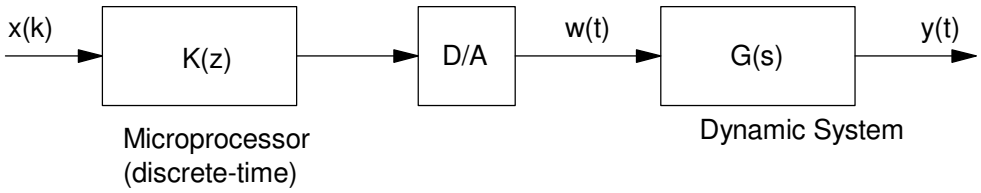
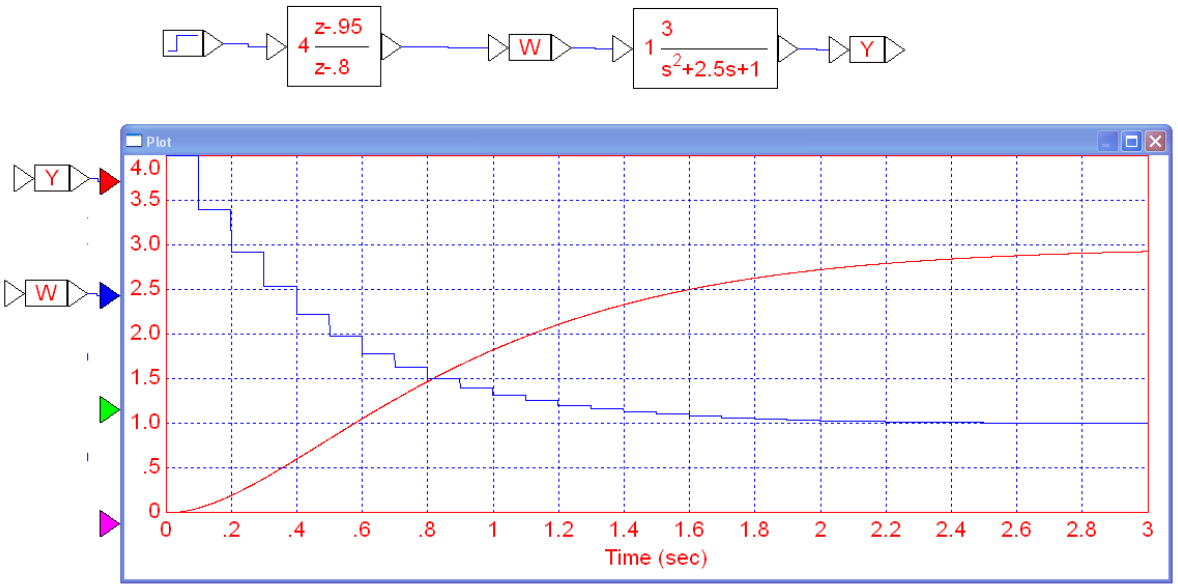
Converting G(s) to G(z)

Finaly, let's look at the case where you have a hybrid system: part of the system is a discrete-time difference equation, part of the system is a continuous time differential eqaution. This situation is often encountered with microprocessor-controlled systems where the microprocessor outputs a voltage at discrete times (the sampling rate), which then drive an analog system such as a DC motor.



Hybrid System: Part Discrete-Time, Part Continuous Time

What the output should look like is something like this:



VisSim simulation of a hybrid system: a discrete-time system with a sampling rate of 100ms driving an analog system
Note that w(t) changes at discrete times (the sampling rate of K(z) whereas y(t) is analog

Such a system is difficult to analyze:

- It it were all continuous-time, we could use LaPlace transforms
- If it were all discrete-time, we could use z-transforms.

As it is, we don't really have the mathematical tools to analyze such a system.

Since this is a difficult problem to solve, we use an engineering-type solution: change the problem.

- Convert K(z) to its continuous-time equivalent, K(s), then analyze using LaPlace transforms, or
- Convert G(s) to it's discrete-time equivalent, G(z), then analyze using z-transforms.

