Signals and Systems

Filter Analysos & Design

Topics:

- Transfer Functions
- Filter Analysis
- Filter Design
- fmsearch()

Differential Equations and Transfer Functions

Most of the world is described by differential equations - hence the reason you are required to Calculus I-III and Differential Equations. What differential equations do is they describe the energy in a system and how it changes. Take for example a circuit with inductors and capacitors. The energy in an inductor is

$$E = \frac{1}{2}LI^2 \qquad \text{Joules}$$

The change in energy (from Calculus) is

$$\frac{d}{dt}(E) = VI = LI\,\frac{dI}{dt} \qquad \text{Watts}$$

From this, you get the fundamental equation for inductors:

$$V = L\frac{dI}{dt}$$

Similarly, for capacitors, the energy is

$$E = \frac{1}{2}CV^2 \qquad \text{Joules}$$

The change in energy is

$$\frac{d}{dt}(E) = VI = CV\frac{dV}{dt}$$

From this, you get the fundamental equation for capacitors:

$$I = C\frac{dV}{dt}$$

In short,

- A circuit with inductors and capacitors is described by a differential equation, and
- The order of the differential equation is the number of energy storage elements in the circuit (i.e. the number of capacitors and inductors.)

The same holds for most (all?) fields of engineering: it all comes down to energy. The energy in a system defines its present state. How the energy flows through a system defines how it behaves. This change inherently is a differential equation.

Suppose you have a system which is described by a differential equation, such as

$$y''' + 4y'' + 6y' + 8y = 10x' + 30x$$

where prime notation means derivative

$$y' \equiv \frac{dy}{dx}$$

A trick you will learn in Math 265 is to solve such an equation is to guess the form of the answer. In electrical and computer engineering, the usual guess is that all functions are in the form of a complex exponential

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

where 's' is a complex number. A special property of the number e is that it's derivative is

$$\frac{d}{dt}(e^{st}) = s \cdot e^{st}$$

Likewise, with this assumption, differentiation becomes multiplying by 's', and the s-operator means the derivative of'

sY means *the derivative of y(t)*

Going back to our differential equation, with this assumption, the differential equation becomes

$$s^3Y + 4s^2Y + 6sY + 8Y = 10sX + 30X$$

(note: standard notation uses lower-case for differential equations and capital letters when using s-notation - also known as LaPlace transforms.)

If you solve for Y, then

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right)X$$

or

$$Y = G(s) \cdot X$$

G(s) is called the *transfer function* of the system. Essentially, it is the gain from X to Y. Note that you can go both ways:

- Given a differential equation relating X and Y, you can find the transfer function by replacing each derivative with an 's' and solving for Y in terms of G(s)X.
- Given the transfer function relating X and Y, you can find the differential equation by cross multiplying and replacing each 's' with $\frac{d}{dt}$

Example: Find the differential equation relating X and Y given

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right)X$$

Solution: First, cross multiply

$$(s^3 + 4s^2 + 6s + 8)Y = (10s + 30)X$$

Next, replace each 's' with $\frac{d}{dt}$

$$y''' + 4y'' + 6y' + 8y = 10x' + 30x$$

or equivalently

$$\frac{d^3y}{dt^3} + 4\frac{d^2y}{dt^2} + 6\frac{dy}{dt} + 8y = 10\frac{dx}{dt} + 30x$$

Transfer Functions with Sinusoidal Inputs:

DC Inputs: Assume a system is described by the following transfer function

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right)X$$

Find y(t) assuming

$$x(t) = 2$$

Solution: The basic assumption with the LaPlace transform (s-notation) is that all functions are of the form of e^{st}.

$$x(t) = a \cdot e^{st}$$

Given that x(t) = 2, s must be zero.

$$x(t) = 2 \cdot e^{0t} = 2$$

The transfer function is true for all s. Since X only exists at s=0, you only care about the gain at that particular value of s.

