Chapter 6 Design of Discrete Time Controller – Input/Output Approaches

Controller design process is directly coupled with the description of the plant dynamics. Since there are two approaches to plant description, transfer function (I/O mapping) and state space model. The controller design process can be roughly categorized into input/output (I/O) approach and state space approach. The I/O approaches are sometimes called the *classical techniques* since the majority of the contributions were done between 1930s and 1950s by researchers like Evans (root locus), Nyquist, Bode and Wiener. The state-space approach mainly started in the 1960s and is sometimes referred to as the *modern techniques*. In light of recent developments, it is more appropriate to use the terms *I/O approach* and the *state space approach*.

In this chapter, we will first discuss the indirect approaches for designing discrete-time controllers. These approaches involve translating an existing continuous-time controller to a discrete-time controller using various approximations (emulations). The second portion of the chapter will discuss various direct approaches to designing discrete-time controllers in the transfer function domain.

6.1 Approximating Continuous-Time Controllers (Emulations)

The transition from pure-analog to digital control experienced a period where "backward compatibility" is important. Because of the concern for change, or put it in another way – the existing control systems have been proven to work, and the factories are cranking out products, the motivation for change is therefore low. Realizing the benefits of digital control – better reliability, flexibility, expandability, and potentially lower prices, people began to try digital control systems, commonly demanding that the discrete-time systems should mimic the continuous-time controllers they are replacing. The technique to "translate" analog designs into digital designs is thus important in the early days of digital control. As we will show in later discussion, the approximation techniques relay on a relatively fast sampling rate and may not be applicable if physical constraints limit the achievable sampling frequency.

In this section we assume that a continuous-time controller is given as a transfer function, $C(s)$. It is desired to find an algorithm (difference equation) for microprocessors so that the digital controller $C(z)$ approximates the continuous-time controller, see Figure 6.1.

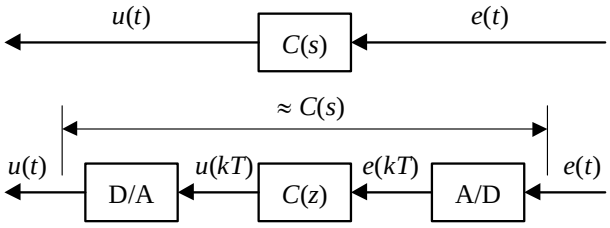


Figure 6.1 Approximating Continuous-Time Controller

Numerical Integration (Approximate Differentiation)

To look for methods to approximate the transfer function, we start by examining a simpler question: what is the equivalent of the differential operator (d/dt or s) in terms of the shift operator (z)? To answer this question, let's start with a single integrator system, i.e.

$$\dot{u}(t) = e(t) \quad \text{or} \quad C(s) = \frac{U(s)}{E(s)} = \frac{1}{s}$$

The solution to the system is

$$u(t) = u(t_0) + \int_{t_0}^{t} e(\tau)\, d\tau\,.$$

At the sample instants

$$u\big((k+1)T\big) = u(T) + \int_{T}^{(k+1)T} e(\tau)\, d\tau \tag{6.1}$$

There are three apparent choices to approximate the above integral in discrete time, i.e. using values only at the sample instants and not in between. Figure 6.2 illustrated these approximations.

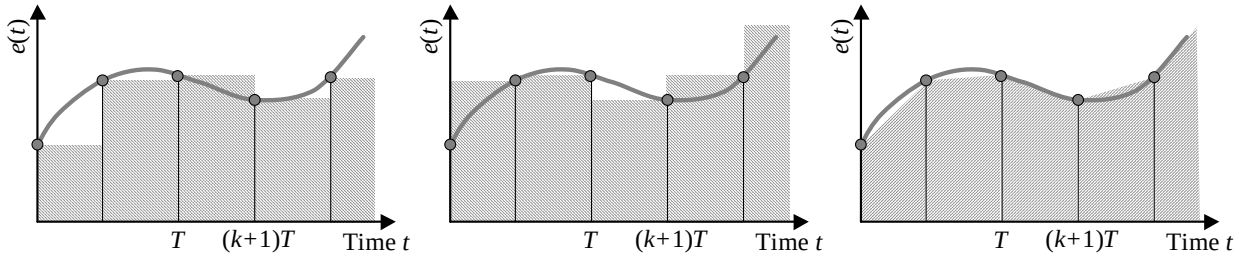


Figure 6.2 Different Types of Integral Approximations

Mathematically, the discrete approximations of Eq. (6.1) can be summarized as follows:

- *Forward Difference*: $u(k+1) \approx u(k) + e(k) \cdot T$
- *Backward Difference*: $u(k+1) \approx u(k) + e(k+1) \cdot T$
- *Trapezoidal Approximation*: $u(k+1) \approx u(k) + \dfrac{\big[e(k+1) + e(k)\big]}{2} \cdot T$

  (*Bilinear Transformation*)
  (*Tustin's Approximation*)

The pulse transfer function from input $E(z)$ to output $U(z)$ is:

- *Forward Difference*: $C(z) = \dfrac{U(z)}{E(z)} = \dfrac{T}{z-1} = \dfrac{Tz^{-1}}{1-z^{-1}}$
- *Backward Difference*: $C(z) = \dfrac{U(z)}{E(z)} = \dfrac{Tz}{z-1} = \dfrac{T}{1-z^{-1}}$
- *Trapezoidal Approximation*: $C(z) = \dfrac{U(z)}{E(z)} = \dfrac{T}{2}\dfrac{z+1}{z-1} = \dfrac{T}{2}\dfrac{1+z^{-1}}{1-z^{-1}}$

Comparing with the pulse transfer functions above with the continuous-time transfer function we see that the discrete-time approximation of the continuous-time transfer function can be obtained by substituting the Laplace operator s with the following:

- *Forward Difference*: $s \;\rightarrow\; \dfrac{z-1}{T}$ i.e. $C(z) = C(s)\Big|_{s \to \frac{z-1}{T}}$

- *Backward Difference*: $s \rightarrow \dfrac{z-1}{Tz}$ i.e. $C(z) = C(s)\Big|_{s\rightarrow\frac{z-1}{Tz}}$
- *Trapezoidal Approximation*: $s \rightarrow \dfrac{2}{T}\dfrac{z-1}{z+1}$ i.e. $C(z) = C(s)\Big|_{s\rightarrow\frac{2}{T}\frac{z-1}{z+1}}$

Example 6.1 *Discrete-Time Approximation of Continuous-Time Transfer Functions*
Using the three approximation methods to find the discrete-time equivalent of a lead compensator

$$C(s) = \frac{10s+1}{s+1}$$

Compare the approximation result by plotting the frequency response of the continuous-time controller and the discrete-time approximation for sampling periods $T = 0.5$ and 1.