Either way, we need to convert between the s-plane and the z-plane.

s to z Conversion Method #1: Substitution

One way to between the s-plane and the z-plane is use the approximations

$$\frac{1}{s} \approx T\left(\frac{z}{z-1}\right) \qquad \text{Euler Integration}$$

or

$$\frac{1}{s} \approx \frac{T}{2}\left(\frac{z+1}{z-1}\right) \qquad \text{Bilinear Integration}$$

These come from two types of numerical integration. The former (Euler integration) approximates the area under a curve (the integral) as a bunch of rectanges with

Area = Width * Height

If

$$Y = \frac{1}{s}X$$

then for Euler integration

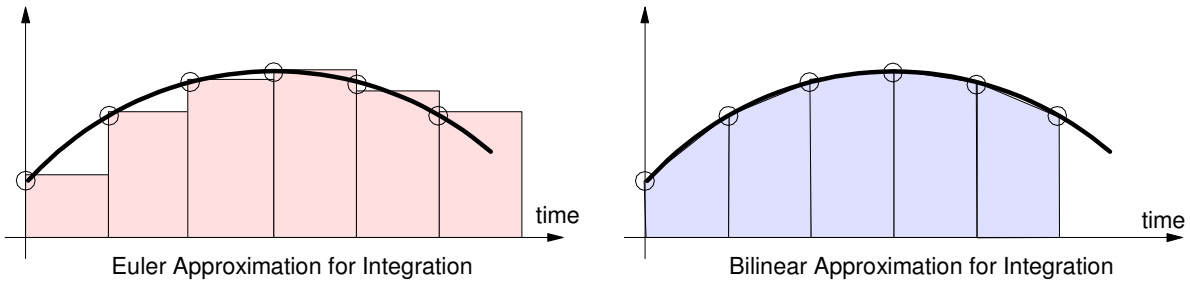
$$y(k) \approx y(k-1) + T \cdot x(k)$$

$$Y = \left(\frac{Tz}{z-1}\right)X$$

For bilinear integration

$$y(k) \approx y(k-1) + T\left(\frac{x(k)+x(k-1)}{2}\right)$$

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



Euler and Bilinear integration approximate the integral of a function as the are of rectangles (Euler) or trapezoids (bilinear)

This method works and is used, but the algebra can get rather intensive. Plus, it gives you no intuition as to what is happening.

Method 2: Mapping s to z as $z = e^{sT}$

My preferred method uses a 1-to-1 mapping from the s-plane to the z-plane.

The basic assumption behind the LaPlace transform is that all funcitons are in the form of

$$y(t) = e^{st}$$

If t is sampled every T seconds, then

$$t = kT$$

$$y(kT) = e^{skT}$$

or

$$y(k) = (e^{sT})^k$$

which is the basic assumption behind the z-transform

$$y(k) = z^k$$

This gives the mapping between the s-plane and the z-plane

$$z = e^{sT}$$

This leads to the following method:

i) Convert every pole and zero in the s-plane to a pole and zero in the z-plane as

$$z = e^{sT}$$

ii) Add a gain to match the gain of G(s) and G(z) at some frequency - typically DC

iii) (optional) Add zeros at z = 0 to match the phase at a frequency close to zero

- or -

Add n zeros at z = 0 to match the delay in the system.

Example: Find the discrete-time equivalent of G(s) with a sampling rate of T = 0.1

$$G(s) = \left(\frac{100}{(s+1)(s+3)(s+10)} \right)$$

In Matlab

```
s = [-1;-3;-10]

  - 1.
  - 3.
  - 10.
```

```
T = 0.1

    0.1

z = exp(s*T)

    0.9048374
    0.7408182
    0.3678794
```

meaning

$$G(z) = \left(\frac{k}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)$$

To find k, match the gain at DC

$$\left(\frac{100}{(s+1)(s+3)(s+10)} \right)_{s=0} = 3.3333$$

$$\left(\frac{k}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)_{z=1} = 3.33333$$

```
k = prod(1-z) * 3.3333

    0.0519691
```

meaning

$$G(z) = \left(\frac{0.051969}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)$$