s = 0:

x(t) = 2

$$\left(\frac{10s+30}{s^3+4s^2+6s+8}\right)_{s=0} = 3.75$$

The filter has a gain of 3.75 at this frequency (this value of s). Hence

Output = gain * input

y(t) = G(s) * x(t)

$$y(t) = (3.75) \cdot 2$$

$$y(t) = 7.5$$

Example 2: Assume instead that you have a sinusoidal input:

$$x(t) = 2\cos(3t)$$

From the complex exponential,

$$2\cos(3t) = real(2e^{j3t})$$

meaning

$$s = j3$$

From phasor analysis

$$a + jb \rightarrow a \cdot \cos(\omega t) - b \cdot \sin(\omega t)$$

so

$$X = 2 + j0$$

To solve for y(t), use phasors

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right) \cdot X$$

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right)_{s=j3} \cdot (2 + j0)$$

$$Y = (-1.283 - j0.659) \cdot (2 + j0)$$

$$Y = -2.566 - j1.318$$

This is shorthand (phasor) notation for

$$y(t) = -2.566\cos(3t) + 1.318\sin(3t)$$

If you prefer polar form

$$Y = 2.885\angle -152.8^0$$

which means

$$y(t) = 2.885\cos(3t - 152.8^0)$$

Both answers are correct - it's just a matter of whether you prefer rectangular or polar coordinates.

Example 3: Assume the frequency is 300 rad/sec instead:

$$x(t) = 2\cos(30t)$$

y(t) is then

$$s = j30$$

$$X = 2 + j0$$

Using phasor notation:

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right) \cdot X$$

$$Y = \left(\frac{10s+30}{s^3+4s^2+6s+8}\right)_{s=j30} \cdot (2 + j0)$$

$$Y = (-0.0111 - j0.004) \cdot (2 + j0)$$

$$Y = (-0.0223 - j0.0007)$$

which means

$$y(t) = -0.0223\cos(30t) + 0.0007\sin(30t)$$

Note that the gain changes as frequency changes. This makes G(s) a filter: it's gain changes with frequency.

MATLAB Code:

Input the frequency for s and evaluate G(s)

```
s = 0;
X = 2;
Y = (10*s + 30) / (s^3 + 4*s^2 + 6*s + 8) * (2)

Y =    7.5000

s = j*3;
X = 2 + j*0;
Y = (10*s + 30) / (s^3 + 4*s^2 + 6*s + 8) * (2 + j*0)

Y =  -2.5665 - 1.3179i

s = j*30;
X = 2 + j*0;
Y = (10*s + 30) / (s^3 + 4*s^2 + 6*s + 8) * (2 + j*0)

Y =  -0.0223 - 0.0007i
```

You can also input G(s) as a transfer function and use the MATLAB function evalfr()

```
G = tf([10,30],[1,4,6,8])

      10 s + 30
--------------------
s^3 + 4 s^2 + 6 s + 8

Y = evalfr(G, 0) * 2

Y =    7.5000

Y = evalfr(G, j*3) * 2

Y =  -2.5665 - 1.3179i

Y = evalfr(G, j*30) * 2

Y =  -0.0223 - 0.0007i
```

which are the same answers as before.

Frequency Response of a Filter:

Since the gain is a function of s and hence frequency, the gain depends upon what frequency x(t) is. If x(t0 is given, you can compute the gain at that particular frequency as we did previously. If x(t) is unknown, you can get a feel for what the filter does by plotting the gain and phase over a range.

Example: Determine the gain of G(s) over the range of 0 to 20 rad/sec for

$$G(s) = \left(\frac{10s+30}{s^3+4s^2+6s+8} \right)$$

Option 1: Compute the gain at a bunch of points from 0 to 20 rad/sec, or

Option 2: Use MATLAB. Input the frequencies you want to evaluate:

```
w = [0:5:20]';
s = j*w;
G = (10*s + 30) ./ (s.^3 + 4*s.^2 + 6*s + 8);
```

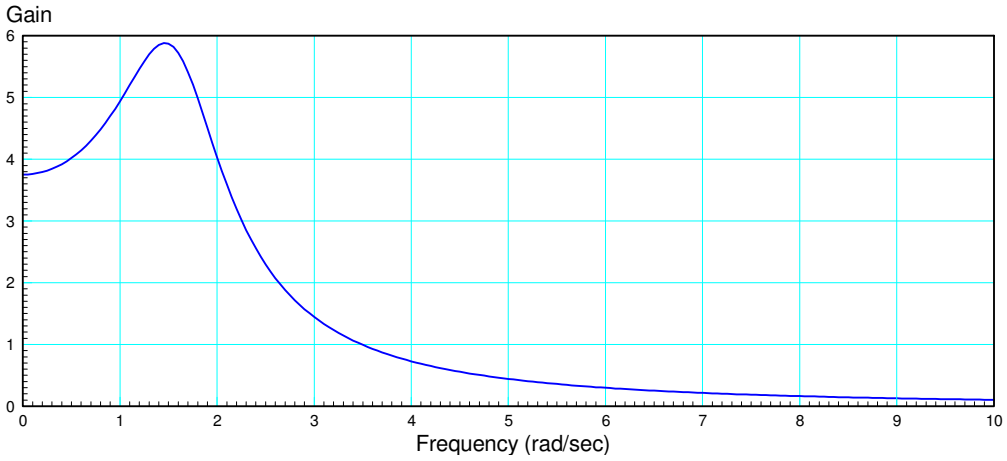
Note that dot-notation is required. MATLAB is a matrix language: matrix A^n squared only makes sense for square matrices. The dot-notation tells MATLAB to treat this as a series of scalar operations, padded together to make an array.