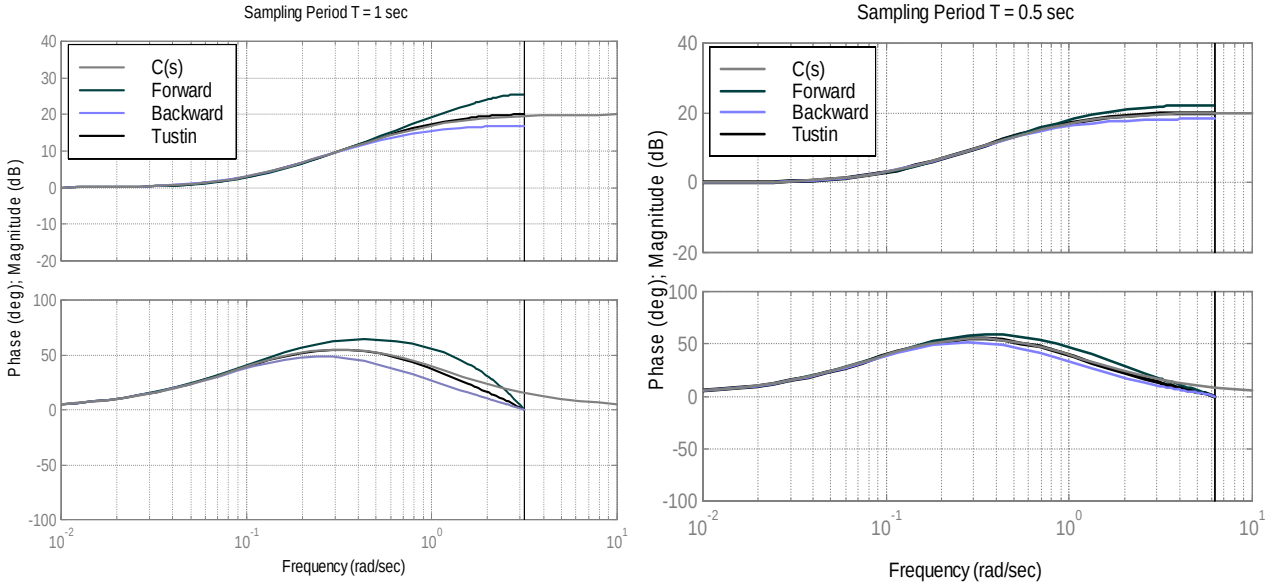
Solution:
The approximations give the following pulse transfer function:

Forward Difference: $C(z) = C(s)\Big|_{s\rightarrow\frac{z-1}{T}} = \dfrac{10z-(10-T)}{z-(1-T)}$

Backward Difference: $C(z) = C(s)\Big|_{s\rightarrow\frac{z-1}{Tz}} = \dfrac{(10+T)z-10}{(1+T)z-1}$

Trapezoidal Approximation: $C(z) = C(s)\Big|_{s\rightarrow\frac{2}{T}\frac{z-1}{z+1}} = \dfrac{(20+T)z-(20-T)}{(2+T)z-(2-T)}$

The frequency response plots are shown below:



From the frequency responses we see that the approximation is generally better for higher sampling rate. The discrepancy between the continuous-time system and the discrete-time system begin to increase around a decade before the Nyquist frequency. This presented a guideline for selecting sampling rate. When using approximation approach to implement discrete-time control, the sampling time should be at least 10 times higher than the highest frequency of interest.

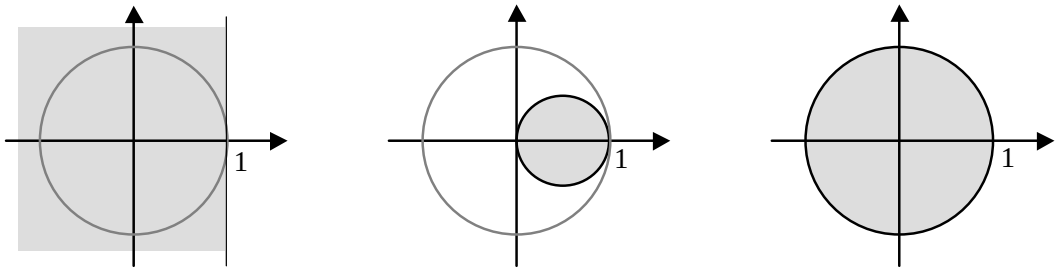
It is of interest to investigate how does the continuous-time stability region, $\mathrm{Re}(s) < 0$, is mapped to the complex *z* plane under the variation approximation. As an example, for forward difference approximation,

$$\mathrm{Re}(s) = \mathrm{Re}\left(\frac{z-1}{T}\right) < 0 \quad \Rightarrow \quad \mathrm{Re}(z-1) < 0 \quad \because T > 0$$

Let $z = \sigma + j\omega$, then

$$\mathrm{Re}(z-1) = \mathrm{Re}(\sigma + j\omega - 1) = \sigma - 1 < 0 \quad \Rightarrow \quad \sigma < 1$$

Hence, by using forward difference approximation, the continuous-time stability region (LHP) is mapped to the half-plane to the left of 1 on the complex *z* plane. Thus, with forward difference approximation, it is possible that a stable continuous-time controller will be approximated by an unstable discrete-time controller. Figure 6.3 illustrated how the continuous-time stability region $\mathrm{Re}(s) < 0$ is mapped on the complex *z* plane for the three approximation methods.



Forward Difference Backward Difference Trapezoidal Approximation

Figure 6.3 Mapping of the Stability Region Between the *s* plane and the *z* Plane

As can be seen from Figure 6.3, backward difference approximation always maps a stable continuous controller to a stable discrete controller. However, some unstable continuous controller can also be transformed into stable discrete controllers. The bilinear transformation (trapezoidal or Tustin's approximation) maps the left half *s* plane into the unit disc. Hence, stable continuous controllers are approximated by stable discrete controllers and unstable continuous controllers are mapped to unstable discrete controllers. In practice, the Tustin's approximation (bilinear transformation) is the approximation of choice for converting continuous-time controllers to discrete-time controllers. In fact, some computer-aided programs (e.g. MATLAB) don't even have the option to approximate with forward or backward difference methods.

Frequency Prewarping

One of the problems with the approximation methods discussed above is that the frequency scale is distorted during the approximation. This will create mismatch in the response of the system if the controller design is based on precise frequency requirements. For example, band-pass and notch filters are designed with specific passing band and notch frequency. The digital filter obtained using the approximation methods may not have the pass-band and notch at the correct frequency. This phenomenon is called *frequency warping*.

To understand the frequency warping phenomenon, we will use the bilinear transformation as an example. Recall that to evaluate the frequency response of a transfer function, for continuous-time systems, the transfer function is evaluated along the imaginary axis, i.e. $s = j\omega$; for discrete-time systems, the pulse transfer function is evaluated along the unit circle, i.e. $z = e^{j\omega T}$. When the discrete pulse transfer function is obtained by using the bilinear transfer

function on a continuous transfer function, the relationship between the "continuous frequency" ω_C and the approximated "discrete frequency" ω_D can be obtained as

$$j\omega_C = s \rightarrow \frac{2}{T}\frac{z-1}{z+1} = \frac{2}{T}\frac{e^{j\omega_D T}-1}{e^{j\omega_D T}+1} = \frac{2}{T}\frac{e^{j\omega_D T/2}-e^{-j\omega_D T/2}}{e^{j\omega_D T/2}+e^{-j\omega_D T/2}} = j\frac{2}{T}\tan\left(\frac{\omega_D T}{2}\right)$$

Hence we have the following relationship

$$\omega_C = \frac{2}{T}\tan\left(\frac{\omega_D T}{2}\right) \tag{6.2}$$

The frequency scale is distorted. Equation (6.2) can be interpreted as follows. Assume that a continuous-time filter is designed to block signals at frequency ω^0. After bilinear transformation, the discrete-time filter will instead block signal transmission at frequency ω_0, where

$$\omega^0 = \frac{2}{T}\tan\left(\frac{\omega_0 T}{2}\right), \text{ i.e. } \omega_0 = \frac{2}{T}\tan^{-1}\left(\frac{T\omega^0}{2}\right).$$

Figure 6.4 shows the distortion of the frequency scale.