To find how many zeros belong at z=0,

a) There is too much delay with this system. Add zeros until the step response is "close"

b) Match the phase at some frequency, such as s = j1

$$\left(\frac{100}{(s+1)(s+3)(s+10)} \right)_{s=j1} = 2.2249 \angle -69.14^0$$

G(z) at s = j1 is:

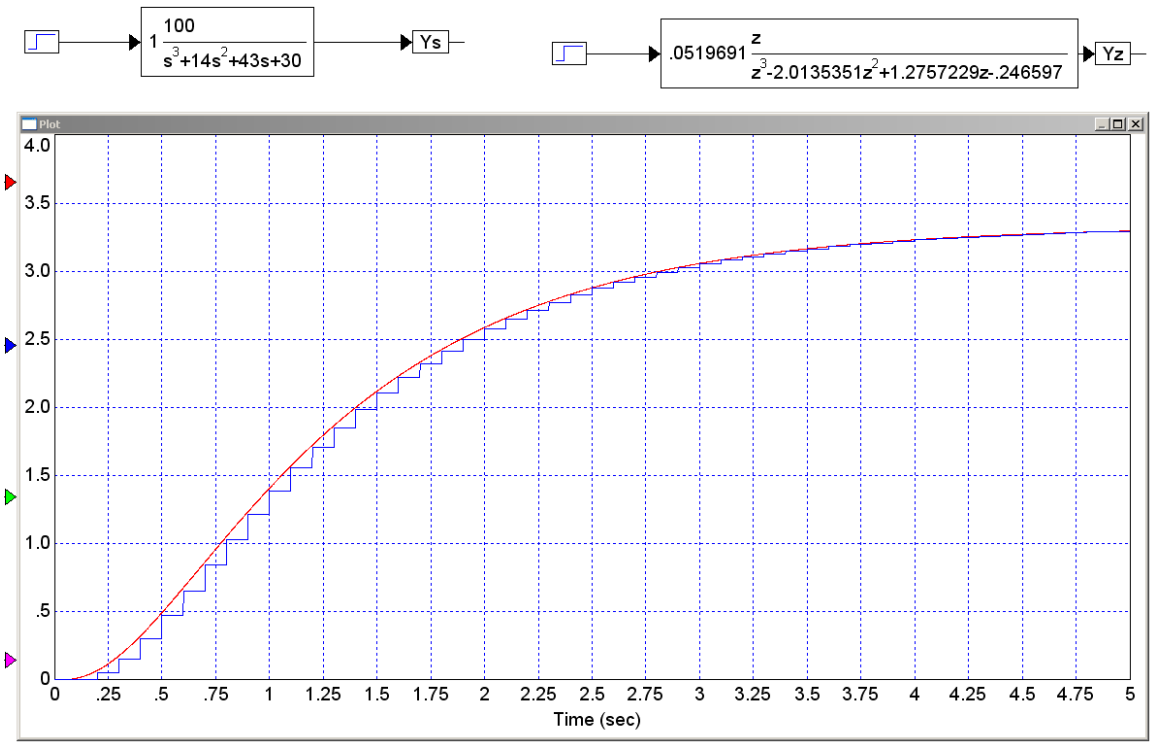
$$\left(\frac{0.051969}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)_{s=j} = 2.2276 \angle -78.04^0$$

so the phase is off by 8.90 degrees. Each zero at z -= 0 adds 5.73 degrees

$$e^{sT} = e^{(j1)(0.1)} = 1 \angle 5.73^0$$

meaning you need to add 1.55 zeros. I don't know how to add a fraction of a zero, so add one zero at z=0, resulting in

$$G(z) \approx \left(\frac{0.051969 \cdot z}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)$$



VisSim Comparison of G(s) to G(z).
Note that G(z) has too much delay. This is due to rounding down the number of zeros at z=0 from 1.55 zeros to 1.00 zero.

