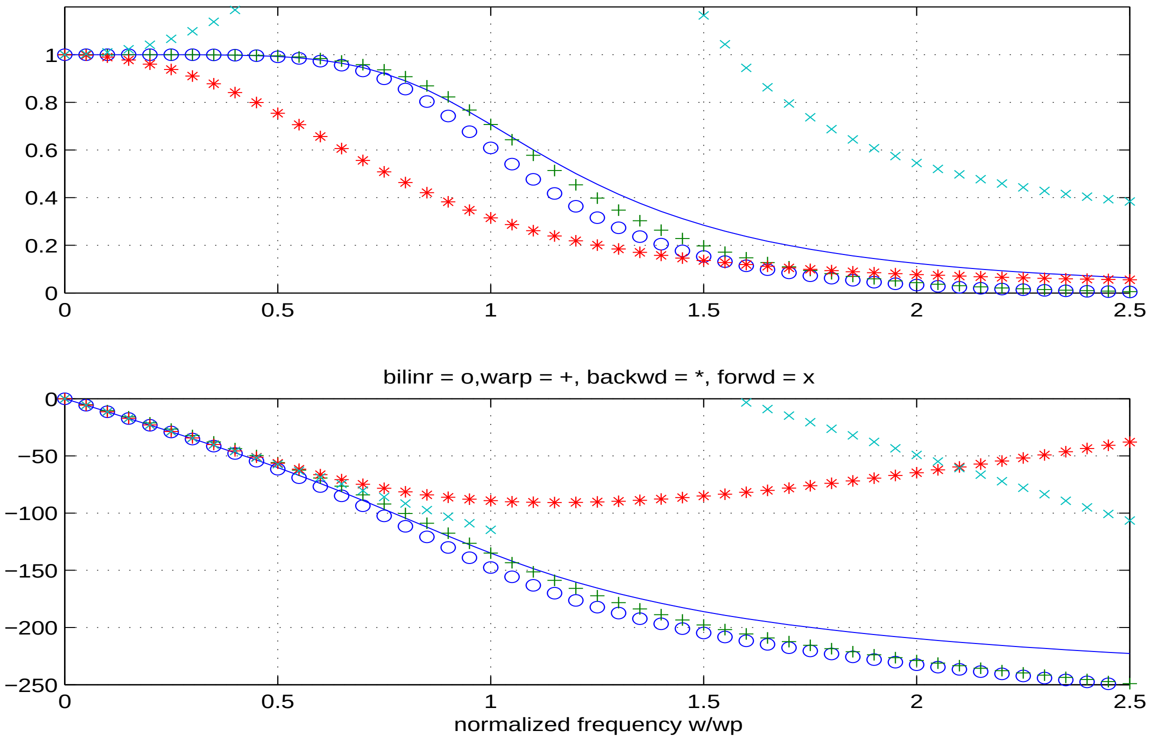


Fig. 6.4b Magnitude and phase of discrete equivalents



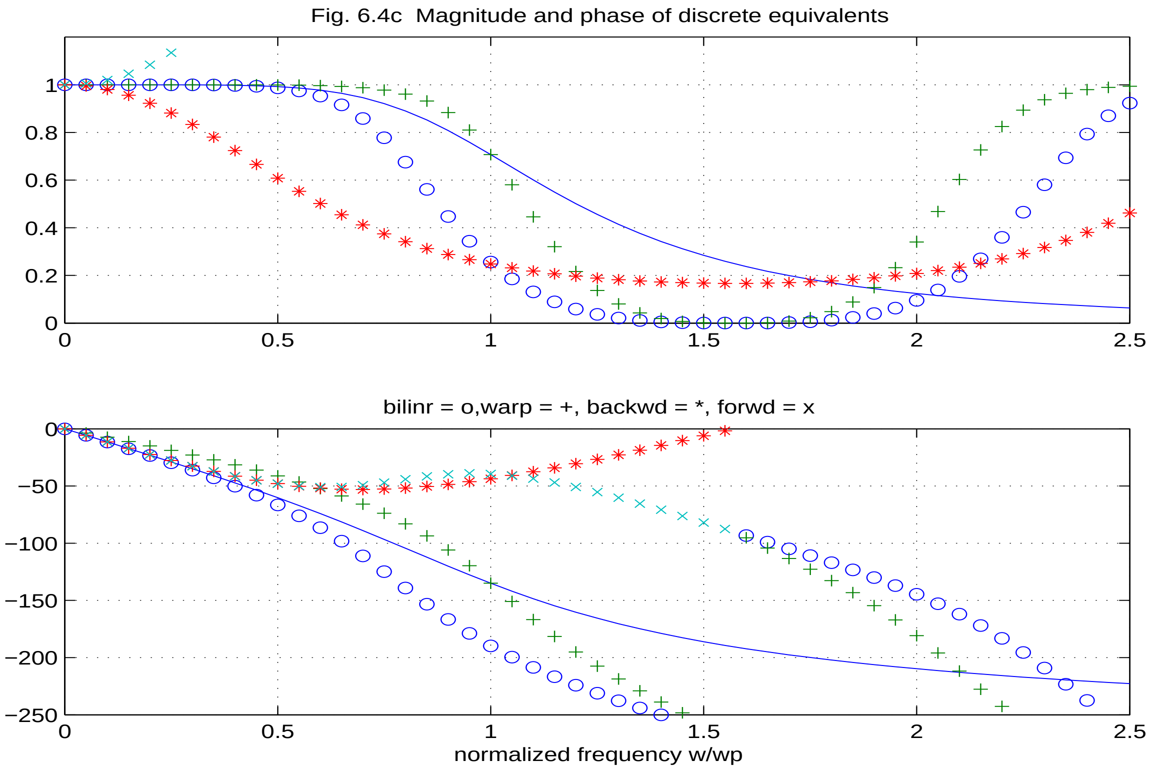


Figure 6.4: Numerical integration of a third-order low-path Butterworth filter using forward rectangular rule, backward rectangular rule, Tustin's bilinear rule, and bilinear