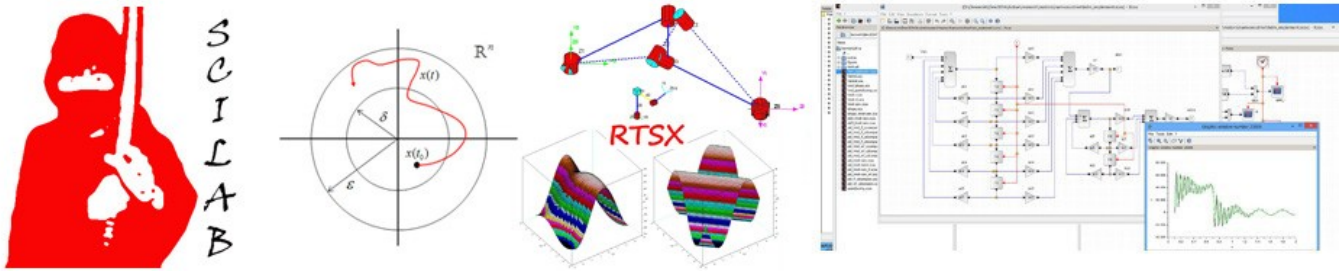


# Scilab Ninja

Control Engineering with Scilab



Tweet

Like

3

G+1

0

Share

Share

3

## Module 7: Continuous to Discrete Conversion Methods

This article is contained in *Scilab Control Engineering Basics* study module, which is used as course material for International Undergraduate Program in Electrical-Mechanical Manufacturing Engineering, Department of Mechanical Engineering, Kasetsart University.

### Module Key Study Points

- Understand 3 methods to convert a continuous-time transfer function to discrete-time
- Understand sampling process and aliasing problem
- Discuss stability region mappings of the 3 conversion methods

In module 6, we study the basics of discrete-time control systems in a nutshell, starting from the process of sampling an analog signal, developing a difference equation, to converting to a discrete transfer function. Stability determination in the Z-domain is also discussed. Typically, control analysis and design is performed in continuous-time, due to the well-established theories and approaches. After a controller that meets the specifications is achieved, it is then converted to discrete-time for implementation. The purpose of this module, therefore, is to elaborate more on the conversion process.

The concept of transformation between two domains is simple. We want to find some relationship between the s and z variables. Consider a system consisting of a single integrator

$$\dot{u}(t) = e(t) \quad (1)$$

with continuous-time transfer function

$$C(s) = \frac{U(s)}{E(s)} = \frac{1}{s} \quad (2)$$

The solution for this system is

$$u(t) = u(t_0) + \int_{t_0}^t e(\tau) d\tau \quad (3)$$

with value at sampling instants

$$u((k+1)T) = u(kT) + \int_{kT}^{(k+1)T} e(\tau) d\tau \quad (4)$$

As shown in Figure 1, the integral term in (4) can be approximated by 3 methods, namely *forward difference*, *backward difference*, and *bilinear transformation* (also known as *Tustin* or *trapezoidal approximation*). They can be described as

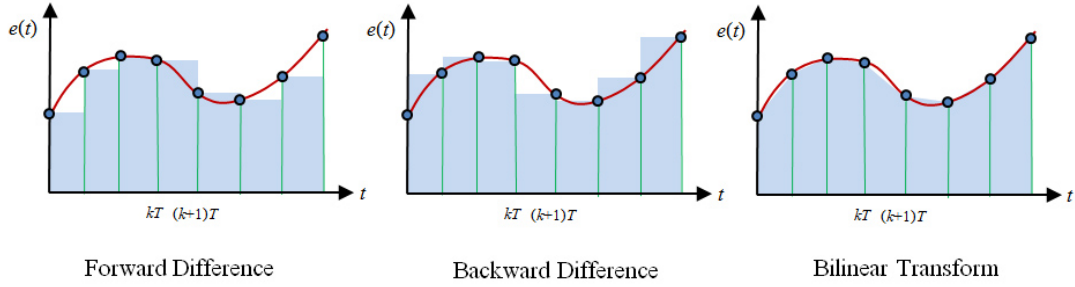


Figure 1 integral approximations by 3 methods

*Forward Difference:*

$$u(k+1) \approx u(k) + e(k)T \quad (5)$$

*Backward Difference:*

$$u(k+1) \approx u(k) + e(k+1)T \quad (6)$$

*Bilinear Transformation:*

$$u(k+1) \approx u(k) + \frac{e(k+1) + e(k)}{2}T \quad (7)$$

Transforming (5)–(7) to discrete-time transfer function  $C(z) = U(z)/E(z)$  results in