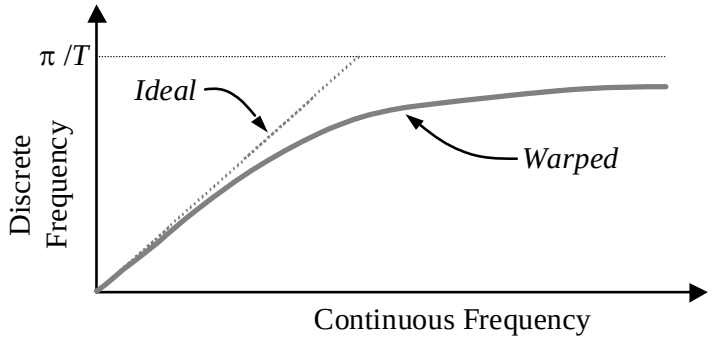


Figure 6.4 Frequency Distortion (Warping) Due to Bilinear Transformation

At low frequency, the distortion is small. However, close to the Nyquist frequency, the distortion becomes significant. If the continuous-time device to be approximated is designed to work at a specific (critical) frequency ω^* (e.g. the center frequency of a band-pass filter, the notch frequency of a notch filter), we can modify the bilinear transformation to match the continuous-time design at that specific frequency by using "*frequency prewarping*":

- *Bilinear Transformation with Prewarping*:

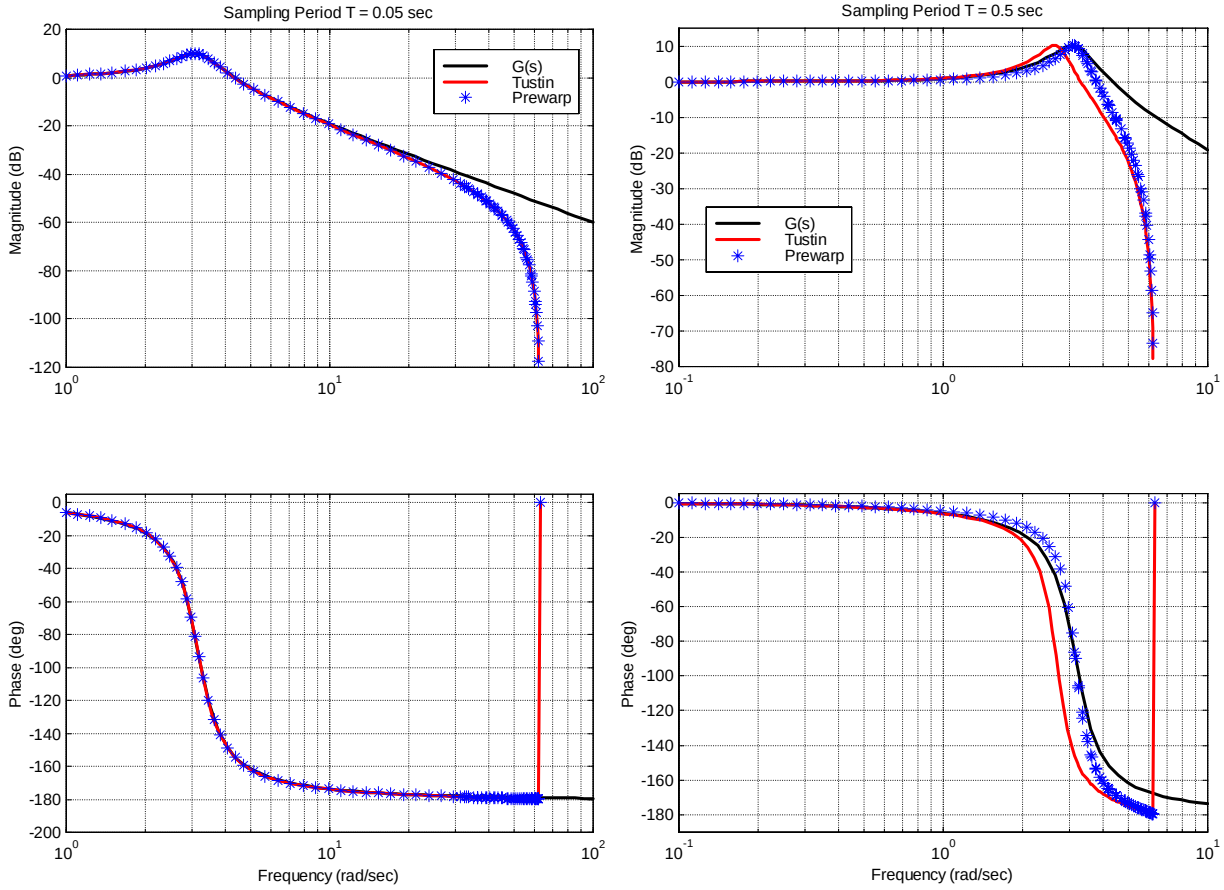
$$s \rightarrow \frac{\omega^*}{\tan\left(\frac{\omega^* T}{2}\right)}\frac{z-1}{z+1} \quad \text{i.e.} \quad C(z) = C(s)\Big|_{s\rightarrow \frac{\omega^*}{\tan\left(\frac{\omega^* T}{2}\right)}\frac{z-1}{z+1}}$$

Using the above transformation we can force the continuous-time controller and its approximation have the same value at frequency ω^*. There is, however, still distortions at other frequencies.

Example 6.2 *Bilinear Transformation with Prewarping*
Suppose we have a continuous-time design

$$G(s) = \frac{10}{s^2+s+10}$$

that is to be translated to the discrete-time. The critical frequency ω^* is around $\sqrt{10}$. When ω^* is much smaller than the Nyquist frequency (e.g. when T = 0.05 sec, $\omega_N = 20\pi$), the distortion will be negligible. When we have a larger sampling period (e.g. when T = 0.5 sec, $\omega_N = 2\pi$), frequency prewarping will become necessary. This can be seen from the following plots:



It can be seen from the above figures, the prewarped frequency response coincided with the continuous-time controller $G(s)$ at the critical frequency $\omega^* = \sqrt{10} = 3.162$.

Selection of Sampling Rate

The choice of sampling rate depends on many issues. One way to determine the sampling period is to use continuous-time arguments. The sampled system can be approximated by a hold circuit, followed by the continuous-time system. For small sampling periods, the transfer function of the hold circuit can be approximates as

$$\frac{1-e^{-sT}}{sT} \approx \frac{1-1+sT-(sT)^2/2+-\cdots}{sT} = 1-\frac{sT}{2}+-\cdots$$

The first two terms correspond to the series expansion of $e^{-sT/2}$. That is, for small T, the hold can be approximated by a time delay of a half of a sampling period. Assume that the system's phase margin can be reduced by 5° to 15°. This gives the following rule of thumb

$$T\omega_C \approx 0.15 \text{ to } 0.5$$

where ω_C is the gain cross-over frequency (in rad/sec) of the continuous-time system. This implies that the sampling rate should be about 10 to 40 times larger than the cross-over frequency.