rule with prewarping at $\omega = 1$, at sampling rates a) $f_s = 10$, b) $f_s = 1$, and c) $f(s) = 0.5$ Hertz.

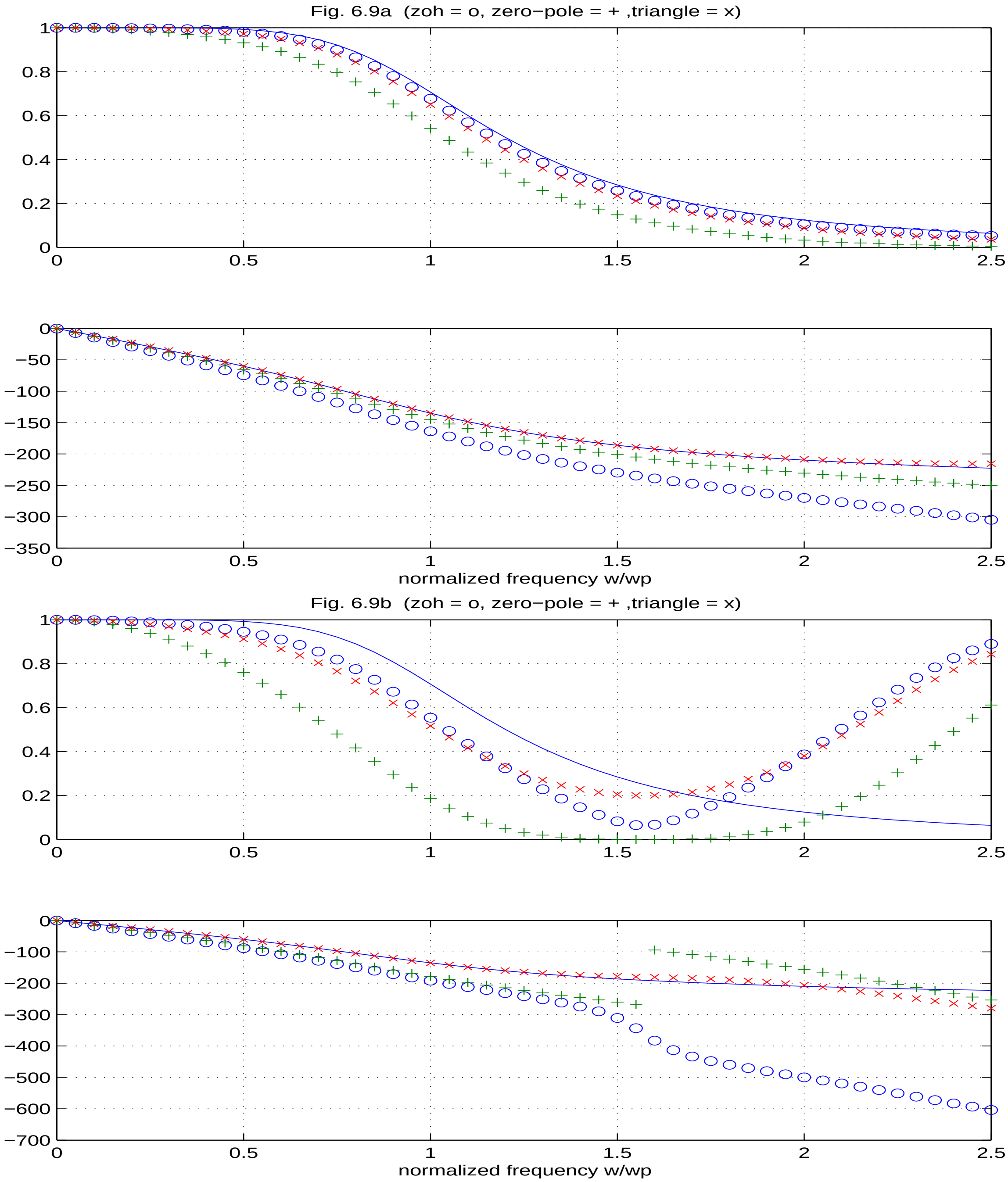


Figure 6.5: Comparing the performance of pole-zero matching, zero-order hold and triangular hold, when sampling rate is a) $f_s = 1$, and b) $f_s = 0.5$ Hertz.