*Forward Difference:*

$$C(z) = \frac{T}{z-1} = \frac{Tz^{-1}}{1-z^{-1}} \quad (8)$$

*Backward Difference:*

$$C(z) = \frac{Tz}{z-1} = \frac{T}{1-z^{-1}} \quad (9)$$

*Bilinear Transformation:*

$$C(z) = \frac{T}{2} \frac{(z+1)}{(z-1)} = \frac{T}{2} \frac{(1+z^{-1})}{(1-z^{-1})} \quad (10)$$

Comparing the discrete-time transfer function from each method to the continuous-time counterpart in (2), we derive the following relationships

*Forward Difference:*

$$s \rightarrow \frac{z - 1}{T} \quad (11)$$

*Backward Difference:*

$$s \rightarrow \frac{z - 1}{Tz} \quad (12)$$

*Bilinear Transformation:*

$$s \rightarrow \frac{2(z - 1)}{T(z + 1)} \quad (13)$$

i.e., to convert a continuous-time transfer function using any of the 3 methods, we substitute the Laplace variable  $s$  with the term on the right for the corresponding relationship. This is best illustrated by an example.

**Ex. 1** Consider a continuous-time controller

$$C(s) = 20000 \frac{(s + 0.01)}{(s + 100)} \quad (14)$$

This can be converted to the following discrete-time transfer functions

*Forward Difference:*

$$C(z) = 20000 \frac{(\frac{z-1}{T} + 0.01)}{(\frac{z-1}{T} + 100)} = 20000 \frac{z - 1 + 0.01T}{z - 1 + 100T} \quad (15)$$

*Backward Difference:*

$$C(z) = 20000 \frac{(\frac{z-1}{Tz} + 0.01)}{(\frac{z-1}{Tz} + 100)} = 20000 \frac{(1 + 0.01T)z - 1}{(1 + 100T)z - 1} \quad (16)$$

*Bilinear Transformation:*

$$C(z) = 20000 \frac{(\frac{2(z-1)}{T(z+1)} + 0.01)}{(\frac{2(z-1)}{T(z+1)} + 100)} = 20000 \frac{(2 + 0.01T)z - 2 + 0.01T}{(2 + 0.01T)z - 2 + 0.01T} \quad (17)$$

One way to investigate accuracy of the three conversion methods is to compare their frequency responses. A tight match of Bode plots in Figure 2 results from conversion with sampling time  $T=0.001$  sec.

