Poles, Zeros, and Frequency Response

ECE 321: Electronics II

Lecture #8

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

Poles, Zeros, and Frequency Response

With the previous circuits, you can build filters with

- Real poles
- Complex Poles, and
- Zeros at $s = 0$

Filter design uses this to places poles and zeros to give a desired frequency response. In this lecture we look at how the poles and zeros affect the gain vs. frequency for a filter.

Filter Analysis: Single Input

Example: Find y(t)

$$Y = \left(\frac{100}{s^2+5s+30}\right) X$$

$$x(t) = 4\cos(6t) + 5\sin(6t)$$

Solution

$$s = j6$$

$$X = 4 - j5$$

$$Y = \left(\frac{100}{s^2+5s+30}\right)_{s=j6} \cdot (4 - j5)$$

$$Y = -18.590 - j9.615$$

$$y(t) = -18.590\cos(6t) + 9.615\sin(6t)$$

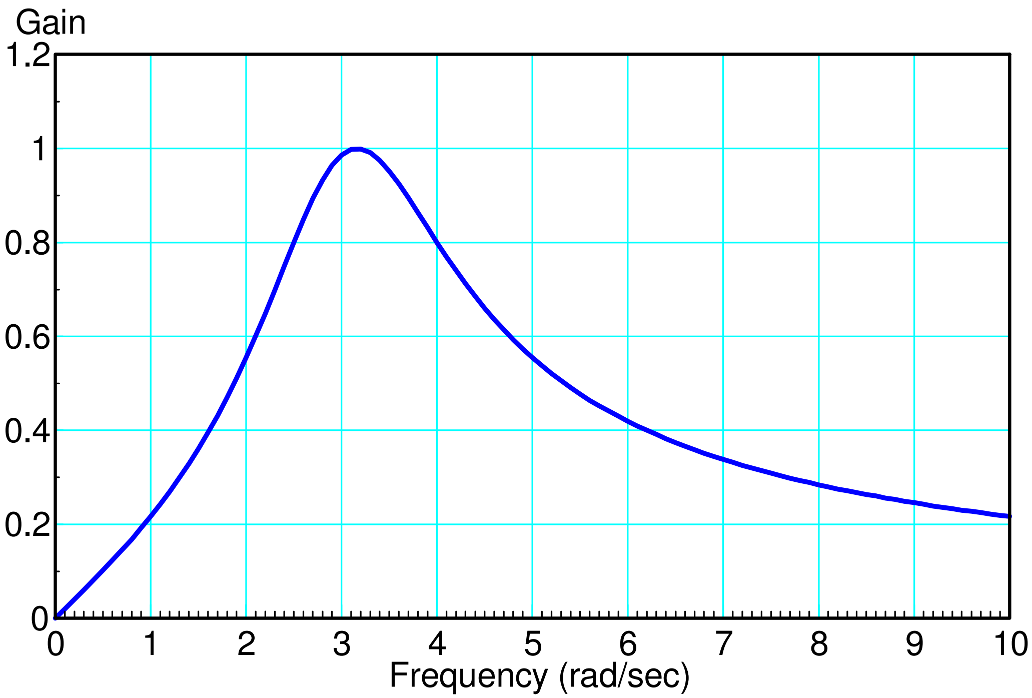
Filter Analysis: Bode Plot

Given a filter, find the gain vs. frequency.

Easy: Just plug into Matlab

$$Y = \left(\frac{2s}{s^2+2s+10}\right)X$$

```
w = [0:0.01:10]';
s = j*w;
G = 2*s ./ (s.^2 + 2*s + 10);
plot(w,abs(G));
xlabel('Frequency (rad/sec)');
ylabel('Gain');
```