6.2 Discrete-Time Equivalent of PID Controllers

Many practical control problems are solved by PID controllers or their variants. The "textbook" version of a PID controller can be represented by:

$$u(t) = K_P \cdot e(t) + K_I \cdot \int_0^t e(\tau)\, d\tau + K_D \cdot \frac{d}{dt} e(t) = K_P \left[e(t) + \frac{1}{T_I} \cdot \int_0^t e(\tau)\, d\tau + T_D \cdot \frac{d}{dt} e(t) \right] \quad (6.3)$$

where the error signal $e(t)$ is the difference between the reference signal (the set point) and the plant output (the measured variable). K_P is the proportional gain, K_I is the integral gain, and K_D is the derivative gain. T_I is the integration time or reset time and T_D is the derivative time or rate time. The relationship between the gain description and the time constant description are

$$T_D = \frac{K_D}{K_P} \quad \text{and} \quad T_I = \frac{K_P}{K_I}.$$

The continuous-time transfer function of a PID controller can be obtained by taking the Laplace transform of Eq. (6.3), i.e.

$$C_{PID}(s) = \frac{K_D s^2 + K_P s + K_I}{s} = \frac{K_P \left(T_I T_D s^2 + T_I s + 1 \right)}{T_I s} \quad (6.4)$$

From Eq. (6.4), it is obvious that a "textbook" PID controller is non-causal and cannot, and should not, be implemented. The main reason is that the derivative term is non-causal and that it amplifies high frequency noise in the measured signals. Hence, the gain of the derivative action must be limited. This can be achieved by introducing an additional low-pass filter to the derivative action:

$$K_D s \approx \frac{K_D s}{\tau_L s + 1}$$

where τ_L is typically chosen to be between $K_D/3$ and $K_D/20$. With the augmentation of a low pass filter, the modified continuous-time PID controller can be written as

$$C_{PID}(s) = \frac{K_D s^2 + K_P s + K_I}{s(\tau_L s + 1)} = \frac{K_P \left(T_I T_D s^2 + T_I s + 1 \right)}{T_I s (\tau_L s + 1)} \quad (6.5)$$

which introduced two zeros, a pole at the origin and another "fast" pole. With the continuous-time PID controller described in Eq. (6.5), any of the previous approximation methods can be used to approximate a PID controller. In particular, Tustin's approximation and ramp invariant methods are both good choices if the sampling time constraint can be met.

In order to "preserve" the PID structure, it is common to use the bilinear transformation to approximate the integral action, and to use the backward difference to approximate the differentiation action. The reason backward difference is used instead of bilinear approximation is that the later will introduce a pole at $z = -1$. Using this approximation, a continuous-time PID controller law, Eq. (6.3) can be written as:

$$
\begin{aligned}
u(k) &= K_P \cdot e(k) + K_I \cdot \frac{T}{2}\frac{q+1}{q-1} \cdot e(k) + K_D \cdot \frac{q-1}{Tq} \cdot e(k) \\
&= \left(K_P - \frac{K_I T}{2} \right) \cdot e(k) + \left(K_I T \right) \sum_{j=0}^{k} e(j) + \frac{K_D}{T} \cdot \left[e(k) - e(k-1) \right] \\
&= K_{P(\text{Digital})} \cdot e(k) + K_{I(\text{Digital})} \cdot \sum_{j=0}^{k} e(j) + K_{D(\text{Digital})} \cdot \left[e(k) - e(k-1) \right]
\end{aligned}
\tag{6.6}
$$

where in Eq. (6.6), q is the one-step advance operator. The above relationship is obtained from the pure approximation point-of-view. There are many practical issues that need to be considered. For example, it is sometimes necessary to move the derivative action to the feedback path and not operate on the reference signal. Integrator anti-windup, where the summation term in (6.9) is is imposed with a saturation, is also necessary for practical implementation of the integral action.

6.3 Direct Design – Input/Output Approaches

In the previous sections, we assumed that a continuous-time controller $C(s)$ has been designed to satisfies the system performance and robustness specifications. Approximation techniques can then be employed to discretize the continuous-time controller. In the direct designs, the continuous-time plant transfer function is first discretized using the appropriate hold device equivalent. Then, the discrete-time controller is designed directly based on the discretized plant transfer function and appropriate design techniques. The control design block diagram is shown in Figure 6.5.

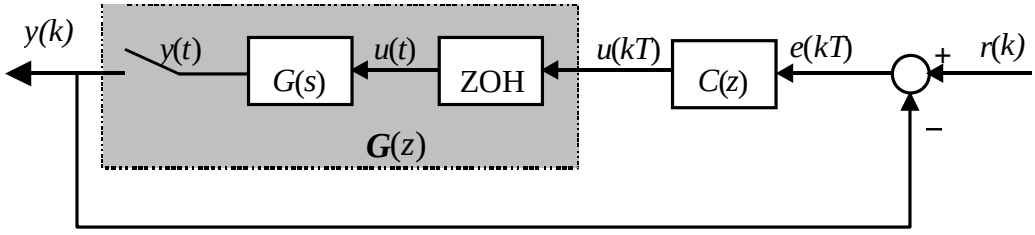


Figure 6.5 Discrete-Time Feedback Control Block Diagram

The closed-loop transfer function from (sampled) reference command $r(t)$ to the (sampled) plant output $y(t)$ is

$$
G_{CL}(z) = \frac{Y(z)}{R(z)} = \frac{G(z)C(z)}{1 + G(z)C(z)} = \frac{L(z)}{1 + L(z)} \tag{6.7}
$$

where $L(z) = G(z)C(z)$ is the (*open*) *loop transfer function*. The closed-loop characteristic equation is

$$
A_{CL}(z) = 1 + G(z)C(z) = 1 + L(z) = 0 \tag{6.8}
$$

For a realizable loop transfer function

$$
L(z) = G(z)C(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_{n-1} z + b_n}{z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n}
$$

The closed-loop characteristic equation can also be written as

$$
A_{CL}(z) = \textit{numerator}(1 + L(z)) = A(z) + B(z) = 0 \tag{6.9}
$$

Notice that from Eq. (6.9), the number of closed-loop poles always matches the number of open loop poles and is equal to n. In the case that $A(z)$ and $B(z)$ have common roots (i.e. pole-zero cancellation in the open loop transfer function), the cancelled open-loop poles remain as the closed-loop poles. Therefore, cancellation of *unstable* poles in the controlled plant and the controller should never take place.

Z-Plane Discrete Pole Location Specifications

In the continuous time domain, the time domain specifications of a system (open or closed loop) relates to the system's pole location. For example, the damping ratio and percent overshoot for a step response is related to the pole's angle in the complex S-Plane; settling time is related to the pole's real part; natural frequency is the magnitude of the pole. Consider now the discrete-time domain signals as sampled from the continuous-time domain, i.e. the poles in the Z-domain is related to the poles in the S-domain by

$$z = e^{sT}$$

Under the s-to-Z mapping, vertical lines of constant real part in the s-plane map to circles of constant radius in the z-plane. Horizontal lines of constant imaginary part in the s-plane map to radial lines of constant angle in the z-plane. Lines of constant damping ratio map into logarithmic spirals in the z-plane. Circles of constant natural frequency in the s-plane map to radial curves in the z-plane. Figure 6.6 shows the lines of constant damping ratio and constant natural frequency according to the pole location mapping. One can use this mapping as guidelines to specify the desirable discrete time pole locations for the corresponding time domain specifications.