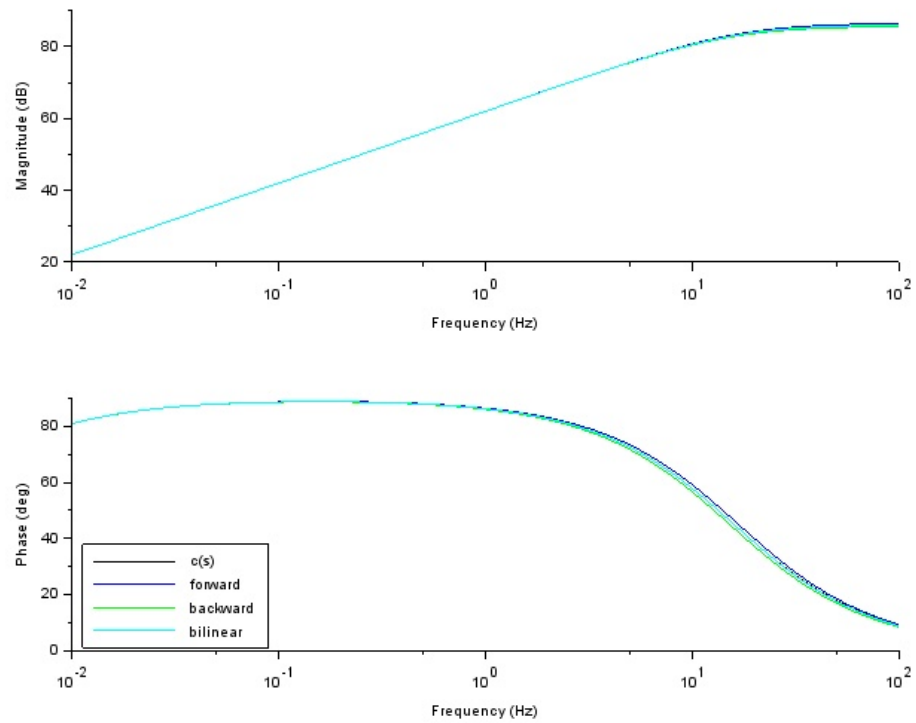


Figure 2 A close match in Bode plots with  $T = 0.001$  sec

As shown in Figure 3, discrepancy is more pronounced with longer sampling period of  $T = 0.01$  sec, especially in high frequency region.

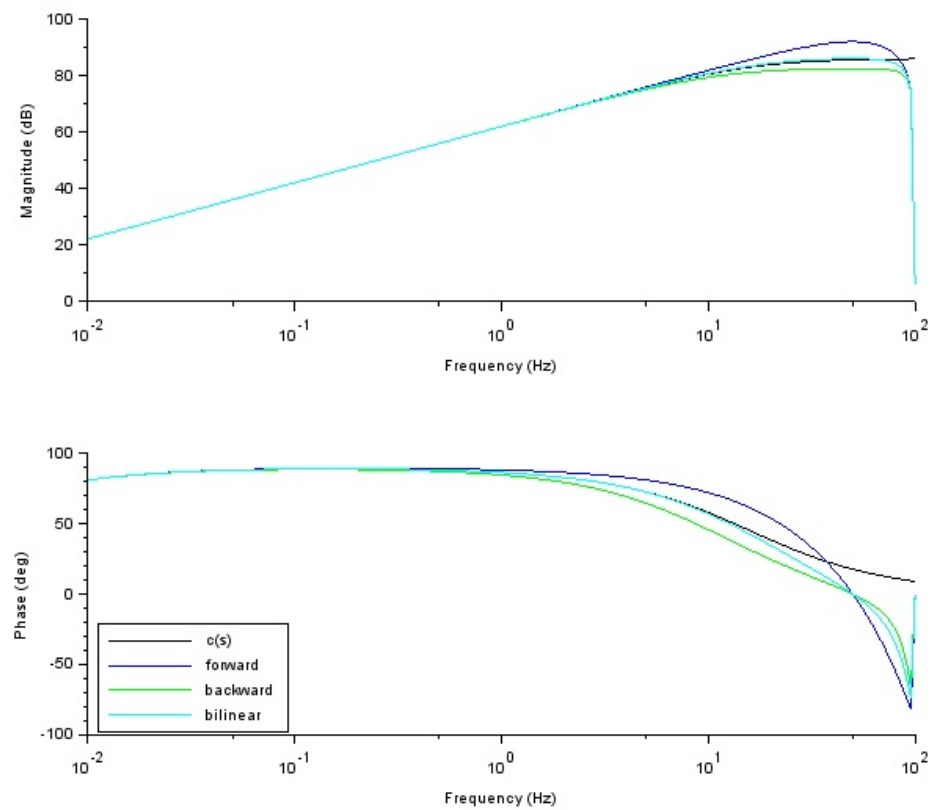


Figure 3 Discrepancy in Bode plots with T = 0.01 sec

**Ex. 2** Consider a continuous-time PID controller with derivative term replaced by a filter transfer function

$$C(s) = K_p + \frac{K_i}{s} + \frac{NK_d}{1 + \frac{N}{s}} \quad (18)$$

Given a sampling period T, the integral term  $K_i/s$  can be represented in discrete-form by