Filter Design

Pick poles and zeros to match a desired frequency response

- harder

This lecture

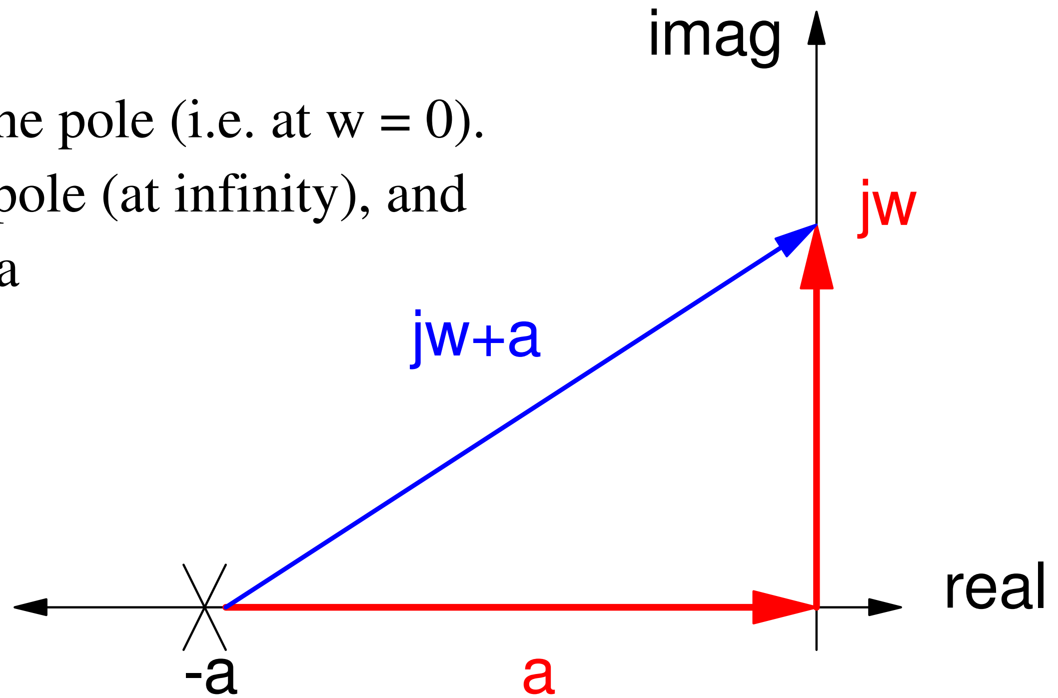
- How do real poles affect the gain vs. frequency
- How to complex poles affect the gain vs. frequency
- How to zeros affect the gain vs. frequency
- Using *fminsearch()* to design a filter

Real Poles vs. Frequency Response

$$Y = \left(\frac{1}{s+a}\right)X = \left(\frac{1}{j\omega+a}\right)X$$

Graphical:

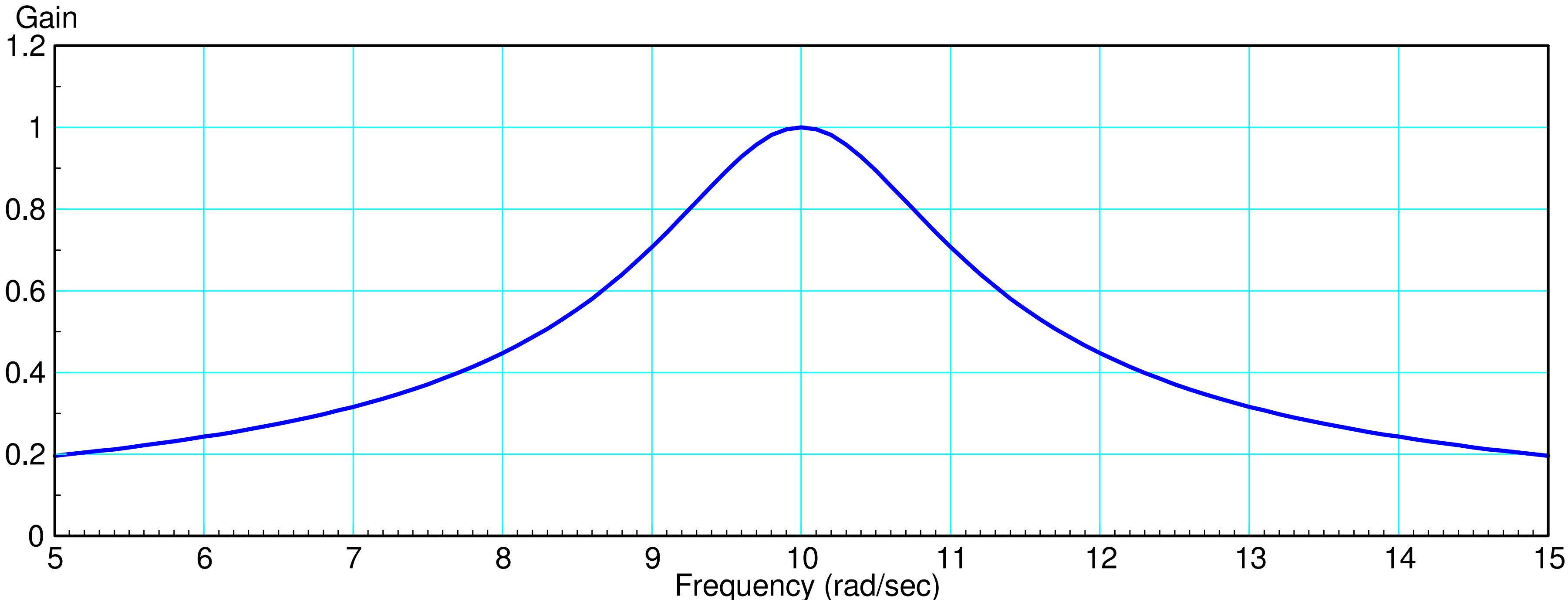
- A maximum when you're closest to the pole (i.e. at w = 0).
- Zero when you're far away from the pole (at infinity), and
- Down by $\sqrt{2}$ when the frequency is ja



