z Transforms

ECE 461/661 Controls Systems

Jake Glower - Lecture #29

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

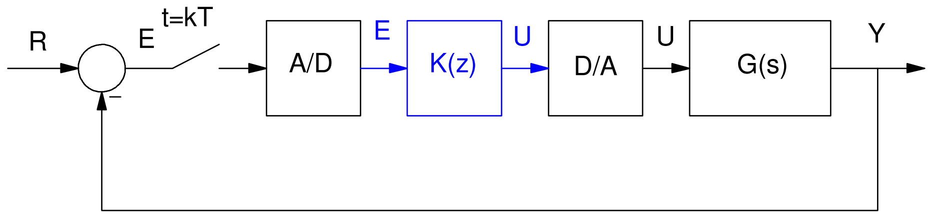
Hardware vs. Software:

Anything you can do in software you can do in hardware.

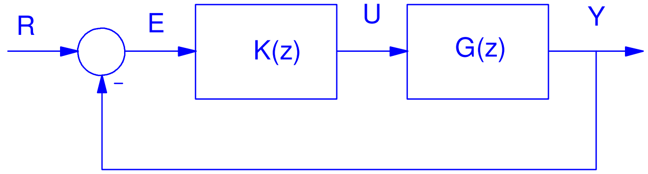
- Replace K(s) with a microprocessor, K(z)
- Microprocessor sees a discrete-time system: G(z)

Here, the microcontroller

- Samples the error every T seconds
- Reads the analog world through an A/D
- Executes a program, K(z)
- Outputs data through a D/A



Actual Feedback Control System



The world as seen by the microcontroller

Why use a microcontroller?

- Code that ran yesterday should also run today.
- DC offsets 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 with microcontrollers:

- Easy to implement difference equations, $K(z)$
- Hard to implement differential equations, i.e. $K(s)$

Difference Equations

- Difference equations describe software
- We need a tool to handle difference equations

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

```
while(1) {

   k = k + 1;

   x1 = x0;
   x0 = A2D_Read(0);

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

   Wait_10ms();

   }
```

LaPlace Transforms and Differential Equations

LaPlace transforms assume all functions are in the form of

$$y = e^{st}$$

This turns differentiation into multiplication by 's'

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

This turns differential equations into algebraic equations

- Assumes algebra is easier than calculus

LaPlace Transform to Differential Equation

Assume

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

then

$$(s^2 + 7s + 12)Y = (8s + 3)X$$

or

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

z-Transform and Difference Equations

Assume all functions are in the form of

$$y = z^k$$

then

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

- zY means "the next value of Y."

z-Transforms turn difference equations into algebraic equations in z.

Implementing K(z) in Software

- Writing a program to implement K(s) is hard
- Writing a program to implement K(z) is easy

Assume

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

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):

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

This is the difference equation which relates X and Y.

iii) Time shift so you only use present and past data

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

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 filter. Assume a sampling rate of 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