*Forward Difference:*

$$\frac{K_i T}{z - 1} \quad (19)$$

*Backward Difference:*

$$\frac{K_i T z}{z - 1} \quad (20)$$

*Bilinear Transformation:*

$$\frac{K_i T}{2} \frac{z + 1}{z - 1} \quad (21)$$

Similarly, the derivative term in (18) can be discretized as

*Forward Difference:*

$$\frac{N}{1 + \frac{NT}{z-1}} = \frac{N(z-1)}{z-1+NT} \quad (22)$$

*Backward Difference:*

$$\frac{N}{1 + \frac{NTz}{z-1}} = \frac{N(z-1)}{(1+NT)z-1} \quad (23)$$

*Bilinear Transformation:*

$$\frac{N}{1 + \frac{NT}{2} \frac{z+1}{z-1}} = \frac{N(z-1)}{(1 + \frac{NT}{2})z + \frac{NT}{2} - 1} \quad (24)$$

Hence, the resulting discrete-time PID controllers are represented by

*Forward Difference:*

$$C(z) = K_p + \frac{K_i T}{z - 1} + \frac{K_d N(z - 1)}{z - 1 + NT} \quad (25)$$

Backward Difference:

$$C(z) = K_p + \frac{K_i T z}{z - 1} + \frac{K_d N(z - 1)}{(1 + NT)z - 1} \quad (26)$$

Bilinear Transformation:

$$C(z) = K_p + \frac{K_i T}{2} \frac{(z + 1)}{(z - 1)} + \frac{K_d N(z - 1)}{(1 + NT/2)z + NT/2 - 1} \quad (27)$$

We want to simulate the step responses of these 3 controllers compared to the continuous-time (18). To aid in the process, an Xcos model [dpidsim.zcos](#) shown in Figure 4 is constructed. The upper loop has the continuous-time PID controller, and the lower loop has the discrete-time version, which can be set up to each of the 3 choices in (25) – (27) by a script file [dpidsim\\_setup.sce](#), containing the following Scilab code.

```
// dpidsim_setup.sce
// setup file for dpidsim_xx.zcos

kp = 200;
ki = 240;
kd = 25;
T = 0.01;
N = 20;

z = poly(0, 'z');

// Forward Difference Method
Inumz = T;
Idenz = z-1;
Dnumz = N*(z - 1);
Ddenz = (z-1+N*T);
//
// Backward Difference Method
//Inumz = T*z;
//Idenz = z-1;
//Dnumz = N*(z - 1);
//Ddenz = (1+N*T)*z - 1;
//
// Bilinear Transform Method
//Inumz = T*(z+1);
//Idenz = 2*(z-1);
//Dnumz = N*(z - 1);
//Ddenz = (1+0.5*N*T)*z + 0.5*N*T -1;
//
```