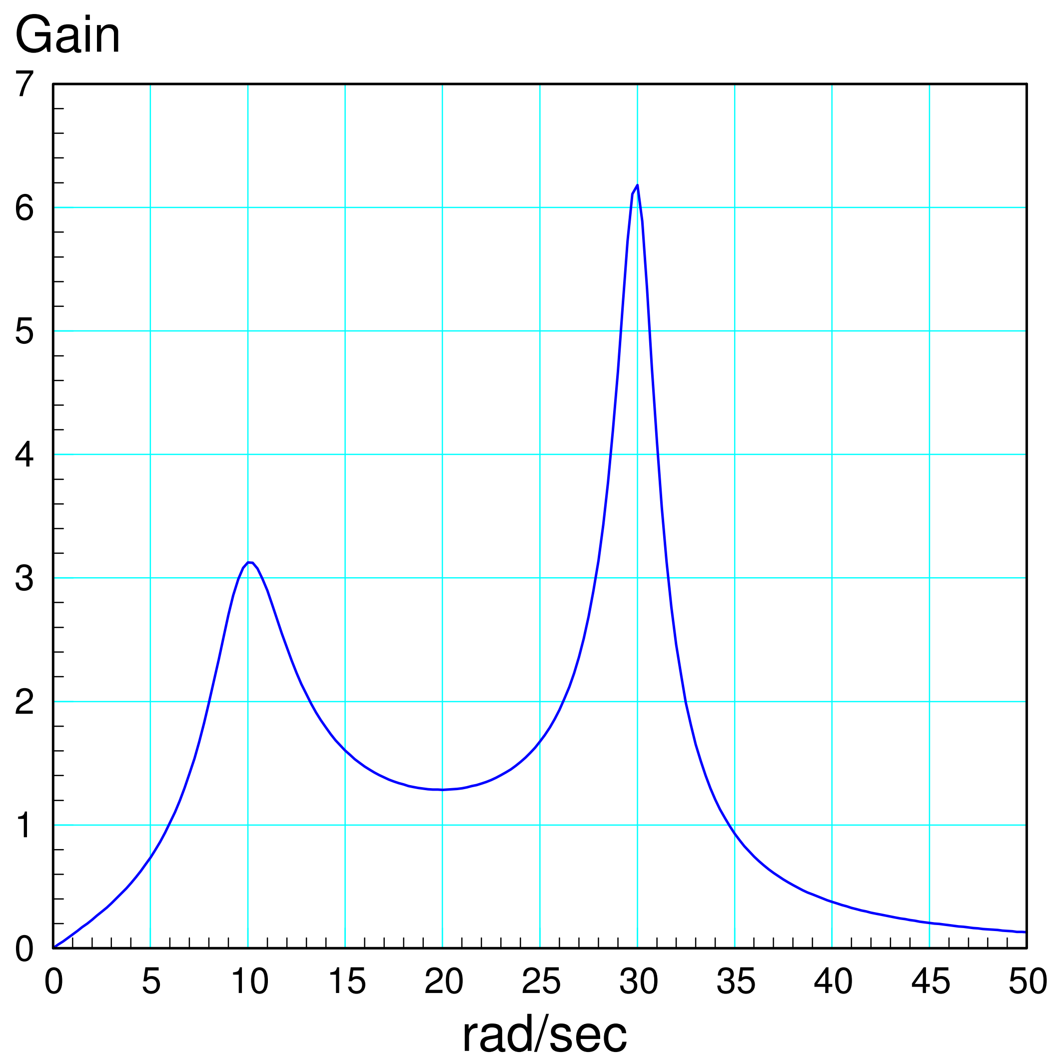
Complex Poles vs. Frequecy Respons

$$Y = \left(\frac{1}{s+1-j10}\right)X$$

- Maximum at s = j10
- Down by $\sqrt{2}$ when 1 rad/sec away from 10 (j9 and j11)