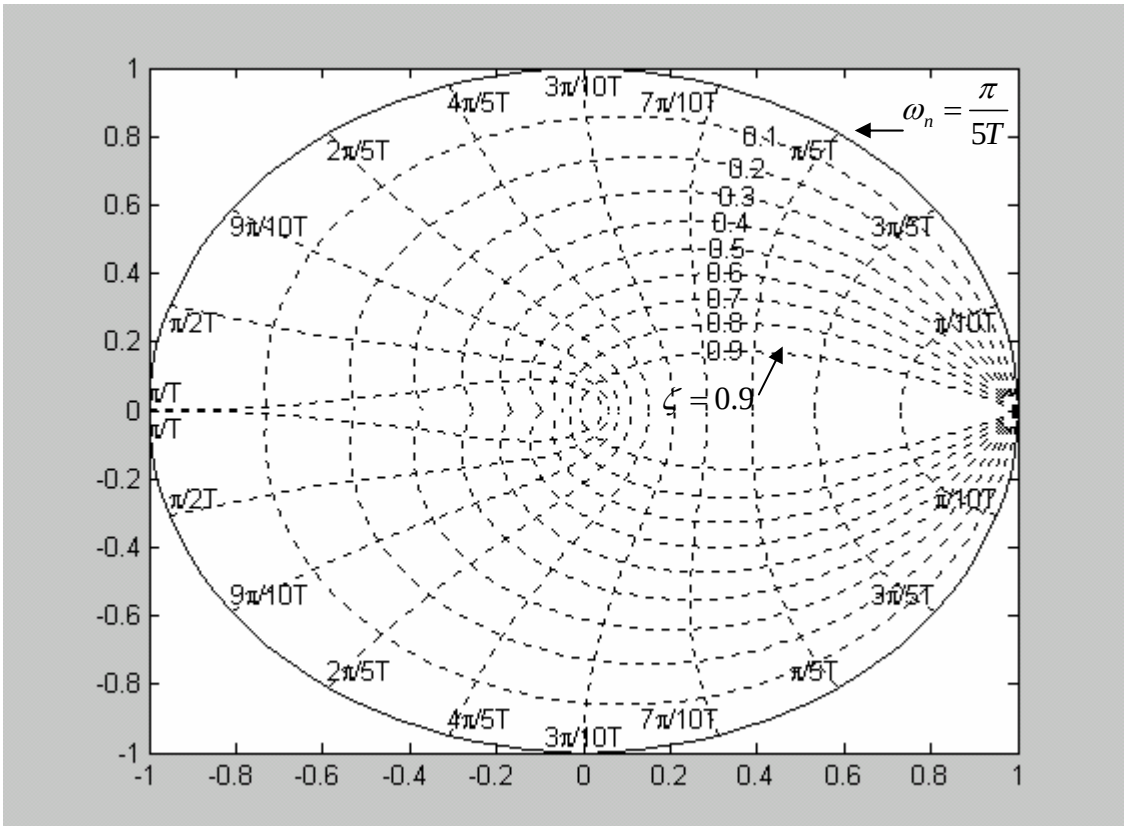


Figure 6.6 Z-domain lines of constant damping ratio and natural frequency

Figure 6.6 can be generated in Matlab by the command zgrid. Customarily, only the upper half of the plot is needed for the design specifications and analysis. The command axis([-1 1 0 1]) will make the plot for only the upper half of the unit circle.

Pole Placement – Root Locus Design

The root locus methods developed for continuous-time systems can be extended to discrete-time systems without modification, except that the stability boundary is changed from the $j\omega$ axis on the *s* plane to the unit circle on the *z* plane.

The reason the root locus method can be extended to the discrete-time system is that the close-loop characteristic equation, Eq. (6.9), for the discrete-time system is of the same form as that of the continuous-time system. However, the pole locations for closed-loop systems on the *z* plane must be interpreted differently form those on the *s* plane.

Angle and Magnitude Conditions

From Eq, (6.8) or (6.9), the closed-loop characteristic equation can be written as

$$L(z) = -1$$

Since $L(z)$ is a complex quantity, the above equation can be split into two equations by equating the angle and the magnitude of the two sides, respectively. Hence we obtain

- *Angle Condition*: $\angle L(z) = \pm 180° \cdot (2k+1), \quad k = 0, 1, 2, \cdots$
- *Magnitude Condition*: $|L(z)| = 1$

The values of *z* that satisfy both the angle and magnitude conditions are the roots of the characteristic equation, or the closed-loop poles.

A plot of the points on the complex plane satisfying the angle condition alone is the root locus. The roots of the characteristic equation (the closed-loop poles) corresponding to a given value of the gain can be located on the root loci by use of the magnitude condition. The details of applying the angle and magnitude conditions to obtain the closed-loop poles are identical to that of the continuous-time systems and will not be repeated here.

Example 6.3 *Root Locus Methods*

Let's look at the digital integral control of a first order continuous-time plant

$$G(s) = \frac{1}{s+1}$$

The discrete-time integral controller is given by

$$C(z) = \frac{K}{1-z^{-1}} = \frac{K\,z}{z-1}$$

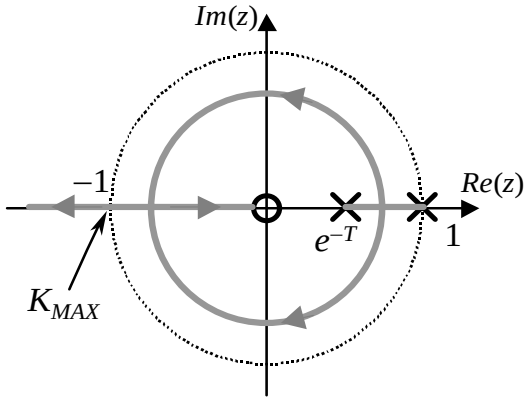
where *K* is the integral control gain. The ZOH equivalent discrete-time plant transfer function is given by

$$G(z) = (1-z^{-1})\mathsf{Z}\left[\frac{G(s)}{s}\right] = (1-z^{-1})\mathsf{Z}\left[\frac{1}{s(s+1)}\right] = \frac{z-1}{z}\left(\frac{z}{z-1} - \frac{z}{z-e^{-T}}\right) = \frac{1-e^{-T}}{z-e^{-T}}$$

The closed-loop characteristic equation is

$$1+G(z)C(z)=1+\frac{1-e^{-T}}{z-e^{-T}}\cdot\frac{K\,z}{z-1}=0 \quad\Rightarrow\quad 1+K\frac{(1-e^{-T})z}{(z-1)(z-e^{-T})}=0$$

The corresponding root locus is shown in the right. At the stability limit (i.e. at K_{MAX}) $z = -1$. Hence we can calculate that



$$K_{MAX}=2\frac{1+e^{-T}}{1-e^{-T}}$$

One observation of the above equation is that as the sampling period increases, the maximum integral gain decreases. However, in the limit, we see that

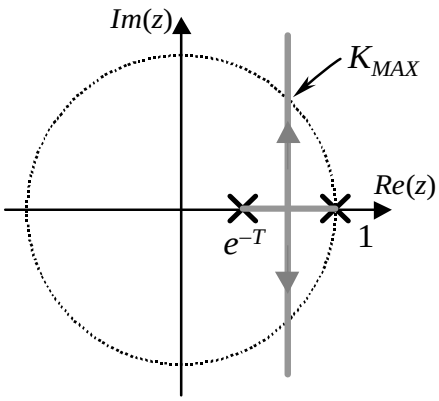
$$K_{MAX}>2$$

If an additional delay is introduced in the plant model to accommodate the computation delay during implementation, i.e.