To print out the result

```
[w,G]

   3.7500
  -0.4294 - 0.1001i
  -0.1020 - 0.0106i
  -0.0448 - 0.0030i
  -0.0251 - 0.0013i
```

Again, note that the gain changes with frequency. A graph is a nice way to show this

```
w = [0:0.05:10]';
s = j*w;
G = (10*s + 30) ./ (s.^3 + 4*s.^2 + 6*s + 8);
plot(w,abs(G));
xlabel('Frequency (rad/sec)');
ylabel('Gain');
```

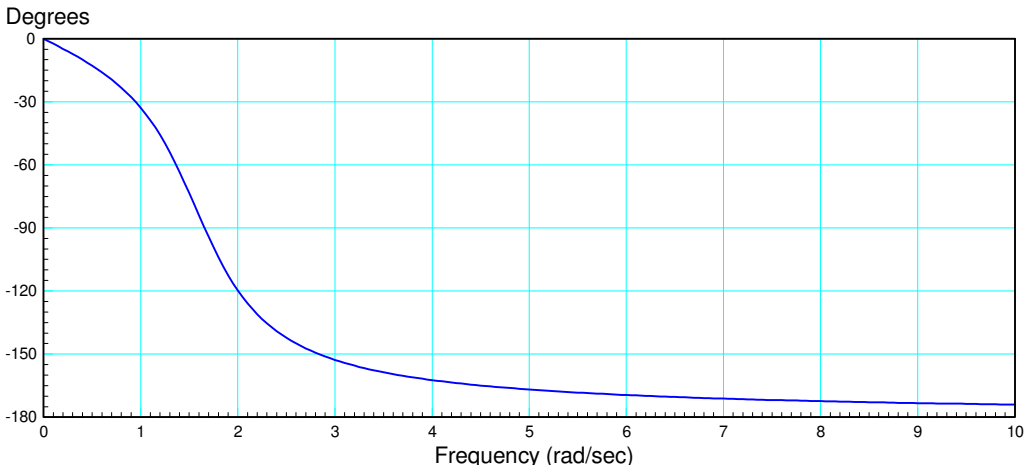


What this graph tells you is:

- The gain is large for frequencies below about 2 rad/sec and small for frequencies above 6 rad/sec. Since this passes low frequencies, it is called *a low-pass filter*
- The system has a resonance (a large gain) for frequencies near 1.5 rad/sec.

You can also plot the phase of G(s)

```
>> plot(w,angle(G)*180/pi);
>> xlabel('Frequency (rad/sec)');
>> ylabel('Angle (degrees)');
```



A negative phase shift is a delay from the input to the output. Other than that, I'm not sure what this tells you (usually we only look at gain and ignore phase).

fminsearch() and m-files

In ECE 343 Signals and Systems as well as ECE 321 Electronics II, you will learn about how to design different types of filters. In ECE 111, you can let MATLAB design the filter for you. Key to this is the function fminsearch().

This function finds the minimum of a cost function. For example, suppose you want to find the value for $\sqrt{2}$. One way is to create a function in Matlab which returns the error

$$e = x^2 - 2$$

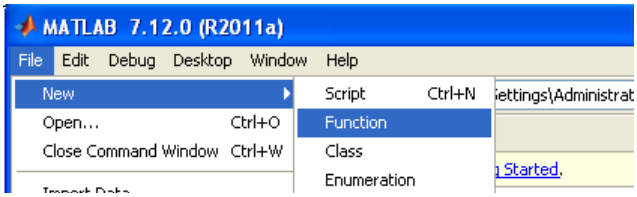
Ideally, the error should be zero x equal to $\sqrt{2}$. To make the minimum of this function the correct answer, define a cost function to be