Example: Determine G(s)



Zero at s = 0

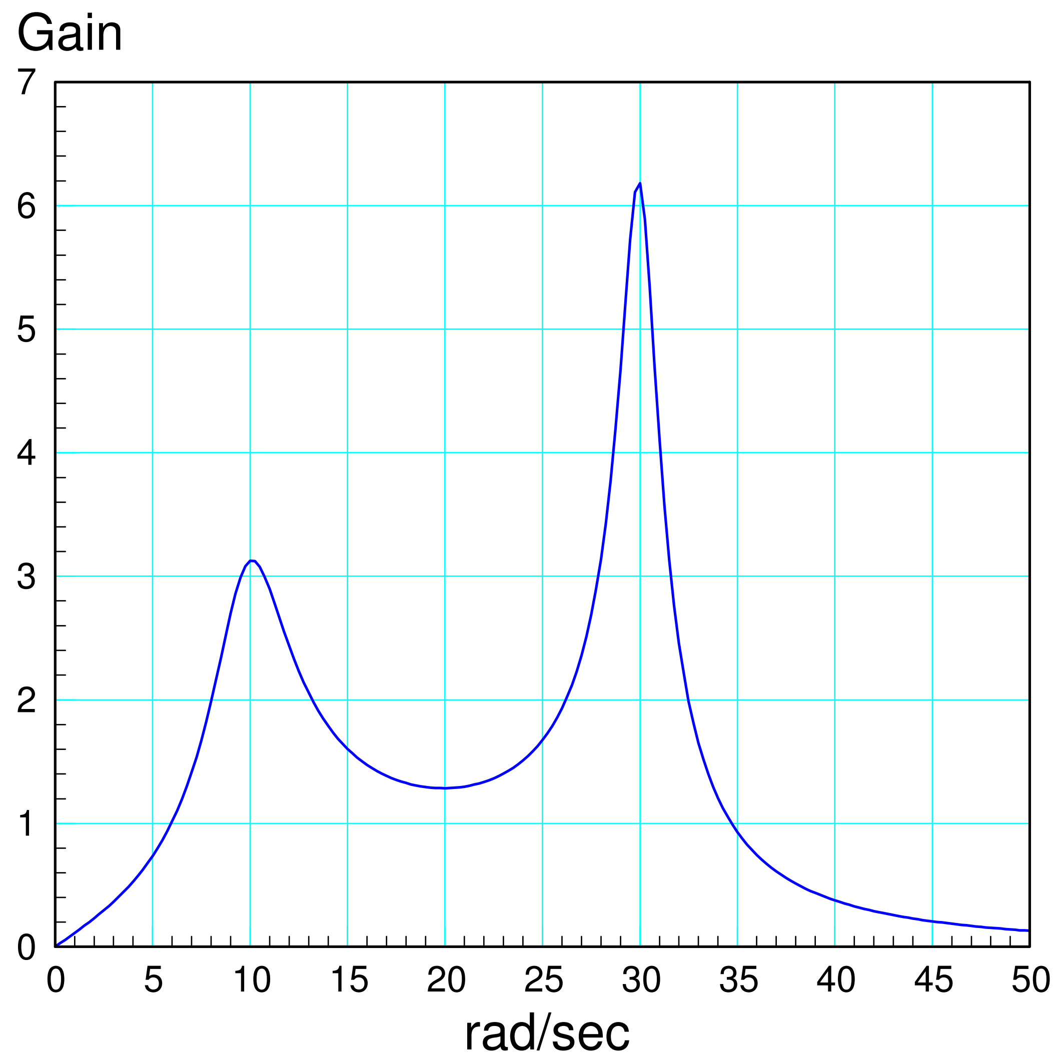
Pole at

- s = j10
- BW = 4 (real = 2)
- s = -2 +/- j10

Pole at

- s = j30
- BW = 2 (real = 1)
- s = -1 +/- j30

$$G(s) \approx \left(\frac{ks}{(s+2 \pm j10)(s+1 \pm j30)} \right)$$



Generalized Filter

$$Y = k\left(\frac{(s+z_1)(s+z_2)}{(s+p_1)(s+p_2)(s+p_3)}\right)X$$

The graphical interpriation for this filter is

$$gain = k \cdot \frac{\Pi(\text{distance from jw to the zeros})}{\Pi(\text{distance from jw to the poles})}$$