In Matlab: Input the system G(s)

```
>> Gs = zpk([],[-1,-3,-10],100)

        100
-----------------
(s+1) (s+3) (s+10)
```

Input G(z). For now, assume the numerator is 1

```
>> T = 0.1;
>> Gz = zpk([],[exp(-1*T),exp(-3*T),exp(-10*T)],1,T)

                1
-------------------------------
(z-0.9048) (z-0.7408) (z-0.3679)
```

Add a gain, k, so that the DC gain matches up

```
>> DCs = evalfr(Gs,0)

    3.3333

>> DCz = evalfr(Gz,1)

   64.1401

>> k = DCs / DCz

    0.0520
```

So, G(z) is.

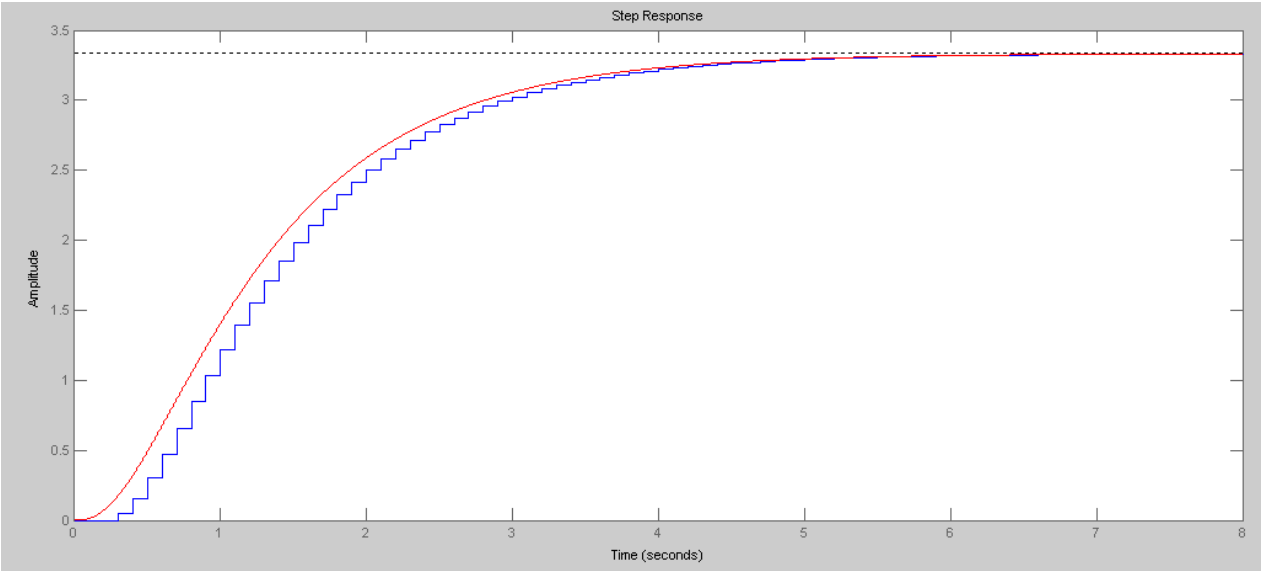
```
>> Gz = zpk([],[exp(-1*T),exp(-3*T),exp(-10*T)],k,T)

            0.05197
-------------------------------
(z-0.9048) (z-0.7408) (z-0.3679)

Sampling time (seconds): 0.1
```

Checking the answer: Plot the step response of G(z) and G(s)

```
>> step(Gz)
>> hold on
>> t = [0:0.001:8]';
>> ys = step(Gs,t);
>> plot(t,ys,'r');
```



Note that there is too much delay in G(z). Adding one or two zeros at z=0 will help