$$G(z)=\frac{1-e^{-T}}{z-e^{-T}}\cdot z^{-1}=\frac{1-e^{-T}}{z(z-e^{-T})}$$

The corresponding closed-loop characteristic equations is

$$1+G(z)C(z)=1+\frac{1-e^{-T}}{z(z-e^{-T})}\cdot\frac{K\,z}{z-1}=0 \quad\Rightarrow\quad 1+K\frac{(1-e^{-T})}{(z-1)(z-e^{-T})}=0$$



The corresponding root locus is shown in the left. At the stability limit (i.e. at K_{MAX}) $z = e^{j\theta}$. Hence, we can calculate the maximum value of the integral gain by substituting $z = e^{j\theta}$ into the above characteristic equation

$$1+K_{MAX}\frac{(1-e^{-T})}{(e^{j\theta}-1)(e^{j\theta}-e^{-T})}=0$$

Solving the above equation we have

$$K_{MAX}=1$$

Remark It is important to note that the poles of the closed-loop transfer function determines the natural modes of the system. The transient response and the frequency response behavior, however, are strongly influenced by the zeros of the closed-loop transfer function. On the *s* plane, adding a zero on the negative real axis near the origin increases the maximum overshoot in response to a step input. Such a zero on the *s* plane is mapped to a zero on the positive real axis on the *z* plane between 0 and 1. Therefore, on the *z* plane, adding a zero on the positive real axis between 0 and 1 will increase the maximum overshoot. In fact, moving a zero toward the point $z = 1$ will greatly increase the maximum overshoot.

Frequency Domain Design

Continuous-time frequency domain design techniques are well established and relatively easy to understand. The commonality among all the frequency domain techniques is the process of

evaluating the frequency response of a LTI system. For continuous-time systems, this is achieved by evaluating the continuous-time transfer function $G(s)$ on the $j\omega$ axis, i.e. replacing the Laplace variable s with the complex number $j\omega$. For discrete-time systems, the frequency response is obtained by evaluating the pulse transfer function $G(z)$ on the unit circle, i.e. replacing the z-transform variable z with the complex number $e^{j\omega T}$. Since $G(e^{j\omega T})$ is not a *rational function* of ω, the simplicity of plotting logarithmic frequency response diagrams that are used in conventional frequency domain techniques will not carry through.

This difficulty can be overcome by transforming the pulse transfer function in the z domain into that in the w domain. The transformation is a bilinear transformation commonly called the w transform.

w Transform

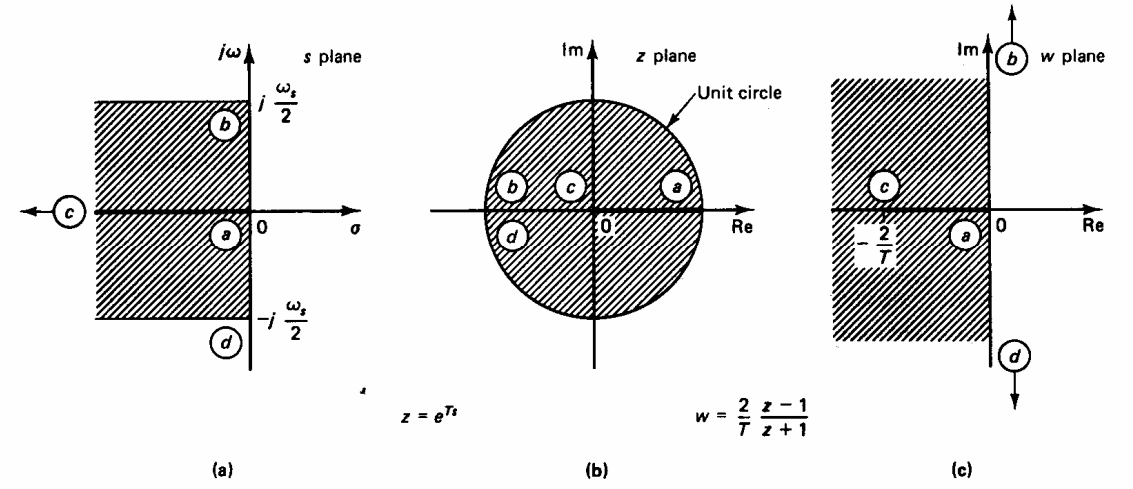
The w transform is defined by

$$z = \frac{1+(T/2)w}{1-(T/2)w} \tag{6.10}$$

where T is the sampling period. By converting a given pulse transfer function in the z domain into a rational function of w, the conventional frequency response methods can be used to obtain a controller in the w domain. The corresponding z domain controller can be obtained by applying the inverse w transform defined by

$$w = \frac{2}{T}\frac{z-1}{z+1}. \tag{6.11}$$

Through the z-transformation and the w transformation, the primary strip of the left half of the s plane is first mapped to the inside of the unit circle on the z plane and then mapped onto the entire left half of the w plane. The two mapping process are depicted in Figure 6.5.



– K. Ogata, *Discrete-Time Control Systems*, Prentice Hall, 2nd Ed., 1995

Figure 6.5 Mapping Among the s Plane, z Plane, and w Plane

Notice that in Figure 6.5, the origin of the z plane is mapped into the point $w = -2/T$ on the w plane. Notice also that, as s varies from 0 to $j\omega_S/2$ along the $j\omega$ axis on the s plane, z varies from 1 to −1 along the unit circle on the z plane, and w varies from 0 to ∞ along the imaginary axis on the w plane.

Although the left half of the *w* plane corresponds to the left half of the *s* plane and the imaginary axis of the *w* plane corresponds to the imaginary axis of the *s* plane, there are differences between the two planes. The major difference is that the behavior on the *s* plane over the frequency range $-\frac{1}{2}\omega_S \le \omega \le \frac{1}{2}\omega_S$ maps to the range $-\infty < v < \infty$, where *v* is the fictitious frequency on the *w* plane. This means that, although the frequency response characteristics of the analog controller (*w* plane) will be reproduced by the discrete controller (*z* plane), the frequency scale on which the response occurs will be compressed from an infinite interval in the analog controller to a finite interval in the discrete controller.

Once the plant pulse transfer function $G(z)$ is transformed into $G(w)$ by means of the *w* transformation, it may be treated as a conventional transfer function in the *w* domain. Since $G(jv)$ is a rational function of the fictitious frequency *v*, conventional frequency domain controller design techniques, e.g. lead-lag compensator, loop shaping, QFT, ...etc, can be used in the *w* domain to design a controller $C(w)$. Once a controller is designed, the inverse *w* transformation can be applied to transform $C(w)$ into $C(z)$.

Since the *w* transformation is a type of bilinear transformation, frequency distortion is present. The fictitious frequency, *v*, in the *w* domain and the actual frequency ω is related as follows:

$$w\Big|_{w=jv} = jv = \frac{2}{T}\frac{z-1}{z+1}\Bigg|_{z=e^{j\omega T}} = \frac{2}{T}\frac{e^{j\omega T}-1}{e^{j\omega T}+1} = j\frac{2}{T}\tan\left(\frac{\omega T}{2}\right)$$

or

$$v = \frac{2}{T}\tan\left(\frac{\omega T}{2}\right) \tag{6.12}$$

Eq. (6.12) gives the relationship between the actual frequency ω and the fictitious frequency *v*. Note that as the actual frequency ω moves from $-\frac{1}{2}\omega_S$ to 0, the fictitious frequency *v* moves from $-\infty$ to 0. And as ω moves from 0 to $\frac{1}{2}\omega_S$, *v* moves from 0 to ∞.