$$J = e^2$$

In MATLAB, you can create a cost-function to compute this

- You pass your guess of (a,b,c)
- It returns the sum-squared error

First, create a function



Second, fill in this function. Suppose for example you want to find what number squared is equal to two. If you guess z, the error is

$$e = x^2 - 2$$

Ideally, the error should be zero. If you define the cost to be the sum-squred error

$$J = e^2$$

then the minimum will be the solution. The function *fminsearch* is MATLAB's function to find the minimum of a functional. Defining the cost function to be

```
function [ J ] = cost( z )

   e = z*z - 2;

   J = e^2;

end
```

results in the following. You can guess different numbers, such as 4, 3, 2, etc.

```
>> cost(4)

   196

>> cost(3)

    49

>> cost(2)

     4
```

The guess that minimizes the cost is the closest you can get. Or, you can let MATLAB do the guessing:

```
>> [a,b] = fminsearch('cost',4)

a =

    1.4143


b =

  1.5665e-008
```

With fminsearch, you tell it

- What function you want to minimize, and
- Your initial guess

It grinds for a while then spits back

- The best it could do, and
- The resulting cost (which should be zero)

Example 2: Assume a chain of length 12 meters hangs from two points: (0,0) and (10,0). What shape of the chain minimizes the potential energy?

Numerical Solution: Guess the Y coordinate for X = (1:9). The potential energy is

$$PE = mg(y_1 + y_2 + ... + y_9)$$

The length is

$$L = L_{01} + L_{12} + ... + L_{9:10}$$

Each segment has a length of

$$L_{ij} = \sqrt{\delta x^2 + \delta y^2}$$

$$L_{ij} = \sqrt{1 + (y_j - y_i)^2}$$

The total length should be 12 meters. Penalize the function for lengths differing from 12 as

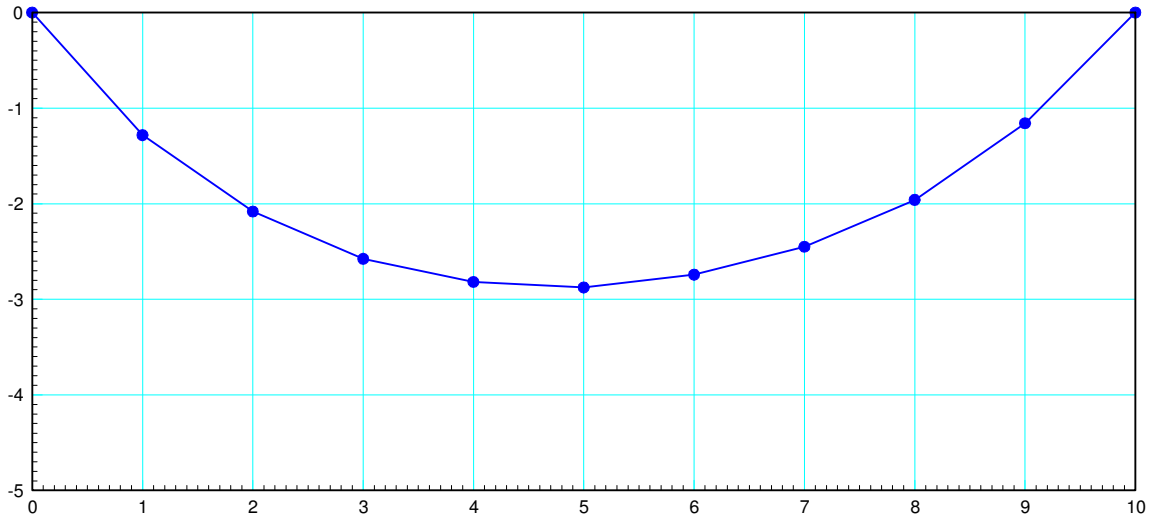
$$J = PE + \alpha(12 - L)^2$$

In MATLAB, a cost function is passed a 9x1 vector (the y location for x = 1:9).

```
function [ J ] = cost_chain( Z )

   Y = [0;Z;0];
   PE = sum(Y);
   L = 0;
   for i=2:11
      L = L + sqrt(1 + (Y(i) - Y(i-1))^2);
   end

   E = (12 - L);
   J = PE + 100*E^2;

   plot([0:10],Y,'.-');
   ylim([-5,1]);
   pause(0.01);

   end
```

It helps for the initial guess to be somewhat close to the correct answer. Let the initial guess be

```
>> y = i .* (i-10);
>> [a,b] = fminsearch('cost',0.2*y)

a =

   -1.2549
   -2.0215
   -2.5037
   -2.7737
   -2.8609
   -2.7736
   -2.5036
   -2.0212
   -1.2549


b =

  -19.8852
```

