Part II. Effect of component variations on pole locations in H(s)

Given: Circuit 1 has these nominal component values:
(units are ohms, farads, and henries)

```
R1 = 1.0000
R2 = 1.0000
C1 = 0.1218
C2 = 0.2940
L1 = 0.2940
L2 = 0.1218
```

This circuit's voltage magnitude-squared frequency response function $|H(f)|^2$ may be expressed in terms of only $\{p_1, p_2, SF\}$ instead of component values $\{R_1, R_2, C_1, C_2, L_1, L_2\}$:

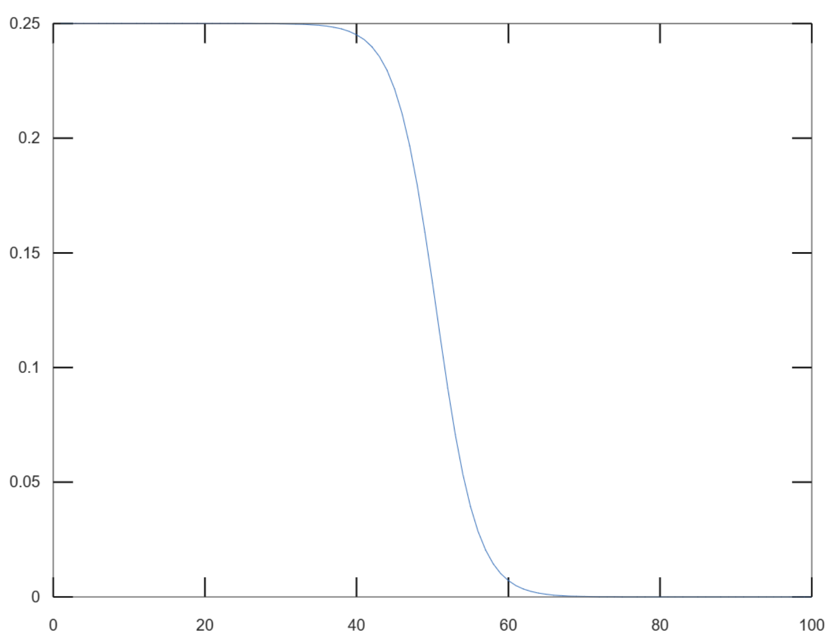
$$|H(s)|^2 = \left| \frac{SF}{(s-p_1)(s-p_1^*)(s-p_2)(s-p_2^*)} \right|^2, \text{ where } s = j2\pi f.$$

The goal of this part of Project 2 is to observe the effects of random variations in $\{C_1, C_2, L_1, L_2\}$ on pole values $\{p_1, p_2\}$.

a) Calculate |H(f)| 2 for Circuit 1, using the provided function Frequency_response1, plot the result on a log frequency scale, using f = logspace(-1,1,100).

Hmag2 = Frequency_response1(1.0000, 1.0000, 0.1218, 0.2940, 0.2940, 0.1218, f = logspace(-1,1,100))

plot(Hmag2);



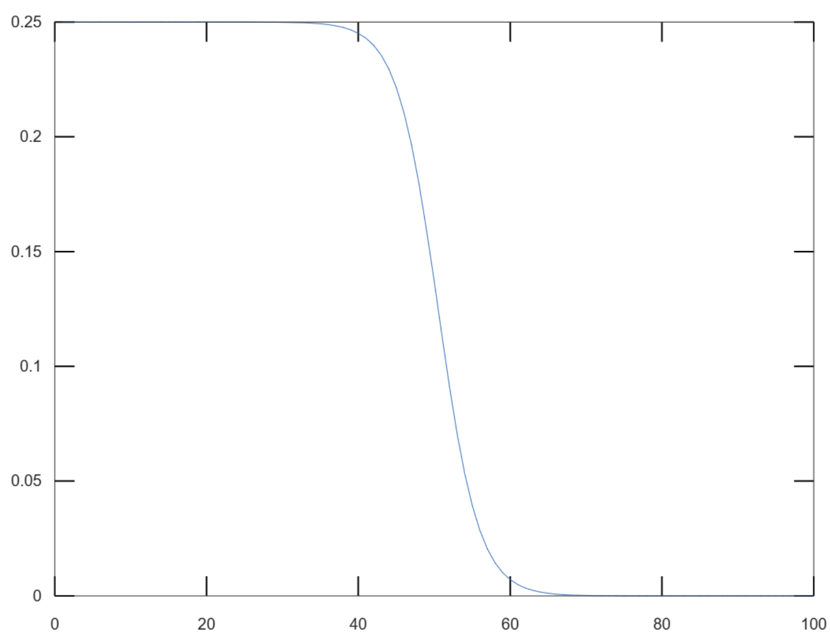
b) Find pole values {p1, p2} and scale factor SF.

[p1,p2,SF] = Poles(1.0000, 1.0000, 0.1218, 0.2940, 0.2940, 0.1218);

c) Find and plot |H(f)|^ 2 from {p1, p2, SF} to demonstrate that all information is conveyed by these three parameters.

