z-Transforms

NDSU ECE 376

Lecture #27

Inst: Jake Glower

Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

Introduction:

Anything you can do in software you can do in hardware, and visa versa.

In Circuits II and Electronics II, we deisgn filtes in the s-plane. These include

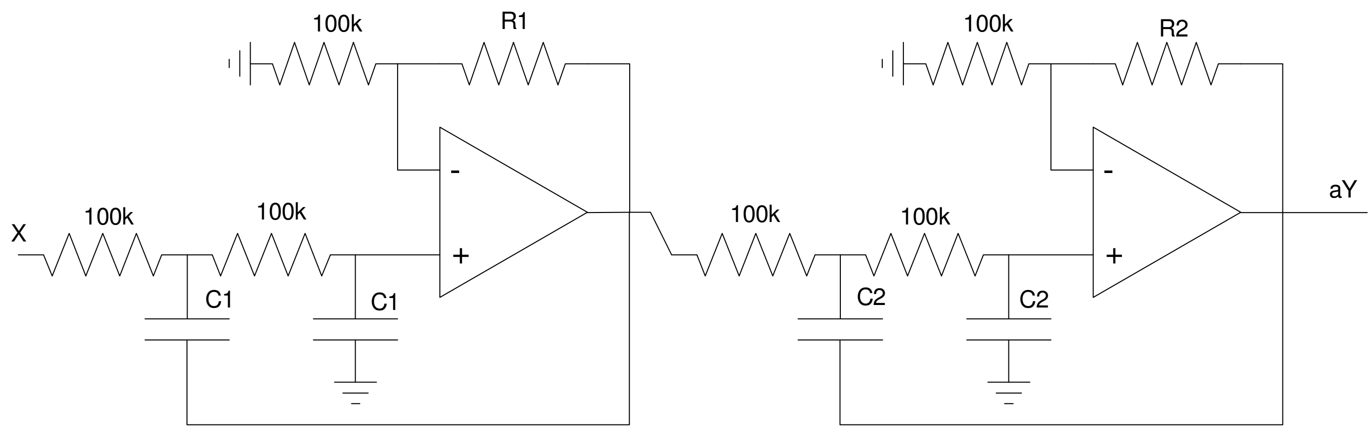
- RC low-pass filters,
- RLC band-pass filters, and
- Active filters

To describe these filters, differential equations are used. This results in transfer functions in 's' where

sY

means

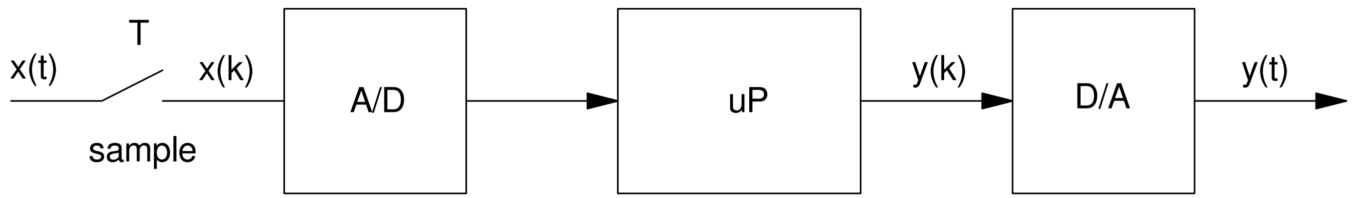
the derivative of y(t)



Digital Filters

With a microprocessor, you

- Samples the error every T seconds
- The sampled signal is sent through an analog to digital (A/D) converter
- A program on the microcontroller computes the output (software), and then
- The output of the microcontroller is sent to a digital to analog (D/A) converter, producing y(t)



Advantages of Digital Filters

- Code that ran yesterday should also run today.
- DC offsetts don't exist in software. Zero plus zero is zero.
- If you want a more complex controller, you just add lines of code.
- If you want to change the controller, you just download a new program.