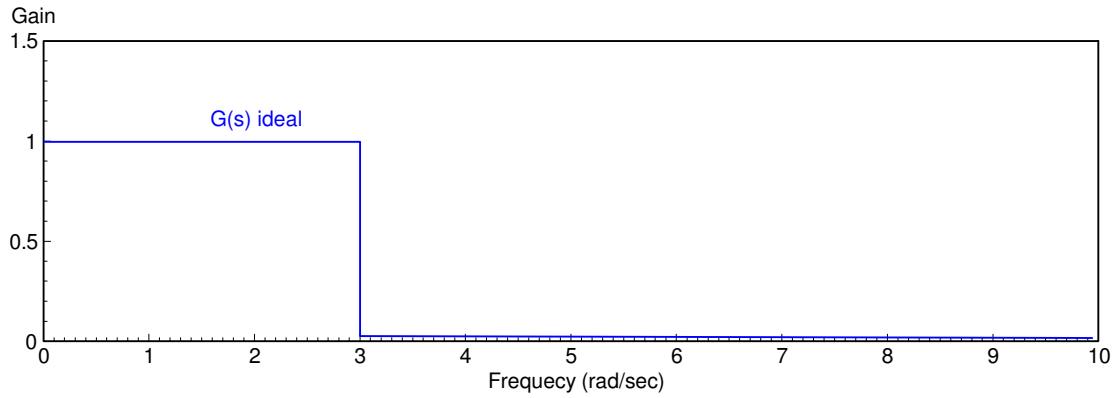


Approximate Shape of a Hanging Chain 12m Long

Filter Design with fminsearch:

Suppose you want to come up with a filter with the following gain vs. frequency (or as close as you can get)

$$|G(s)| = \begin{cases} 1 & \omega < 3 \\ 0 & \omega > 3 \end{cases}$$



Desired gain vs. frequency

Assume you guessed G(s) as a transfer function with four unknown terms (a, b, c, d, e):

$$G(s) = \left(\frac{a}{\left(s^2+bs+c\right)\left(s^2+ds+e\right)} \right)$$

To find the 'best' value for these three terms,

- First, define a cost function to define how 'good' this filter is
- Then, optimize (a, b, c, d, e)

One way to define how 'good' a filter is is to

- Compute the gain of G(s) from 0 to 10 rad/sec
- Take the difference between the actual gain and the desired gain,
- Square the difference and define the goodness of this filter to be the sum-square error

If the sum-squared error is zero, the filter is perfect. If perfection is not possible, the value of (a,b,c,d) that minimize this sum-squared difference is the best you can do.

Likewise, define the cost function to be the sum-squared error

$$J = \sum \left(|G(j\omega)| - G_{ideal}(j\omega)^2 \right)$$

First, define the cost-function as an m-file:

```
function [ J ] = costF( z )

   a = z(1);
   b = z(2);
   c = z(3);
   d = z(4);
   e = z(5);

   w = [0:0.1:10]';
   s = j*w;

   Gideal = 1 * (w < 3);

   G = a ./ ( (s.^2 + b*s + c).*(s.^2 + d*s + e) );

   e = abs(Gideal) - abs(G);

   J = sum(e .^ 2);

   plot(w,abs(Gideal),w,abs(G));
   ylim([0,1.2]);
   pause(0.01);

end
```