Hmag2 = Frequency_response_from_poles(-2.4047 + 5.8065i,-5.8055 + 2.4051i,779.85,logspace(-1,1,100)) ;

plot(Hmag2);

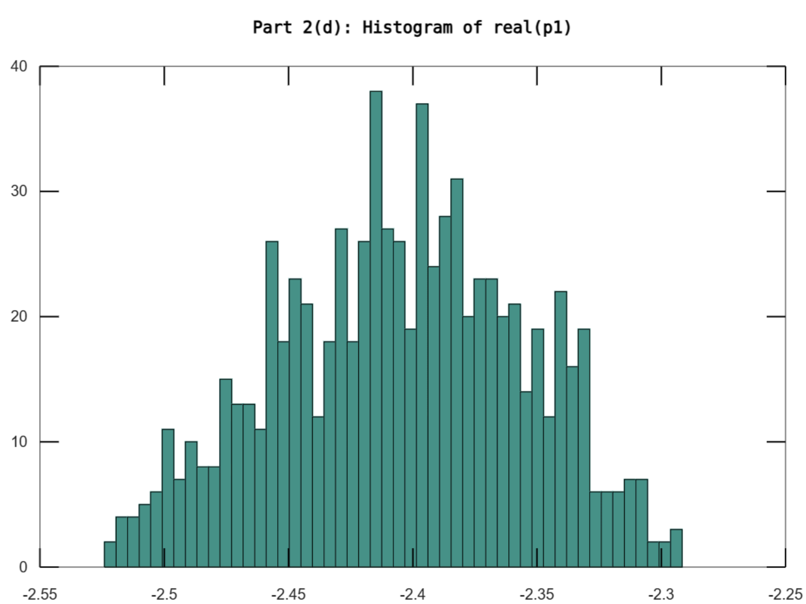


d) Keeping R1 = 1 and R2 = 1, randomize the values of capacitors and inductors, then calculate the corresponding {p1, p2}. (Generate samples of {C1, C2, L1, L2} from continuous random variables uniformly-distributed over the range [nominal component values ±5%]). Repeat this 1000 times and store the resulting perturbed pole values. Plot the histograms of Re{p1}, Im{p1}, Re{p2}, and Im{p2}.

```
[p1, p2, SF] = Poles(R1, R2, C1, C2, L1, L2);

p = 5/100;
poles = [];
for n = 1:1000
        r1 = R1;
        r2 = R2;
        c1 = C1*(1+p*(2*rand-1));
        c2 = C2*(1+p*(2*rand-1));
        l1 = L1*(1+p*(2*rand-1));
        l2 = L2*(1+p*(2*rand-1));
        [P1, P2, SF] = Poles(r1, r2, c1, c2, l1, l2);
        poles = [poles; [P1, P2]];

end

rP1 = real(poles(:,1));
iP1 = imag(poles(:,1));
rP2 = real(poles(:,2));
iP2 = imag(poles(:,2));
```

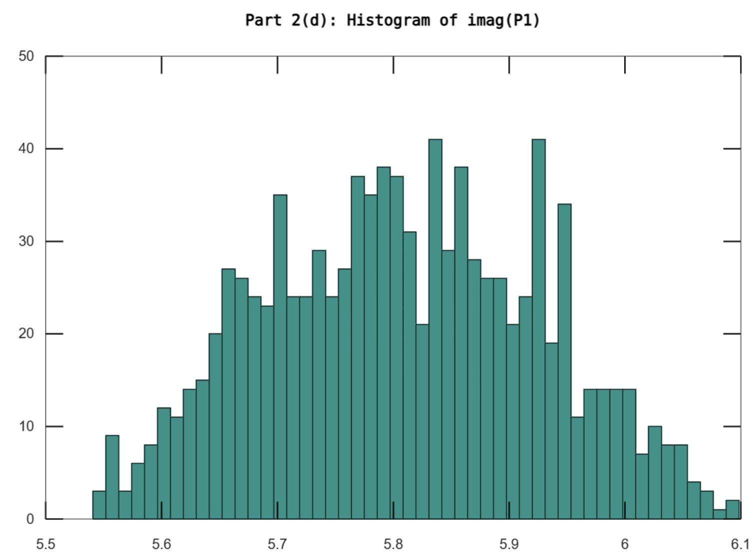
Histogram of Re{p1}

```
hist(rP1, 50)
mrP1 = mean(rP1);
srP1 = std(rP1);
title('Part 2(d): Histogram of real(p1)')
```