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

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

$$Y = (1.5z^{-1} - 0.5z^{-2}Y + 0.2(1 - 0.9z^{-1})X$$

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

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

```
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:

- You can implement K(z) exactly
- To change a filter, change one line of code
- Complex poles and zeros are easy to implement
  - Code doesn't care if polynomials have real or complex roots
- The order of the filter is how much data you need to remember
  - 3rd-order filters use data from 3 samples ago
  - 4th-order filters use data from 4 samples ago

Also

- s-domain: Avoid having more zeros than poles
  - Results in differentiation
  - Amplifies noise
- z-domain: Avoid having more zeros than poles
  - Results in non-causal system
  - Predicts the future

Also also,

- You must have integer powers of s
  - $s^1 Y$ means "the derivative of Y"
  - $s^{0.3} Y$ means "the 0.3th derivative of Y"
  - I don't know what 0.3 derivatives are

- You must have integer powers of z
  - $z^{-1} Y$ means "the value of y(k) the previous time you called the subroutine"
  - $z^{-0.3} Y$ means "the value of y(k) the previous 0.3 time you called the subroutine"

I don't know how to call a subroutine 0.3 times

Handout: Determine the difference equation that relates X and Y

$$Y = \left(\frac{0.2z}{(z-0.8)(z-0.6)}\right)X$$

Relating s and z

LaPlace assumes

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

Assume

$$t = kT$$

$$y(kT) = e^{skT} = (e^{st})^k = z^k$$

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

This is the assumption behind z-Transforms, implying

$$\boxed{z = e^{sT}}$$

Phasor analysis for G(s)

Find y(t)

$$Y = \left(\frac{20}{(s+1)(s+5)}\right) X \qquad x(t) = 3\sin(4t)$$

Express in phasor form

$$X = 0 - j3$$

$$s = j4$$

Output = Gain * Input

$$Y = \left(\frac{20}{(s+1)(s+5)}\right)_{s=j4} (0 - j3) = -2.3816 + j0.2582$$

meaning

$$y(t) = -2.3816\cos(4t) - 0.2582\sin(4t)$$

Phasor Analysis for G(z)

Find y(t).

$$Y = \left(\frac{0.02z}{(z-0.9)(z-0.8)}\right) X \qquad x(t) = 3\sin(4t) \qquad \text{T} = 10\text{ms}$$

Solution: Convert to phasors

$$X = 0 - j3$$

$$s = j4$$

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

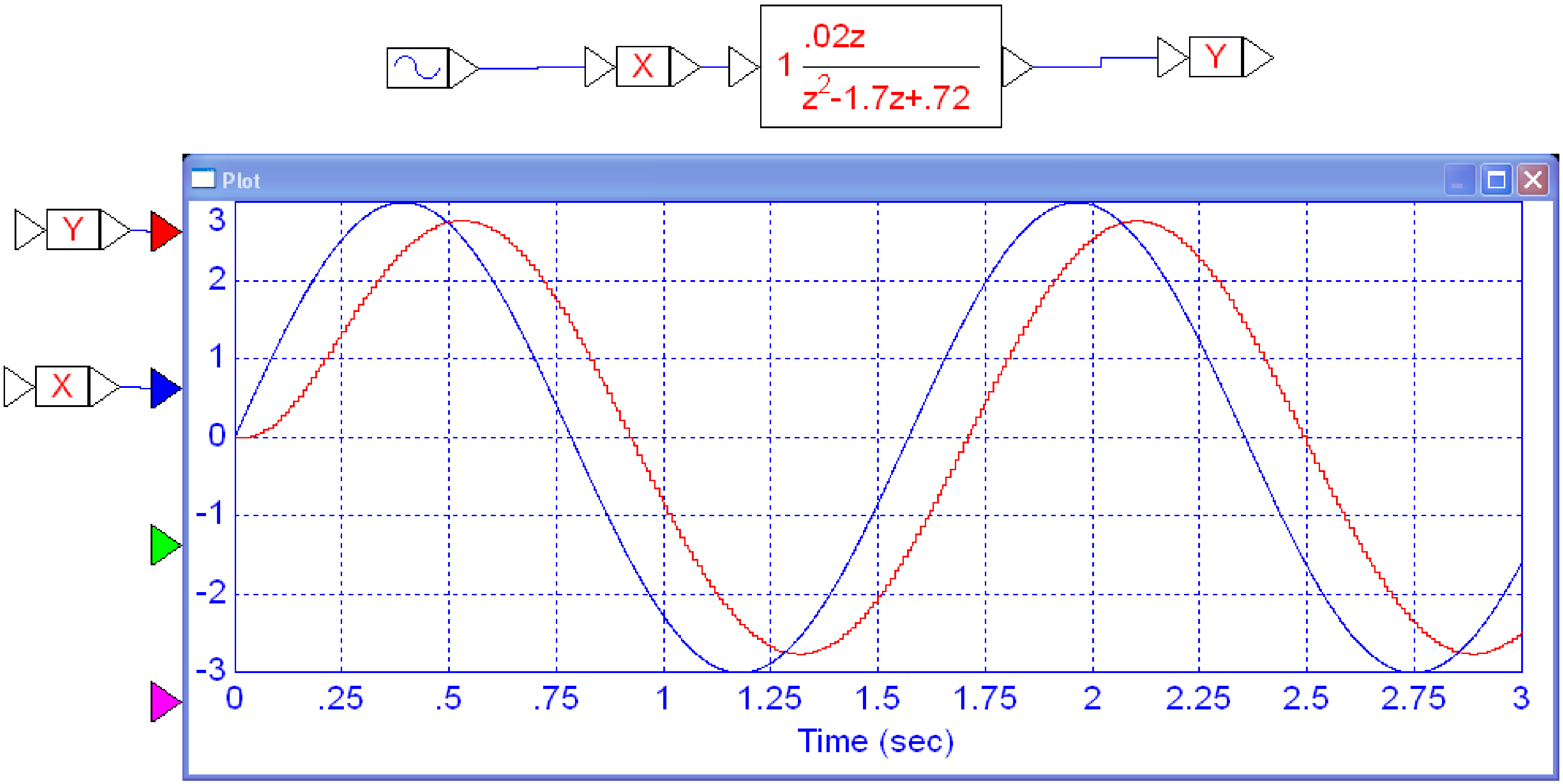
$$Y = \left(\frac{0.02z}{(z-0.9)(z-0.8)}\right)_{z=1\angle 2.291^0} (0 - j3) = -1.4226 - j2.3663$$

meaning

$$y(t) = -1.4226\cos(4t) + 2.3663\sin(4t)$$

It isn't obvious, but G(z) is a filter

- Gain varies with frequency



Handout: Determine y(t)

$$Y = \left(\frac{0.2z}{(z-0.8)(z-0.6)} \right) X$$

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

$$T = 0.1$$

z-Transforms for various functions

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.

Unit Step: The unit step is

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

It's z-transform can be derives 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,

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

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