Problem:

- LaPlace transforms don't work well when describing software

Sample Code

```
while(1) {

    k = k + 1;                       // k = iteration number

    x1 = x0;                         // x(k-1)
    x0 = A2D_Read(0);                // read in x(k) from the A/D

    y1 = y0;                         // y(k-1)
    y0 = y1 + 0.2*(x0 - 0.9*x1);

    Wait_10ms();

    }
```

This requires a difference equation

$$y(k) = y(k-1) + 0.2(x(k) - 0.9x(k-1))$$

LaPlace Transforms

Assume

$$y = e^{st}$$

then

$$\frac{dy}{dt} = s \cdot e^{st} = sY$$

This turns differential equations into transfer functions in 's'

$$Y = \left(\frac{8s+3}{s^2+7s+12}\right) X$$

means

$$\frac{d^2y}{dt^2} + 7\frac{dy}{dt} + 12y = 8\frac{dx}{dt} + 3x$$

z-Transform

Assume a sampling rate T

$$t = kT$$

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

Then

$$y(k+1) = z^{k+1} = z \cdot z^k = z \cdot y(k)$$

'zY' means "the next value of Y"

This in turn converts difference equations into algebraic equations in z.

Implementing K(z) in Software

$$Y = G(z)X = \left(\frac{a_2 z^2 + a_1 z + a_0}{z^3 + b_2 z^2 + b_1 z + b_0}\right) X$$

i) Cross multiply:

$$(z^3 + b_2 z^2 + b_1 z + b_0)Y = (a_2 z^2 + a_1 z + a_0)X$$

ii) Convert back to the time domain, noting that zY means y(k+1):

$$y(k+3) + b_2 y(k+2) + b_1 y(k+1) + b_0 y(k) = a_2 x(k+2) + a_1 x(k+1) + a_0 x(k)$$

or

$$y(k) + b_2 y(k-1) + b_1 y(k-2) + b_0 y(k-3) = a_2 x(k-1) + a_1 x(k-2) + a_0 x(k-3)$$

iii) Solve for y(k)

$$y(k) = -b_2 y(k-1) - b_1 y(k-2) - b_0 y(k-3) + a_2 x(k-1) + a_1 x(k-2) + a_0 x(k-3)$$

iv) Write this in code:

```
while(1) {

    x3 = x2;            // x(k-3)
    x2 = x1;            // x(k-2)
    x1 = x0;            // x(k-1)
    x0 = A2D_Read(0);   // read x(k) from the A/D

    y3 = y2;            // y(k-3)
    y2 = y1;            // y(k-2)
    y1 = y0;            // y(k-1)

    y0 = -b2*y1 - b1*y2 - b0*y3 + a2*x1 + a1*x2 + a0*x3;
    D2A(y0);            // output y(k) to the D/A converter
    Wait_10ms();
    }
```

Example 2: Implement the following digital filter (T = 10ms)

$$Y = \left(\frac{0.2z(z-0.9)}{(z-1)(z-0.5)}\right)X$$

Solution: Multiply it out

$$Y = \left(\frac{0.2(z^2-0.9z)}{z^2-1.5z+0.5}\right)X$$

Cross multiply and solve for the highest power of zY

$$(z^2 - 1.5z + 0.5)Y = 0.2(z^2 - 0.9z)X$$

$$z^2 Y = 1.5z - 0.5Y + 0.2(z^2 - 0.9z)X$$

meaning

$$y(k) = 1.5y(k-1) - 0.5y(k-2) + 0.2(x(k) - 0.9x(k-1))$$

In code, only one line changes

```
while(1) {

   x2 = x1;          // x(k-2)
   x1 = x0;          // x(k-1)
   x0 = A2D_Read(0);  // read in x(k) from the A/D

   y2 = y1;          // y(k-2)
   y1 = y0;          // y(k-1)

   y0 = -1.5*y1 +0.5*y2 + 0.2*(x0 - 0.9*x1);

   Wait_10ms();

   }
```

Note:

- Filters in the z-domain can be implemented exactly in software. That isn't true in the LaPlace domain.
- To change the filter, you just change one line of code. That's much easier than building a new op-amp filter.
- Complex poles and zeros are not a problem in the z-domain. All you care about are the coefficients in the numerator and denominator polynomials.
- If you have a 3rd-order filter, you need to remember the 3 previous values of the inputs and outputs. A 4th-order filter remembers the 4 previous values.

One other important thing to note:

- In the s-domain, we don't like to have more zeros than poles. More zeros than poles means you're differentiating the input. This tends to create a noise amplifier.
- In the z-domain, you cannot have more zeros than poles. More zeros than poles means you're using future values of the input - which I don't know how to do.