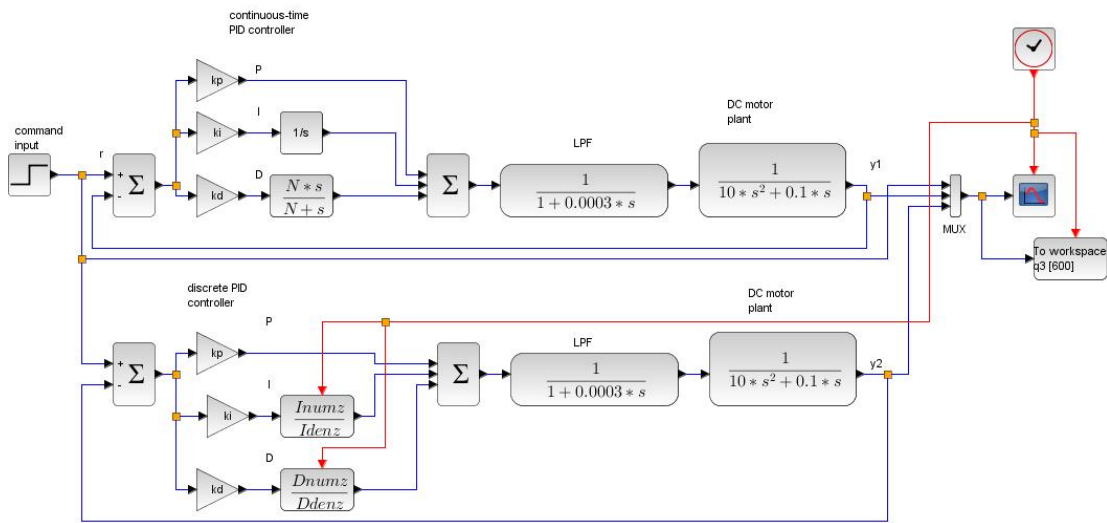


Figure 4 `dpidsim.zcos` model for step response comparison

Uncomment each of the 3 choices to use a particular conversion method and run the script before each simulation. Change the output variable name in `To Workspace` block to `q1`, `q2`, `q3`, in order with the 3 methods. Then use this script `compare3.sce` to plot the step response comparison.

The first comparison is performed with  $T = 0.01$  sec. The result is shown in Figure 5. The step responses from all 3 methods match well with the one from continuous-time PID.

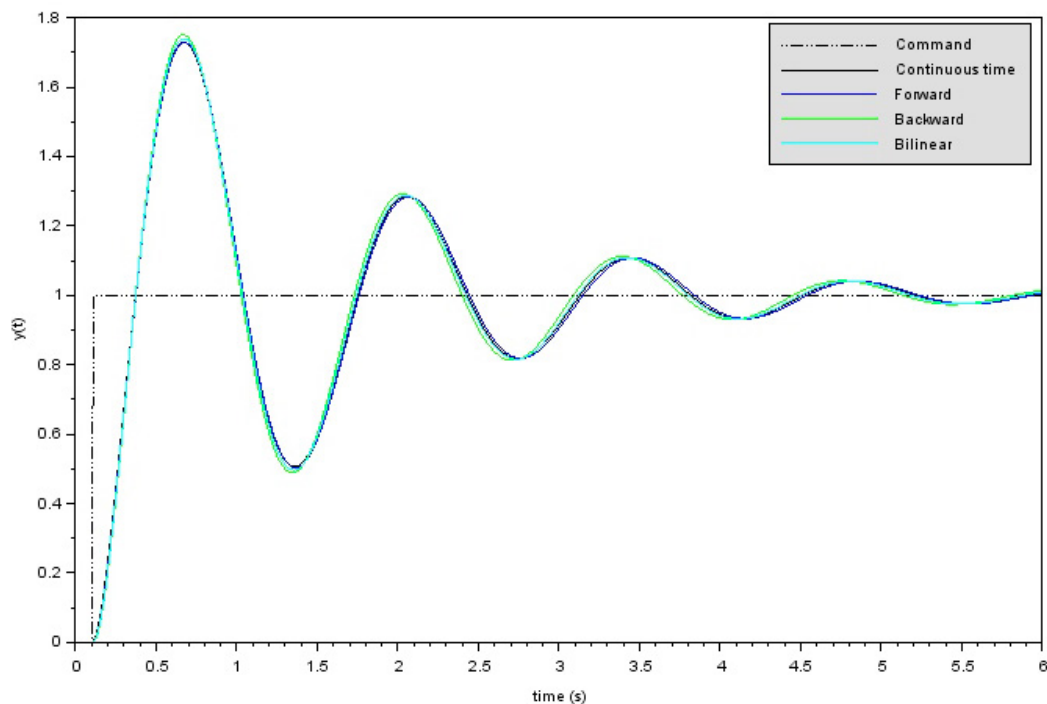


Figure 5 comparison result with  $T = 0.01$  sec

When the sampling period is increased, approximation errors result in the 3 conversion methods, as shown in Figure 6 and 7 with  $T = 0.05$  and  $0.08$ , respectively. It is interesting to note that the discrete PID from forward difference method (blue) seems to yield a pretty close match of step responses to the continuous-time, while the poorest is from backward difference

(green).