Example 2: Complex Poles

This also works with complex poles and zeros

Determine a discrete-time system, G(z), which behaves approximately the same as G(s)

$$Y = \left(\frac{100(s+j5)(s-j5)}{(s+1+j4)(s+1-j4)(s+2)} \right)$$

```
-->sn = [j*5,-j*5]'
-->T = 0.1
-->zn = exp(ns*T)

    0.8775826 - 0.4794255i
    0.8775826 + 0.4794255i

-->poly(nz)

    1.  - 1.7551651    1.

-->sd = [-1+j*4,-1-j*4,-20]'

  - 1. - 4.i
  - 1. + 4.i
  - 20.

-->zd = exp(sd*T)

    0.8334105 - 0.3523603i
    0.8334105 + 0.3523603i
    0.1353353

-->poly(zd)

    1.  - 1.8021562    1.0443104  - 0.1108032
```

meaning

$$G(z) = k\left(\frac{z^2-1.755z+1.}{z^3-1.802z^2+1.044z-0.110} \right)$$

To find k, match the DC gain:

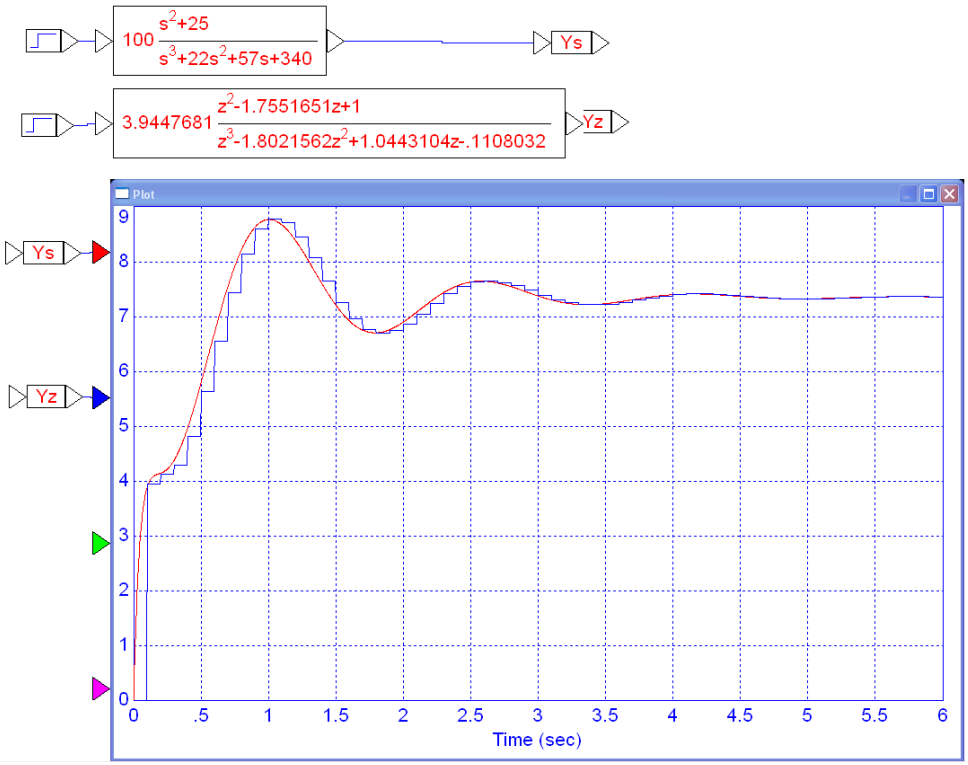
```
-->DC = 100*25/340

    7.3529412

-->k = DC*prod(1-zd)/prod(1-zn)

    3.9447681
```

$$G(z) = 3.944\left(\frac{z^2-1.755z+1.}{z^3-1.802z^2+1.044z-0.110} \right)$$



Step Response of G(s) and G(z)

In Matlab: Input the system G(s)

```
>> z1 = j*5;
>> z2 = -j*5;
>> p1 = -1+j*4;
>> p2 = -1-j*4;
>> p3 = -2;
>> Gs = zpk([z1,z2],[p1,p2,p3],100)

   100 (s^2 + 25)
--------------------
(s+2) (s^2 + 2s + 17)
```