Note that

- If you're close to a zero, the gain is small (multiply by a small number)
- If you're close to a pole, the gain is large (divide by a small number)

So, a design strategy could be

- **Place zeros near frequencies you want to reject**
- **Place poles near frequencies you want to pass.**

Filter Design using *fminsearch*

Problem: Design a filter to approximate an ideal low-pass filter with a gain of

$$G_{ideal}(s) \approx \begin{cases} 1 & \omega < 4 \\ 0 & otherwise \end{cases}$$

Guess filter parameters

- poles, zeros, gain

Compute G(jw)

Compute the difference

$$E(j\omega) = |G_{ideal}(j\omega)| - |G(j\omega)|$$

Compute the cost

$$J = \int_0^{10} E^2(j\omega) \cdot d\omega$$

Use *fminsearch* to reduve the cost as much as possible

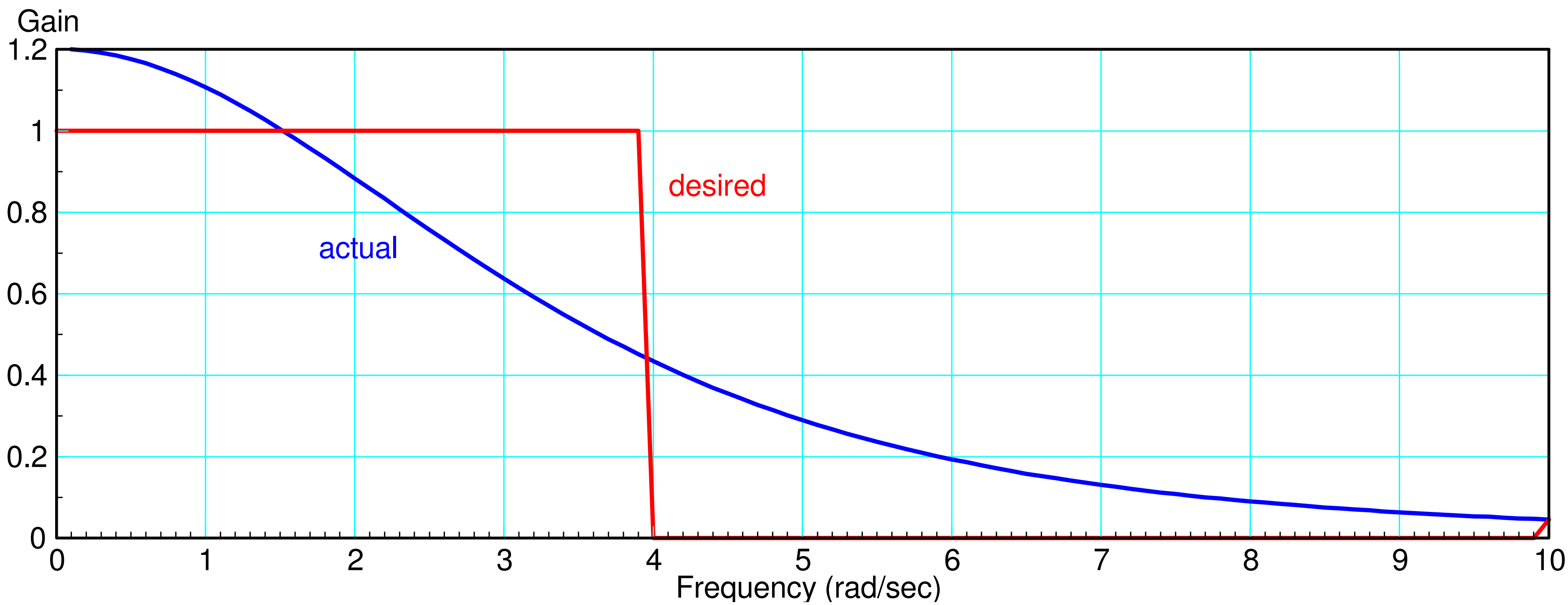
Real Poles:

$$G(s) = \left(\frac{a}{(s+b)(s+c)(s+d)(s+e)} \right)$$

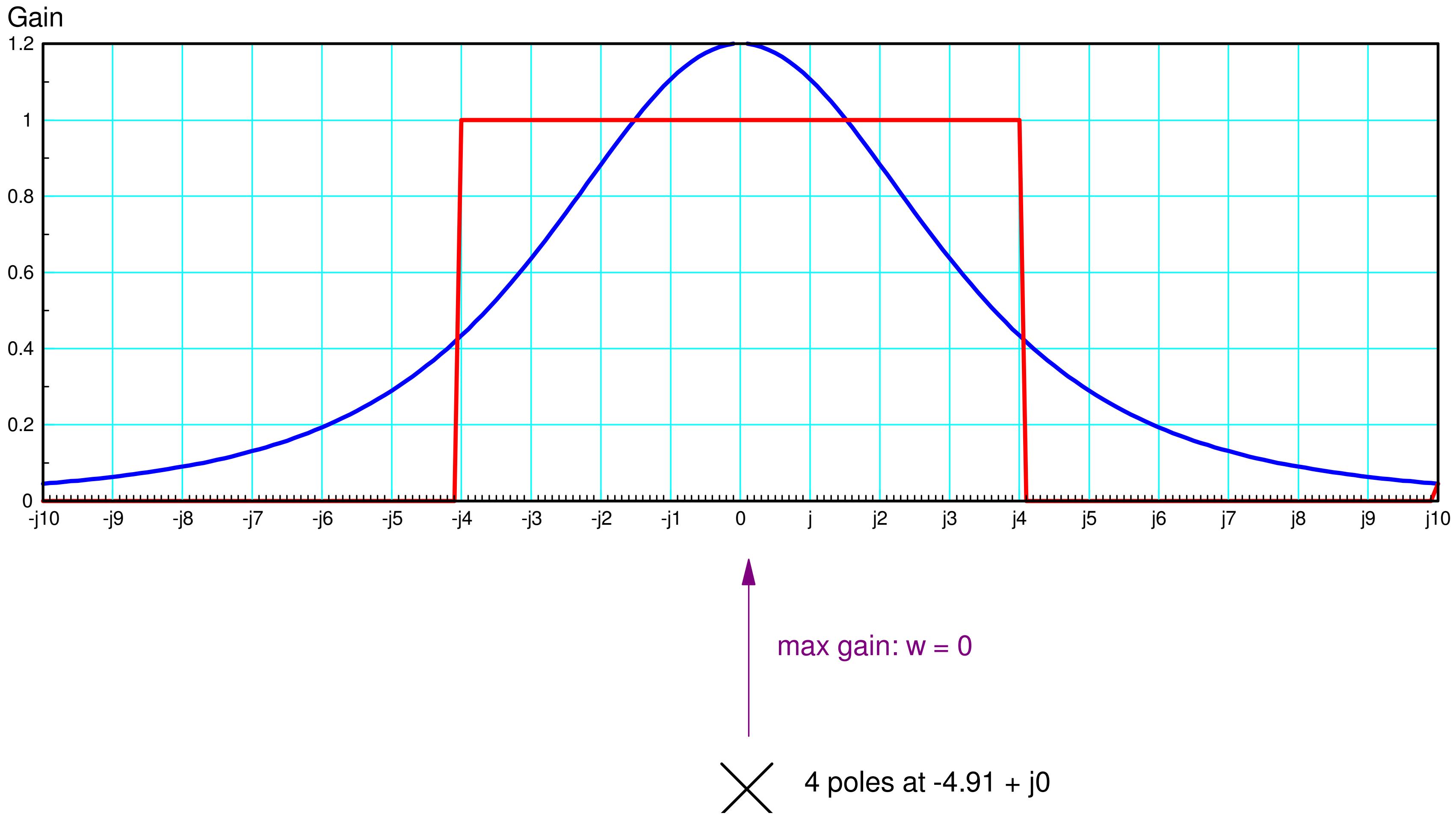
```
function [ J ] = costf( z )
 a = z(1);
 b = z(2);
 c = z(3);
 d = z(4);
 e = z(5);

 w = [0:0.01:10]';
 s = j*w;
 Gideal = 1 .* (w < 4);

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

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

 J = sum(E .^ 2);

end
```

Solution: Not great with just real poles

```
[a,b] = fminsearch('costf',[100,2,3,4,5])

a =   697.8575     4.9165     4.9165     4.9165     4.9165
b =   55.3564
```



Pole Location vs. Gain: $G(s) = \left(\frac{697}{(s+4.91)^4} \right)$



Complex Poles: $G(s) = \left(\dfrac{a}{\left(s^2 + bs + c\right)\left(s^2 + ds + e\right)} \right)$

```
function [ J ] = costf( z )
 a = z(1);
 b = z(2);
 c = z(3);
 d = z(4);
 e = z(5);

 w = [0:0.01:10]';
 s = j*w;
 Gideal = 1 .* (w < 4);

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

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

 J = sum(E .^ 2);

end
```