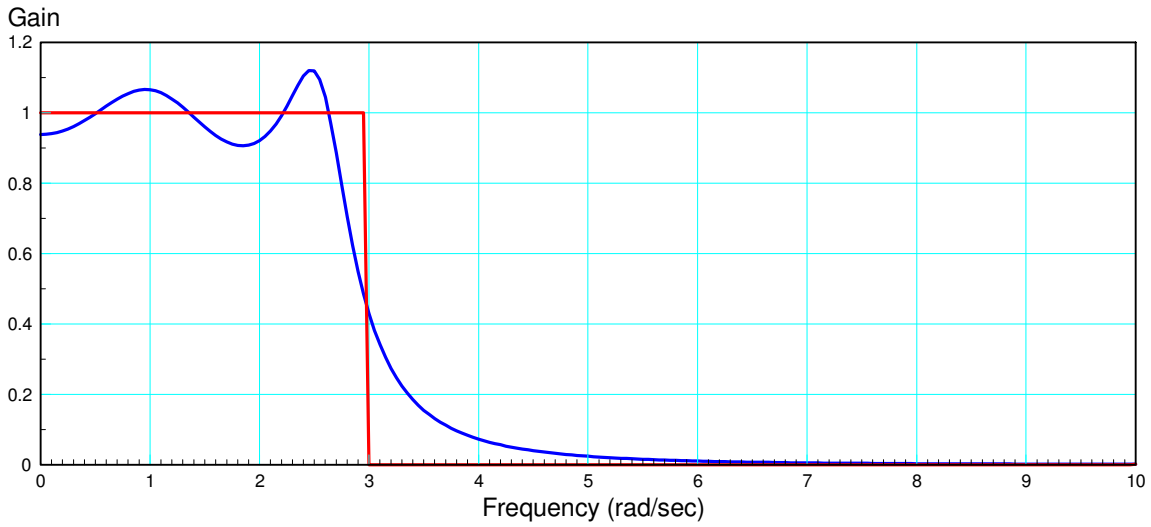
Call fminsearch with an initial guess for (a,b,c,d)

```
>> [Z,e] = fminsearch('costF',[1,2,3,4,5])

          a         b         c         d         e
Z =   10.9474    1.6224    1.7317    0.6141    6.7413

e =    0.9575

>>
```

The best you can do with this cost function is

$$G(s) = \left(\frac{10.9474}{\left(s^2+1.6224s+1.7317\right)\left(s^2+0.6141s+6.7413\right)} \right)$$

The resulting gain vs. frequency is:



Gain vs. Frequency for the target (blue) and optimized G(s) (red) found with fminsearch()

Sidelight: The filter isn't arbitrary. The gain of the filter is

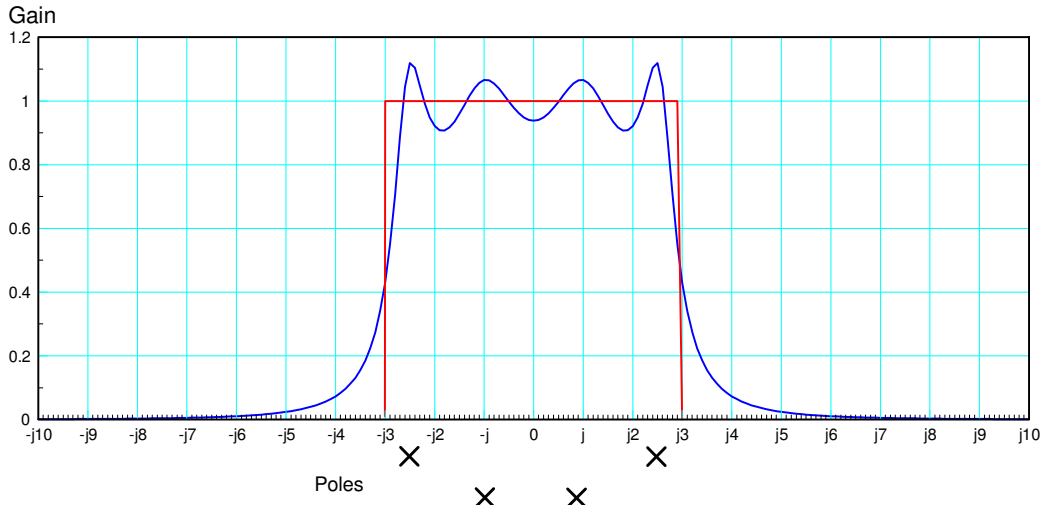
$$G(s) = \left(\frac{k}{(s-p_1)(s-p_2)(s-p_3)(s-p_4)} \right)$$

When s is close to a pole

- The denominator term becomes small, meaning
- The gain becomes large.

G(s) has four poles, which produce large gains near the complex part of the pole:

- $s = -0.8112 \pm j1.0362$
- $s = -0.3071 \pm j2.5782$



Summary

A filter is any circuit where the gain varies with frequency (i.e. any circuit which contains inuctors and/or capacitors).

The transfer function tells you the gain from the input to the output

- As well as the differential equation that filter satisfies

Filter analysis is pretty easy:

- Plug in s = jw into the transfer function

Filter design is a lot harder

- Move the poles and zeros around
- Zeros attenuate the gain near them
- Poles amplify the gain near them

Matlab's *fminsearch()* does a pretty good job at designing filters

- All you need to do is define a function whose miniimum is close to the desired filter