Also also

- You have to have integer powers of s. $s^{1/2}Y$ means "the half-derivative of Y". I have no idea what a half-derivative is. $s^{1/2}Y$ doesn't make sense.
- You have to have integer powers of z. $z^{1/2}Y$ means "the value of Y next time you half-call the subroutine." I know how to call a subroutine one time. I know how to call it two times. I don't know how to call a subroutine half a time. $z^{1/2}Y$ doesn't make sense either.

Find the response of G(z) for a sinusoidal input

LaPlace assumes

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

If

$$t = kT$$

then

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

or

$$y(k) = \left(e^{sT}\right)^k$$

This is identical to the assumption behind z-transforms.

$$\boxed{z = e^{sT}} \qquad s \to j\omega \qquad z = e^{j\omega T}$$

(note: TI calculators need to be in radian mode for this to work.)

Find y(t) given G(s):

$$Y = \left(\frac{20}{(s+1)(s+5)}\right) X$$

where

$$x(t) = 3\sin(4t)$$

Solution: Evaluate at s = j4

$$Y = \left(\frac{20}{(s+1)(s+5)}\right)_{s=j4} (0 - j3) = -2.066 + j0.947$$

meaning

$$y(t) = -2.066\cos(4t) - 0.947\sin(4t)$$

Find y(t) given G(z) (T = 10ms)

$$Y = \left(\frac{0.02z}{(z-0.9)(z-0.8)}\right) X$$

$$x(t) = 3\sin(4t)$$

Solution: Evaluate at

$$s = j4$$

$$z = e^{sT} = e^{j0.04} = 1\angle 2.291^0$$

$$Y = \left(\frac{0.02z}{(z-0.9)(z-0.8)}\right)_{z=1\angle 2.291^0} (0 - j3) = -1.423 - j2.366$$

meaning

$$y(t) = -1.423\cos(4t) + 2.366\sin(4t)$$

You can verify this in VisSim:

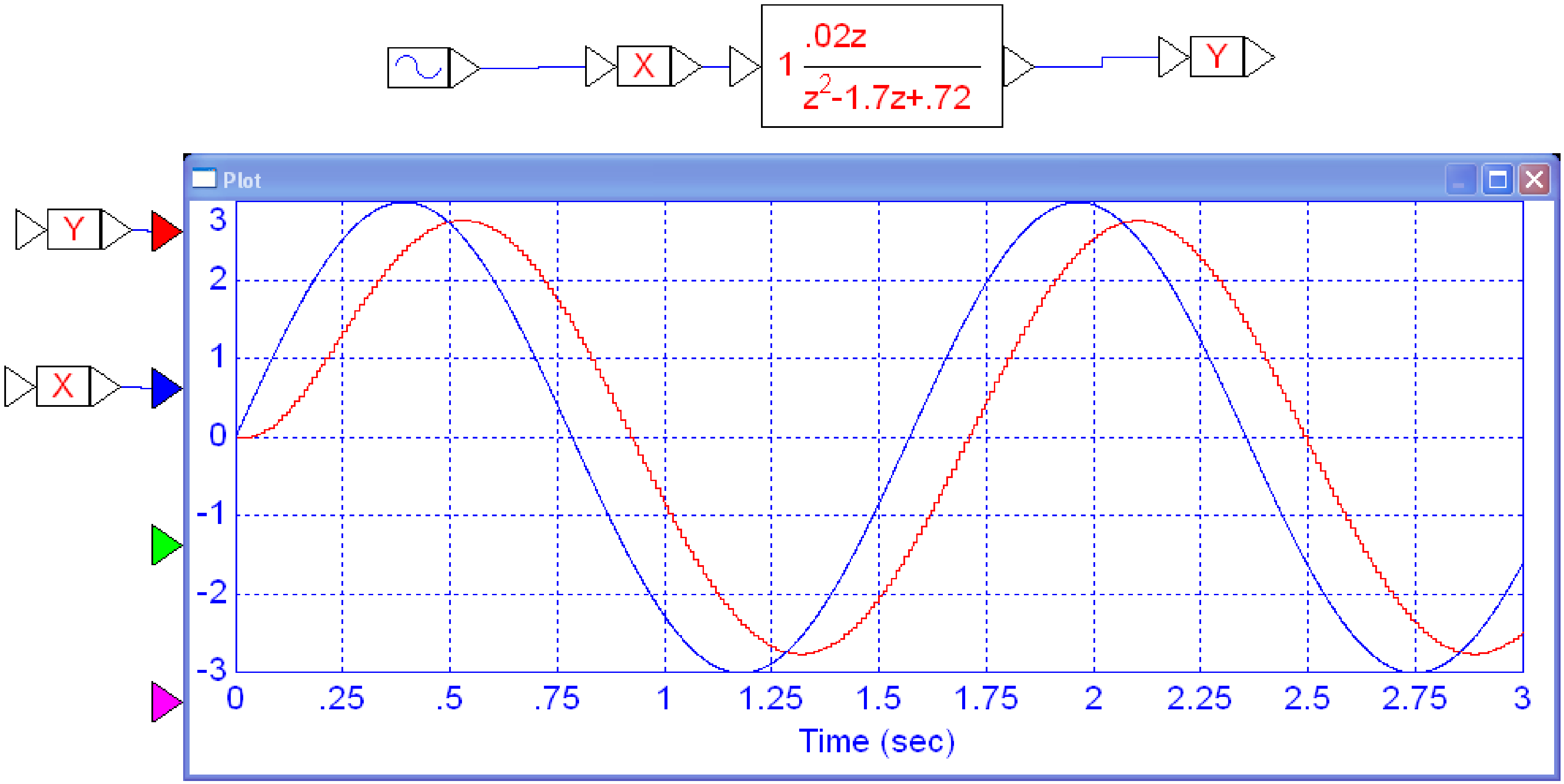


Table of z-transforms

If you want to find the output of a filter G(s) with a step input, you use LaPlace transforms along with a table of LaPlace transforms and partial fraction expansion.

Similarly, if you want to find the output of a filter G(z) with a step input, you use z-transforms along with a table of z-transforms and partial fraction expansion.

i) Delta Function $\delta(k)$. The discrete-time delta function is

$$\delta(k) = \begin{cases} 1 & k = 0 \\ 0 & otherwise \end{cases}$$

k	0	1	2	3	4	5	6	7
delta(k)	1	0	0	0	0	0	0	0

The z-transform of a delta function is '1', just like the s-domain.

ii) Unit Step: The unit step is

$$u(k) = \begin{cases} 1 & k \geq 0 \\ 0 & otherwise \end{cases}$$

It's z-transform can be deriveds as follows. The unit step is:

k	0	1	2	3	4	5	6	7
u(k)	1	1	1	1	1	1	1	1
(1/z)*u(k)	0	1	1	1	1	1	1	1
Subtract								
(1-1/z)u(k)	1	0	0	0	0	0	0	0

So,

$$\left(1 - \tfrac{1}{z}\right) u(k) = \left(\tfrac{z-1}{z}\right) u(k) = 1$$

$$u(k) = \tfrac{z}{z-1}$$

iii) Decaying Exponential. Let

$$x(k) = a^k u(k)$$

k	0	1	2	3	4	5	6	7
x(k)	1	a	a^2	a^3	a^4	a^5	a^6	a^7
a*(1/z)*x	0	a	a^2	a^3	a^4	a^5	a^6	a^7
Subtract								
(1-a/z)x	1	0	0	0	0	0	0	0

so

$$(1 - \tfrac{a}{z})X = (\tfrac{z-a}{z})X = 1$$

$$X = (\tfrac{z}{z-a})$$

These let you create a table of z-transforms like we had in the s-domain:

function	y(k)	Y(z)
delta	$\delta(k)$	1
unit step	$u(k)$	$\left(\frac{z}{z-1}\right)$
decaying exponential	$a^k u(k)$	$\left(\frac{z}{z-a}\right)$
damped sinewave	$2b \cdot a^k \cdot \cos(k\theta + \phi) \cdot u(k)$	$\left(\frac{(b\angle\phi)z}{z-(a\angle\theta)}\right) + \left(\frac{(b\angle-\phi)z}{z-(a\angle-\theta)}\right)$