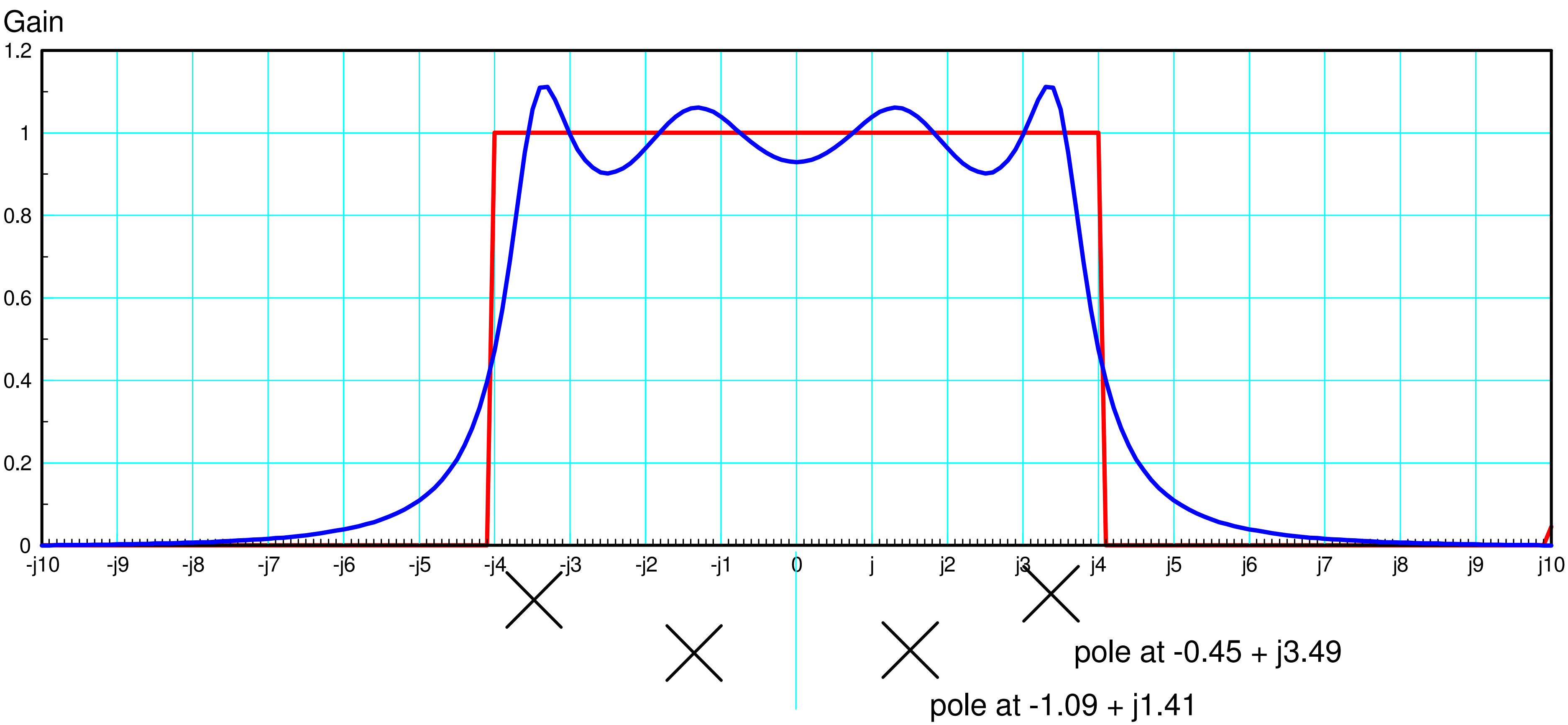
Minimizing the cost:

```
>> [a,b] = fminsearch('costf',10*rand(1,5))

a =   36.6716   0.8314   12.3599  2.1860   3.1799
b =   13.0720
```

meaning

$$G(s) = \left(\frac{36.67}{\left(s^2+0.8314s+12.3599\right)\left(s^2+2.1860s+3.1799\right)} \right)$$

The gain vs. frequency and pole location looks like:



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

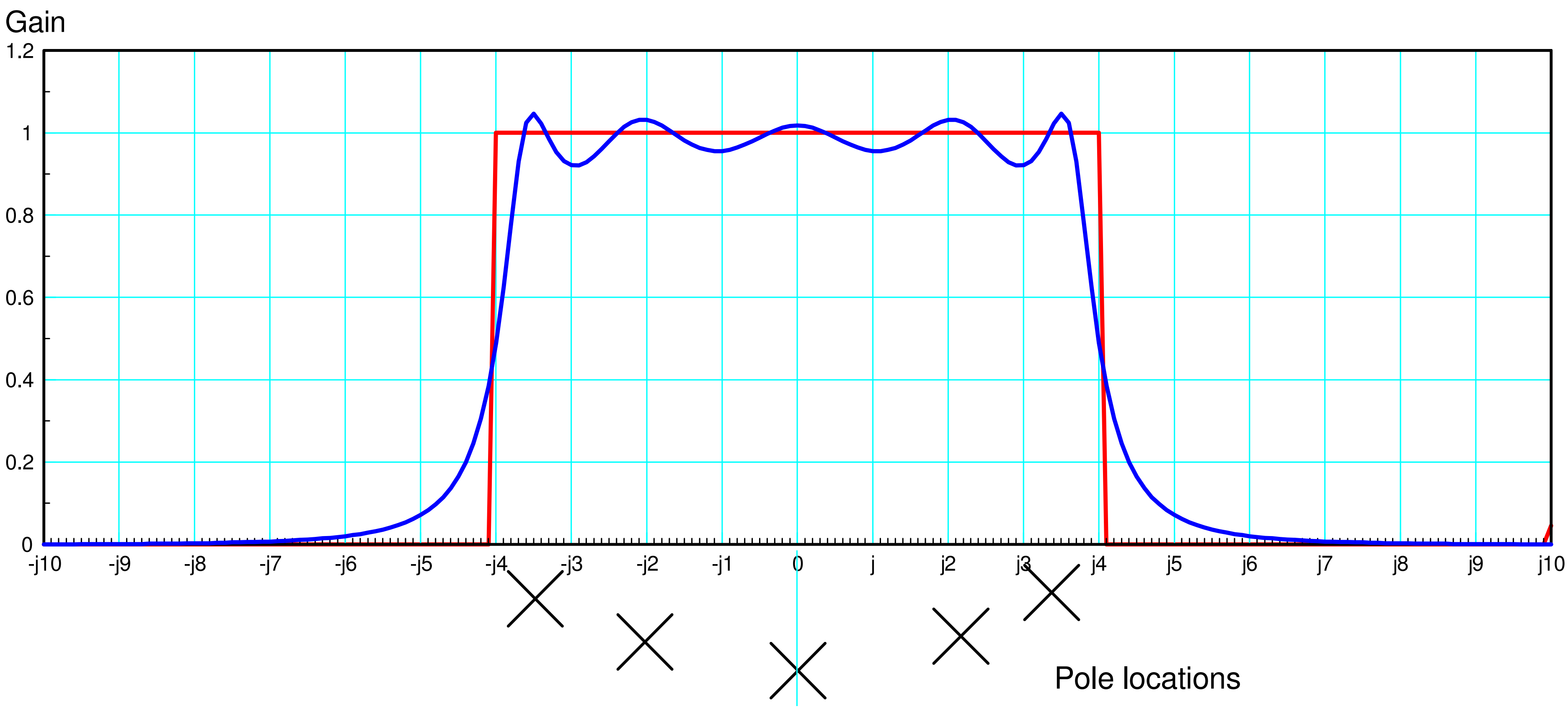
```
function [ J ] = costf( z )
 a = z(1);
 b = z(2);
 c = z(3);
 d = z(4);
 e = z(5);

 w = [0:0.01:10]';
 s = j*w;
 Gideal = 1 .* (w < 4);

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

 G = abs(G);

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

 J = sum(E .^ 2);

end
```

Running in Matlab:

```
>> [a,b] = fminsearch('costf',10*rand(1,5))

a =     1.2226    0.6761   13.5006    1.8855    5.7318

b =     9.6110
```

meaning

$$G(s) = \left(\frac{96.4}{(s+1.222)(s^2+0.6761s+13.5)(s^2+1.88s+5.73)} \right)$$



Note that there is definately a pattern here:

- You scatter N poles in the pass-band
- Place the poles on an ellipse spanning the pass-band

Summary

Filter analysis is simple

- Plug in s = jw

Filter design is a little harder

- Place zeros by frequencies you want to reject
- Place poles by frequencies you want to pass
  - Complex part of pole tells you the resonance frequency
  - Rel part of the pole tells you the bandwidth
- fminsearch() can be used to design filters