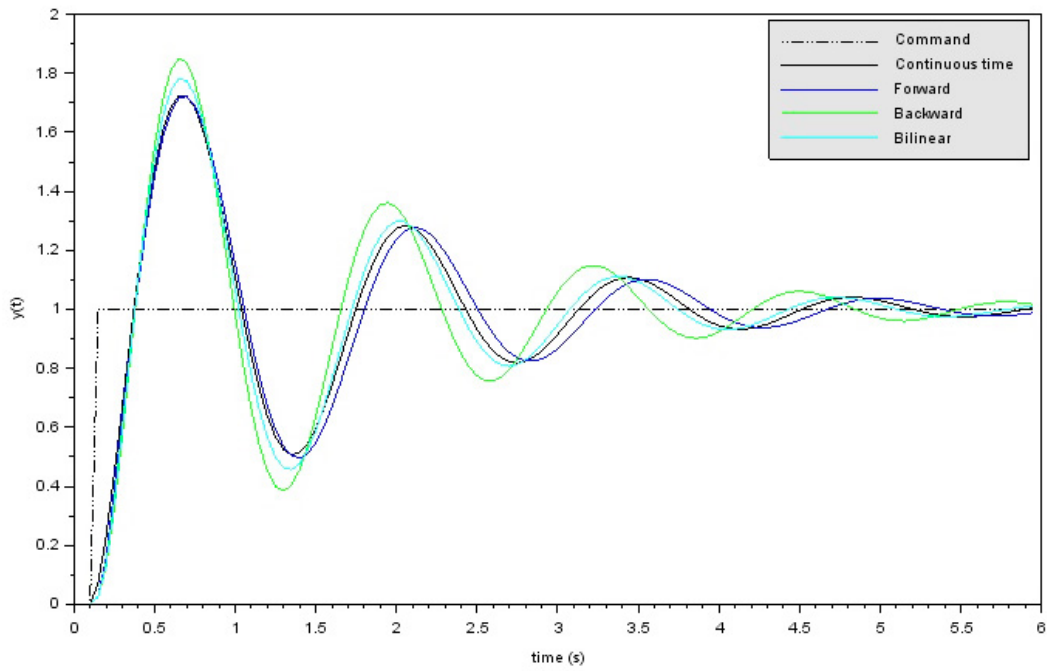


Figure 6 comparison result with  $T = 0.05$  sec

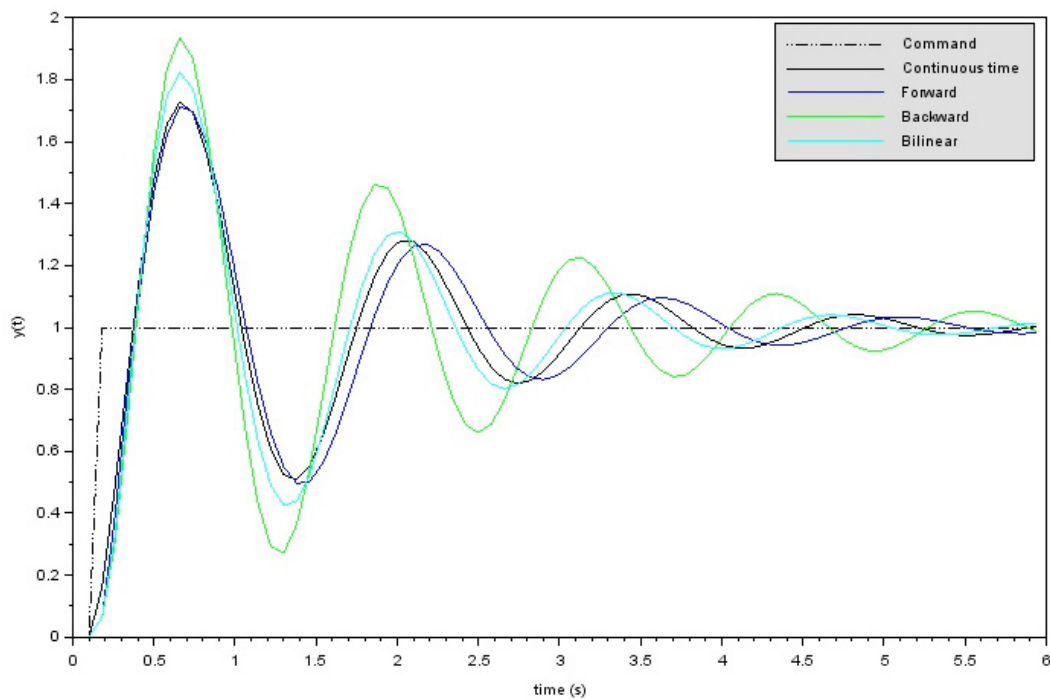


Figure 7 comparison result with  $T = 0.08$  sec

So, from this simulation one may select forward difference as preferred conversion method. The controller expression is less complicated than that from bilinear transform, and it tends to perform better than the backward difference candidate in terms of time domain response. But wait, there is one important issue to be determined.



Perform the simulation again with  $T = 0.05$ . Controller parameters remain the same, except only the filter coefficient  $N$  is changed from 20 to 50. The