Time Response in the z-Domain

Find y(t) assuming x(t) is a unit step:

$$Y = \left(\frac{0.2z}{(z-0.9)(z-0.5)}\right)X$$

i) Replace X(z) with the z-transform of a step

$$Y = \left(\frac{0.2z}{(z-0.9)(z-0.5)}\right)\left(\frac{z}{z-1}\right)$$

ii) Use partial fractions

$$Y = \left(\frac{0.2z}{(z-1)(z-0.9)(z-0.5)}\right)z = \left(\left(\frac{4}{z-1}\right) + \left(\frac{-4.5}{z-0.9}\right) + \left(\frac{0.5}{z-0.5}\right)\right)z$$

iii) Now apply the table entries

$$y(k) = 4 - 4.5 \cdot (0.9)^k + 0.5 \cdot (0.5)^k \qquad k >= 0$$

Complex Poles: Find the step response for:

$$Y = \left(\frac{0.2z}{\left(z - 0.9\angle 10^0\right)\left(z - 0.9\angle -10^0\right)} \right) X$$

i) Replace X with its z-transforrm (a unit step)

$$Y = \left(\frac{0.2z}{\left(z - 0.9\angle 10^0\right)\left(z - 0.9\angle -10^0\right)} \right) \left(\frac{z}{z-1} \right)$$

ii) Factor our a z and use partial fractions

$$Y = \left(\left(\frac{5.355}{z-1} \right) + \left(\frac{2.98\angle 153.97^0}{z - 0.9\angle 10^0} \right) + \left(\frac{2.98\angle -153.97^0}{z - 0.9\angle -10^0} \right) \right) z$$

iii) Convert back to time using the table of z-transforms

$$y(k) = 5.355 + 4.859 \cdot (0.9)^k \cdot \cos\left(10^0 \cdot k - 153.97^0\right) \qquad k >= 0$$

Time Value of Money

You can also solve time-value of money problems using z-transforms.

Assume you borrow $100,000 for a house. How much do you have to pay each month to pay off the loan in 10 years?

- Assume 6% interest per year (0.5% per month).

Solution: Let x(k) be how much money you owe today. The amount you owe next month, x(k+1), is

$$x(k+1) = 1.005x(k) - p + X(0) \cdot \delta(k)$$

where 'p' is your monthly payment starting at k=1. (a step delayed by one sample). Converting to the z-domain

$$zX = 1.005X - p\left(\frac{z}{z-1}\right)\left(\frac{1}{z}\right) + X(0)$$

$$zX = 1.005X - p\left(\frac{1}{z-1}\right) + X(0)$$

$$(z-1.005)X = X(0) - p\left(\frac{1}{z-1}\right)$$

$$X = \left(\frac{X(0)}{z-1.005}\right) - p\left(\frac{1}{(z-1)(z-1.005)}\right)$$

Using partial fractions

$$zX = \left(\frac{z}{z-1.005}\right)X(0) - pz\left(\frac{1}{(z-1)(z-1.005)}\right)$$

$$zX = \left(\frac{z}{z-1.005}\right)X(0) + pz\left(\left(\frac{200}{z-1}\right) - \left(\frac{200}{z-1.005}\right)\right)$$

Converting back to the time domain

$$zx(k) = 1.005^{k}X(0) - 200p(1.005^{k} - 1)u(k)$$

Divide by z (delay one sample)

$$x(k) = 1.005^{k-1}X(0) - 200p(1.005^{k-1} - 1)u(k-1)$$

After 120 payments (10 years), the balance should be zero

$$x(k) = 1.005^{k-1}X(0) - 200p(1.005^{k-1} - 1)u(k-1)$$

$$x(121) = 0 = \$181,939 - 200p(0.8194)$$

$$p = \$1110.02$$

Your monthly payments are $1,110.02 starting at month #1 and continuing for 120 payments.

If you stretch this out to 30 years (k = 360 payments), the monthly payment becomes

$$x(361) = 0 = \$602,257 - 200p(5.0226)$$

$$p = \$599.55$$

Note: Paying off the loan over a time span 3 times longer

- Reduces the monthly payments by only 46% less, and
- Increases the total amount you'll pay on the loan from $133,224 to $215,838.

Also also: That's pretty much all a business calculator is: a calculator which does z-transforms where the keys are renamed

- interest rate
- initial loan value, and
- number of payments