Eq. (6.12) can also be used to translate the actual frequency into the fictitious frequency. For example, if the bandwidth specification is ω_B in the actual frequency, then the corresponding bandwidth specification in the *w* domain is

$$v_B = \frac{2}{T}\tan\left(\frac{\omega_B T}{2}\right).$$

Similarly, $G(jv_1)$ corresponds to $G(e^{j\omega_1 T})$, where

$$\omega_1 = \frac{2}{T}\tan^{-1}\left(\frac{v_B T}{2}\right).$$

Design Procedure in the w Domain

The direct digital design procedure for a discrete control system shown in Figure 6.5 using the *w* transformation can be stated as follows:

1. Obtain a discrete-time equivalent pulse transfer function $G(z)$ of the continuous-time plant $G(s)$ using an appropriate hold device. For a ZOH equivalent:

$$G(z) = (1-z^{-1})\mathcal{Z}\left[\frac{G(s)}{s}\right]$$

2. Transform $G(z)$ into a transfer function $G(w)$ in the w domain through the w transformation, Eq. (6.10), i.e.

$$G(w) = G(z)\Big|_{z=\frac{1+(T/2)w}{1-(T/2)w}}$$

3. Calculate the fictitious frequency response by substituting $w = jv$ into $G(w)$.

4. Translate critical frequencies ω^* into the corresponding fictitious frequencies v^* through the following relationship:

$$v^* = \frac{2}{T}\tan\left(\frac{\omega^* T}{2}\right)$$

5. Use your favorite frequency domain design method to find a controller $C(w)$ that will satisfy the design specifications in the w domain.

6. Transform the controller $C(w)$ into the corresponding discrete-time controller $C(z)$ through the inverse w transformation, Eq. (6.11), i.e.

$$C(z) = C(w)\Big|_{w=\frac{2}{T}\frac{z-1}{z+1}}.$$

Example 6.4 *Controller Design in the w Plane*
Given the following plant:

$$G(s) = \frac{1}{s(s+1)},$$

Find a discrete time controller $C(z)$ such that if the plant in under digital control with sample period $T = 0.01$ [sec], the following specifications are satisfied:

- Phase Margin (PM) > 40° at gain cross-over frequency greater than 1 [Hz]
- Gain margin (GM) > 10 dB at phase cross-over frequency greater than 1 [Hz]
- Steady state error < 5% for unit ramp input.

Solution:
1. Find ZOH equivalent plant model:

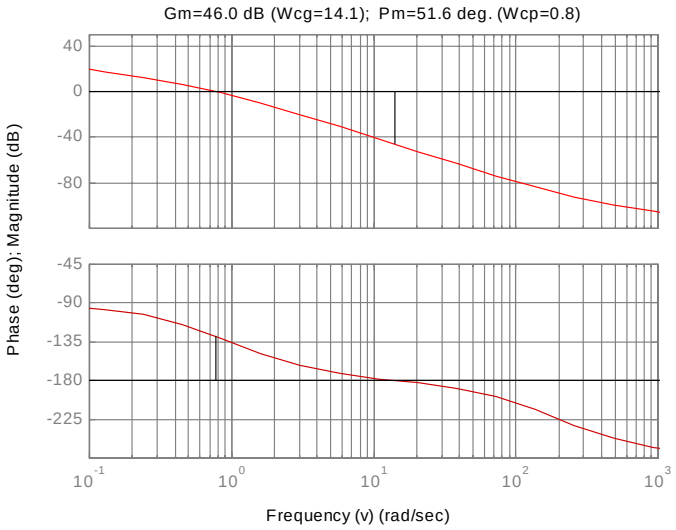
$$G(z) = (1-z^{-1})\mathcal{Z}\left[\mathcal{L}^{-1}\left[\frac{G_P(s)}{s}\right]\right] = (1-z^{-1})\mathcal{Z}\left[\mathcal{L}^{-1}\left[\frac{1}{s^2(s+1)}\right]\right] = \frac{4.9834\times10^{-5}(z+0.9967)}{(z-1)(z-0.99005)}$$

2. Transform $G_P(z)$ into the w-plane with the following transformation:

$$z = \frac{1+(T/2)w}{1-(T/2)w} = \frac{1+0.005w}{1-0.005w} \quad \Rightarrow \quad G(w) = G(z)\Big|_{z=\frac{1+0.005w}{1-0.005w}}$$

$$\Rightarrow \quad G(w) = \frac{4.9834\times10^{-5}\left(\frac{1+0.005w}{1-0.005w}+0.9967\right)}{\left(\frac{1+0.005w}{1-0.005w}-1\right)\left(\frac{1+0.005w}{1-0.005w}-0.99005\right)} = \frac{1.000036\cdot\left(\frac{w}{121012.12}+1\right)\left(1-\frac{w}{200}\right)}{w\left(\frac{w}{0.999975}+1\right)}$$

3. Plot Bode plot of $G_P(w)$ by substituting $w = jv$:



4. Design for steady state performance
   The transfer function from reference to error is

$$G_E(w) = \frac{1}{1+L(w)} = \frac{1}{1+G(w)C(w)}$$

   at steady state, the error for unit ramp reference should be less than 5%

$$\Rightarrow \quad e_{SS} = \lim_{t\to\infty} e(t) = \lim_{w\to 0} wE(w) = \lim_{w\to 0} wG_E(w)R(w) = \lim_{w\to 0} w\cdot\frac{1}{1+G(w)C(w)}\cdot\frac{1}{w^2}$$

$$= \lim_{w\to 0}\frac{1}{1+G(w)C(w)}\cdot\frac{1}{w} = \frac{1}{1+\left(G(w)\cdot w\right)\big|_{w=0} C(0)} < 0.05$$

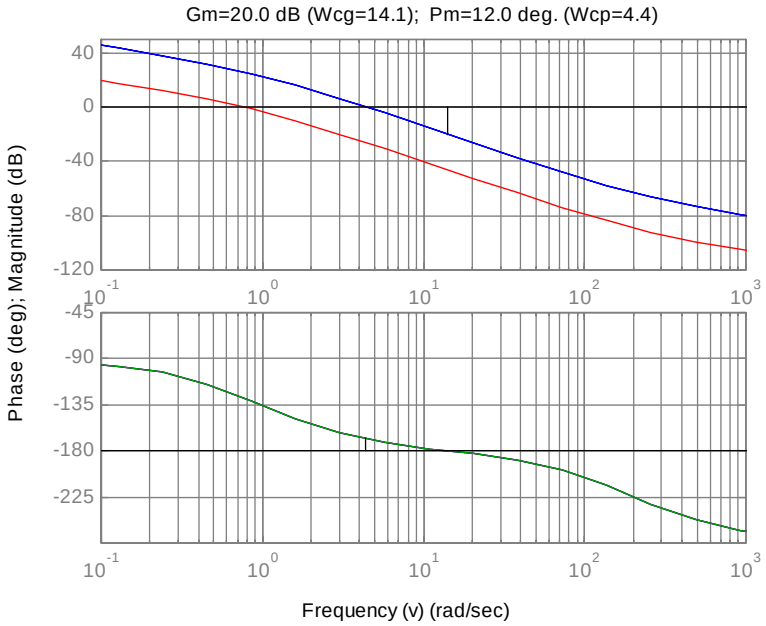
$$\Rightarrow \quad G_C(0) > 19$$

   The steady state gain of the controller should be greater than 19. Let $C(0) = 20$.
   Let the controller in the w-plane be

$$C(w) = C(0)\cdot\overline{C}(w)$$

   where $\overline{C}(w)$ is a unity gain transfer function.