comparison result is shown in Figure 8. The response from discrete controller converted by forward difference is the only one that goes unstable. What has happened?

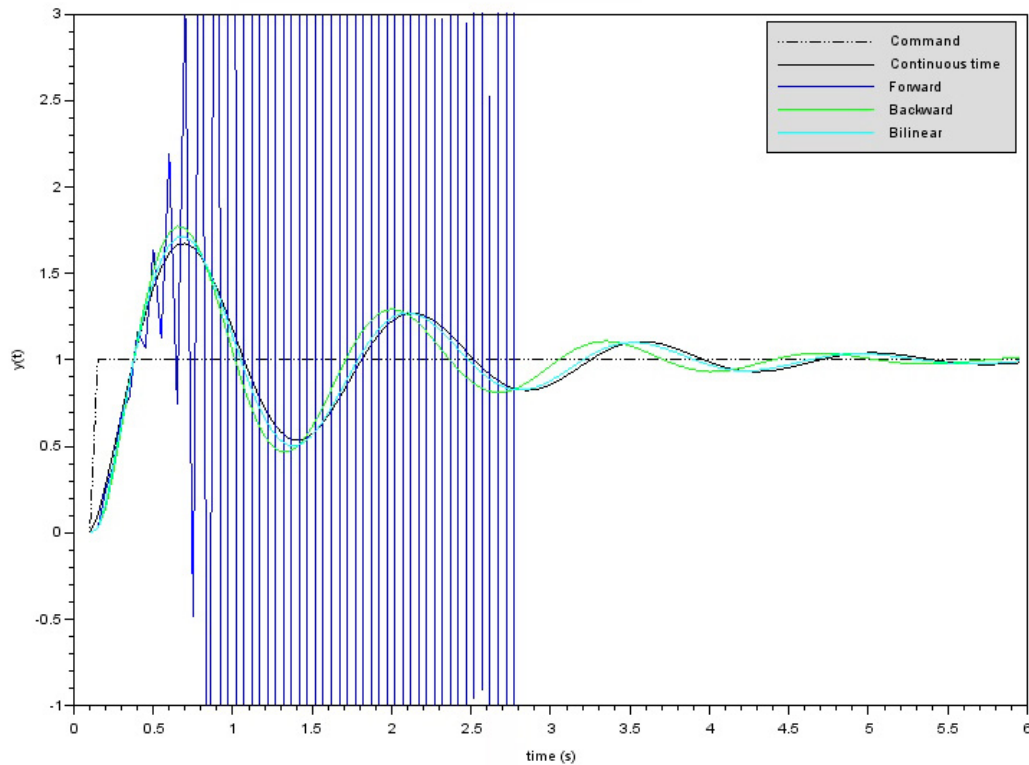


Figure 8 comparison result with  $N$  changed to 50

Take a closer look at the controller itself. First create the continuous-time PID controller (18)

```
-->s=poly(0,'s')
-->Cs = kp + ki/s + kd*N*s/(s+N);
-->Cs = syslin('c',Cs)
Cs =

      12000 + 10240s + 1450s
      -----
              2
      50s + s
```

compute the poles

```
-->roots(Cs.den)
ans =
    - 50.
     0
```

to verify that it is a stable transfer function, except the pole at origin from the integrator term.