Now input G(z). Convert the poles and zeros to the z-plane as e^{sT}.

```
>> T = 0.1;
>> Gz = zpk([exp(z1*T),exp(z2*T)],[exp(p1*T),exp(p2*T),exp(p3*T)],1,T)

         (z^2 - 1.755z + 1)
---------------------------------
(z-0.8187) (z^2 - 1.667z + 0.8187)

Sampling time (seconds): 0.1
```

Add a gain, k, to make the DC gains match up:

```
>> DCs = evalfr(Gs,0)

   73.5294

>> DCz = evalfr(Gz,1)

    8.8913

>> k = DCs / DCz

    8.2699
```

So, the discrete-time model for G(s) is....

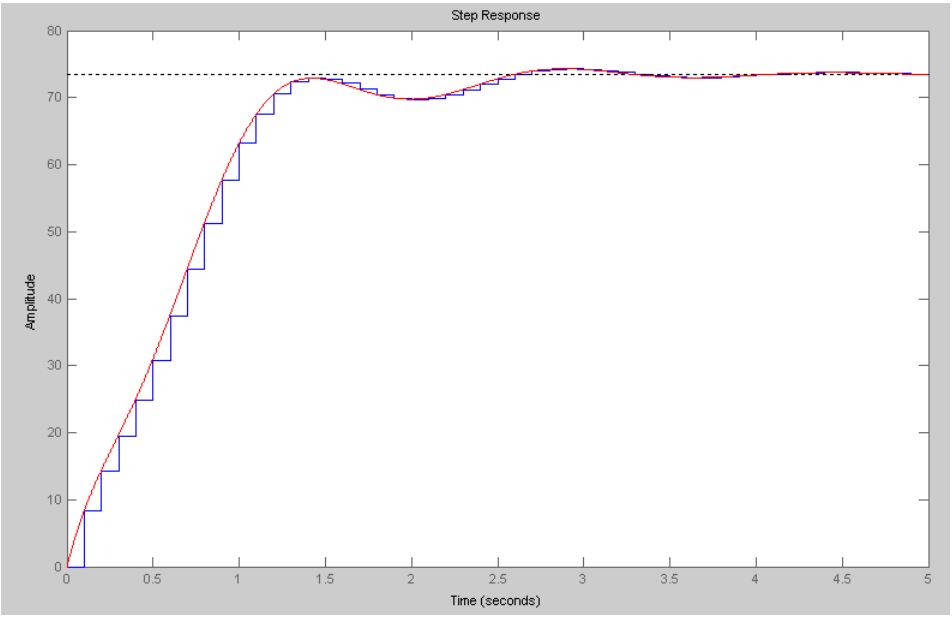
```
>> Gz = zpk([exp(z1*T),exp(z2*T)],[exp(p1*T),exp(p2*T),exp(p3*T)],k,T)

    8.2699 (z^2 - 1.755z + 1)
---------------------------------
(z-0.8187) (z^2 - 1.667z + 0.8187)

Sampling time (seconds): 0.1
```

Check the result by plotting the step response of G(s) and G(z) on the same graph:

```
>> step(Gz)
>> hold on
>> t = [0:0.001:5]';
>> ys = step(Gs,t);
>> plot(t,ys,'r');
```



Changing the Sampling Rate:

Note that the conversion from s to z depends upon the sampling rate. If you change the sampling rate, the filter G(z) changes completely.

For example,

$$G(s) = \left(\frac{100}{(s+1)(s+3)(s+10)} \right)$$

becomes the following with a sampling rate of T = 0.1:

$$G(z) \approx \left(\frac{0.051969 \cdot z}{(z-0.9048)(z-0.7408)(z-0.3687)} \right)$$

If you change the sampling rate to T = 0.01 (10ms), then

$$G(z) \approx \left(\frac{0.00009393 \cdot z}{(z-0.9900)(z-0.9704)(z-0.9048)} \right)$$

Changing the sampling rate is a big deal: it completely changes the system G(z) and likewise will require a complete redesign of the compensator, K(z).