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

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

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

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

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

Table of z-Transforms

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

Example: Real Poles

Find the step response of

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

Use partial fractions

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

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

Use the table

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

Example: Complex Poles

Find y(k)

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

Pull out a z:

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

Expand using partial fractions

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

Convert back to time using the table of z-transforms

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

Notes:

s-Plane: Rectangular works best

- The real part of s tells you the rate at which the exponential decays
- The complex part of s tells you the frequency of oscillations.

z-plane: Polar works best

- The amplitude of z tells you the rate at which the signal decays
- The angle of z tells you the frequency of oscillation.

In this case,

- The signal decays by 10% each sample $(0.9)^k$
- The phase changes by 10 degrees each sample

36 samples per cycle

Handout: Assume x(k) = u(k) (unit step). Find y(k)

$$Y = \left(\frac{0.2z}{(z-0.8)(z-0.6)}\right)X$$

Sidelight: Time Value of Money

Borrow \$100,000 @ 6% interest (0.5% per month).

- What are the monthly payments?
- This is what business calculators compute
- Constant monthly payments starting at month #1 (vs. month #0)

x(k) = loan value at month k

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

z-Transform

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

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

Solve for X

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

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

Do partial fractions

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

Multiply by z so it's in our table of z-transforms

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

Take the inverse z-transform

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

$$zx(k) = (1.005^k \cdot X(0) + 200p(1 - 1.005^k))u(k)$$

Divide by z (delay by one)

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

After 10 years (k=120 payments), the loan is zero

$$x(120) = 0 = \$181,034.50 - 162.069p$$

$$p = \$1117.02$$

Summary:

LaPlace transforms convert differential equations into algebraic equations in 's'

- Makes solving differential equations *much* easier

z-Transforms convert difference equations into algebraic equations in 'z'

- Makes solving difference equations *much* easier

The same procedures used with LaPlace transforms also work with z-transforms

- You just use a slightly different table when converting back to time