Now, form the discrete version from forward difference method

```
-->z=poly(0,'z');
```

```

-->Inumz = T;
-->Idenz = z-1;
-->Dnumz = N*(z - 1);
-->Ddenz = (z-1+N*T);
-->Cz = kp + ki*Inumz/Idenz + kd*Dnumz/Ddenz;
-->Cz = syslin('d',Cz)
Cz =
          2
    968 - 2388z + 1450z
    -----
          2
    - 1.5 + 0.5z + z

```

and check the radii of poles from the origin

```

-->abs(roots(Cz.den))
ans =
    1.5
    1.

```

to see that this discrete controller is unstable, since it has a pole outside unit circle. So, this example shows that a stable continuous-time transfer function can be converted to an unstable discrete-time version, and hence the unbounded response in Figure 8.

Stability can be considered the most important property for a feedback system. Not only an unstable system is rendered useless, it could also be disastrous. So it is helpful to analyze how stability region is mapped between two domains. We elaborate on this for the forward difference method.

The stability criteria in continuous-time domain is  $Re(s) < 0$ . With forward difference method, this translates to

$$Re(s) = Re\left(\frac{z-1}{T}\right) < 0 \quad (28)$$

Since the sampling period  $T$  is always positive, so  $Re(z-1) < 0$ . Let  $z = \sigma + j\omega$ , we have

$$Re(z-1) = Re(\sigma + j\omega - 1) = \sigma - 1 < 0 \Rightarrow \sigma < 1 \quad (29)$$

From this result, we conclude that, using forward difference conversion method, the stability region in continuous-time domain is mapped to the area to the left of 1 in the  $Z$  plane. This explains the possibility that stable continuous-time transfer functions can be approximated by unstable discrete transfer functions.

It is left to the reader to verify that the left half plane of  $S$  plane is mapped by each of the 3 conversion methods to the  $Z$  plane as shown in Figure 9. From this diagram, it is obvious that bilinear transformation is the best choice in terms of stability preservation, according to the fact that stability region in the  $Z$  plane is inside the unit circle. In the backward difference case, some unstable continuous-time systems may be mapped to stable discrete-time systems. This seems to pose less problem, in the general sense that analysis and design is often done in continuous-time domain and the resulting controller is converted to discrete-time at implementation phase.

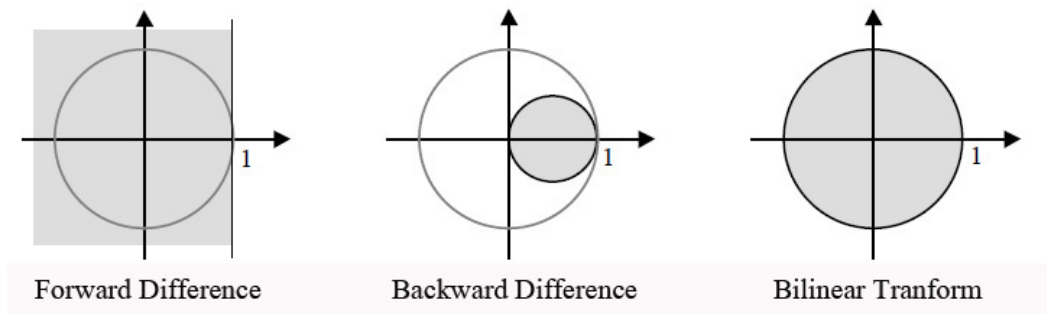


Figure 9 Stability region mapping of the 3 conversion methods

Nevertheless, it can be concluded from this study that bilinear transformation is a preferred choice in most cases. In certain applications such as filter design, a technique called frequency prewarping can be added to yield precise match at a particular frequency.

### Supplement

- [module7.zip](#) : all Scilab and Xcos files used in this module