5. Look at the effect of adding additional open lool gain:
   Ploting the open loop transfer function $\overline{G}(w) = G(w)C(0)$



Current gain cross-over frequency is not enough.

6. From specification, the gain cross-over and phase cross-over frequency should be greater than 1 [Hz] = 2π [rad/sec]. In w-plane, the corresponding "cross-over frequencies" should be:

$$v = \frac{2}{T}\tan\left(\frac{\omega \cdot T}{2}\right) = \frac{2}{0.01}\tan\left(\frac{2\pi \cdot 0.01}{2}\right) = 6.28525\,[\text{rad/sec}]$$

7. At v = 6.28525 [rad/sec],

$$\left|\overline{G}(j6.28525)\right| = 0.5 = -6.02\ \text{dB}$$

$$\angle\overline{G}(j6.28525) = -172.757°$$

8. We need to increase phase at 6.28525 [rad/sec] by at least 40^o :
⇒ Use a lead compensator and increase the phase of the open loop transfer function.

Recall that for a lead compensator

$$G_{LEAD}(w) = \frac{1 + a \cdot \Gamma w}{1 + \Gamma w},$$

the maximum phase lead occurs at

$$v_m = \frac{1}{\sqrt{a} \cdot \Gamma} \quad \text{and the phase increase is} \quad \phi_m = \sin^{-1}\left(\frac{a-1}{a+1}\right)$$

Try the following lead compensator:

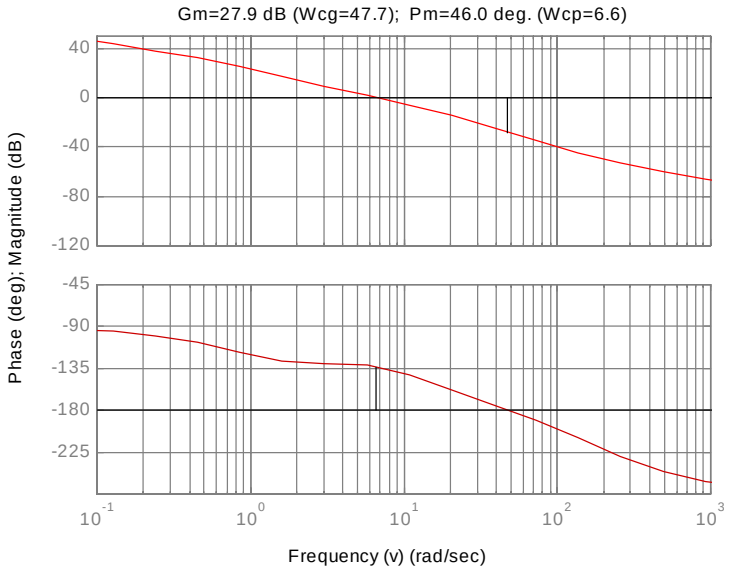
$$G_{LEAD}(w) = \frac{1 + \dfrac{w}{2.93}}{1 + \dfrac{w}{13.48}},$$

that is let $\overline{C}(w) = G_{LEAD}(w)$

9. Check to see if the gain and phase margin specifications are met:
   We will plot the open loop transfer function:

$$L(w) = G(w)\cdot \underbrace{C(0)\overline{C}(w)}_{C(w)}$$

$$= \frac{1.000036\cdot\left(\frac{w}{121012.12}+1\right)\left(1-\frac{w}{200}\right)}{w\left(\frac{w}{0.999975}+1\right)}\cdot 20\cdot\frac{1+\frac{w}{2.93}}{1+\frac{w}{13.48}}$$



10. The lead compensator, $G_{LEAD}(w)$, together with a gain of 20 satified our design specifications, at least in the w-plane. We need to transform the compensator design in w plane back to the z domain using the inverse transform.

    w plane:

$$C(w) = \frac{20\left(1+\frac{w}{2.93}\right)}{\left(1+\frac{w}{13.48}\right)} \quad \text{where} \quad w = \frac{2}{T}\frac{z-1}{z+1}$$

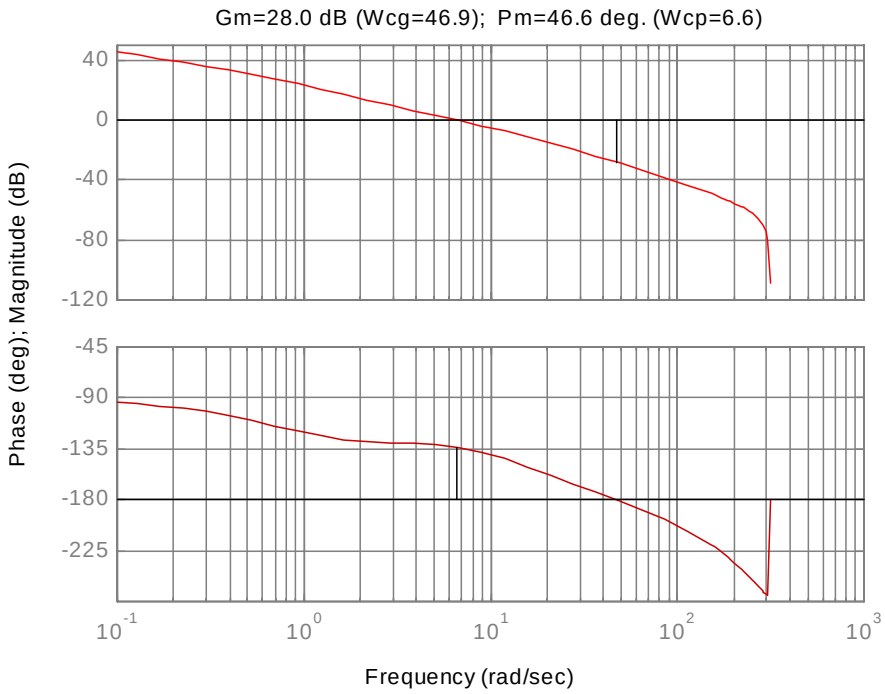
    z domain:

$$C(z) = C(w)\Big|_{w=\frac{2}{T}\frac{z-1}{z+1}} = \left.\frac{20\left(1+\frac{w}{2.93}\right)}{\left(1+\frac{w}{13.48}\right)}\right|_{w=\frac{2}{0.01}\frac{z-1}{z+1}} = 87.4672\frac{z-0.9711}{z-0.8737}$$

$$\Rightarrow \quad C(z) = 87.4672\frac{z-0.9711}{z-0.8737}$$

11. Double check our design by checking the GM and PM of the discrete time open loop transfer function:

$$L(z) = G(z)\cdot C(z) = \frac{4.9834\times 10^{-5}(z+0.9967)}{(z-1)(z-0.99005)}\cdot 87.4672\cdot\frac{z-0.9711}{z-0.8737}$$



Also need to check the steady state error for unit ramp input.