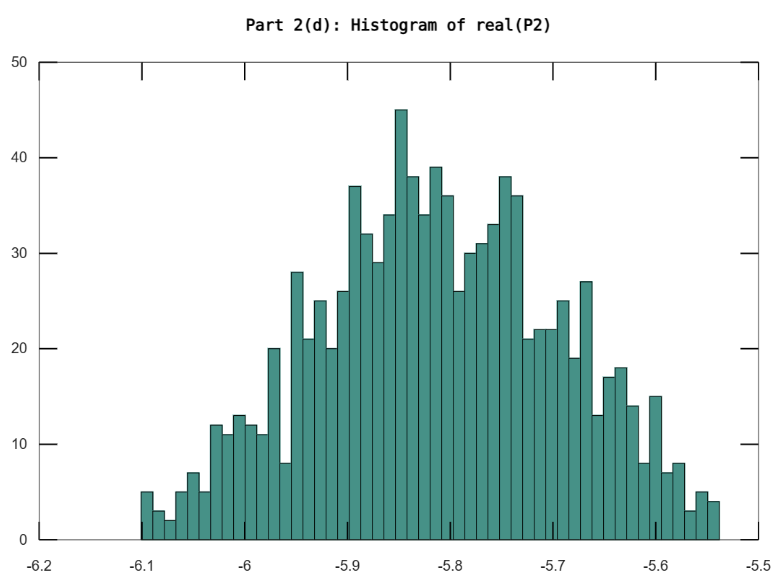
Histogram of Im{p1}

hist(iP1, 50)
miP1 = mean(iP1);
siP1 = std(iP1);
title('Part 2(d): Histogram of imag(P1))



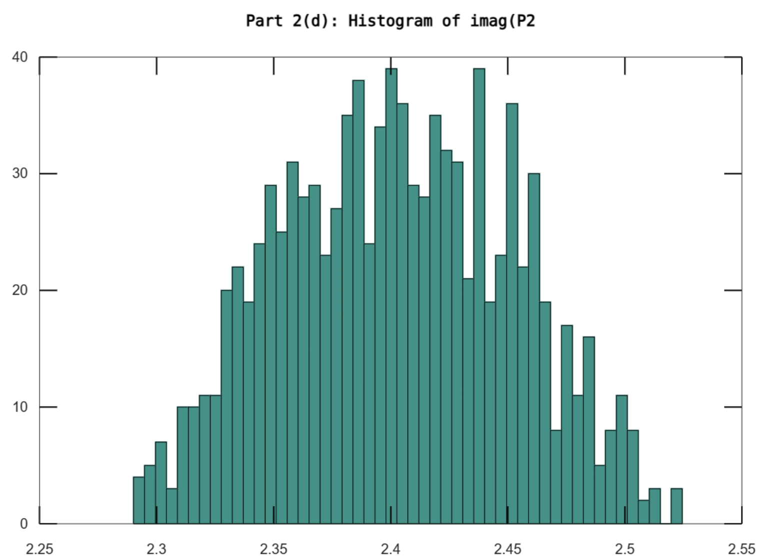
Histogram of Re{p2}

hist(rP2, 50)
mrP2 = mean(rP2);
srP2 = std(rP2);
title('Part 2(d): Histogram of real(P2)')



Histogram of Im{p2}

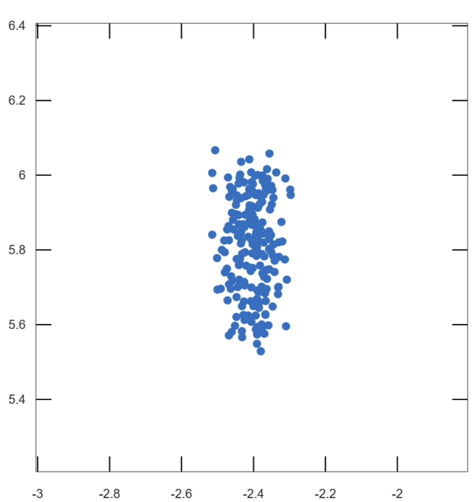
hist(iP2, 50)
miP2 = mean(iP2);
siP2 = std(iP2);
title('Part 2(d): Histogram of imag(P2')



e) Plot the first 200 of the 1000 calculated values of p1 as dots on the complex plane:

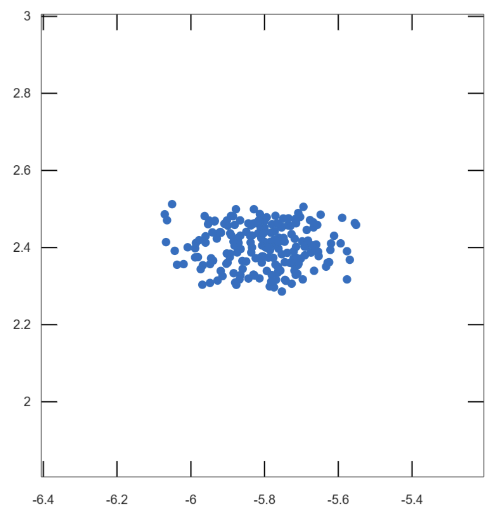
P1 = poles(:,1);
P2 = poles(:,2);

plot(P1(1:200),'.')
axis([real(p1)+[-0.6,0.6],imag(p1)+[-0.6,0.6]])
axis square



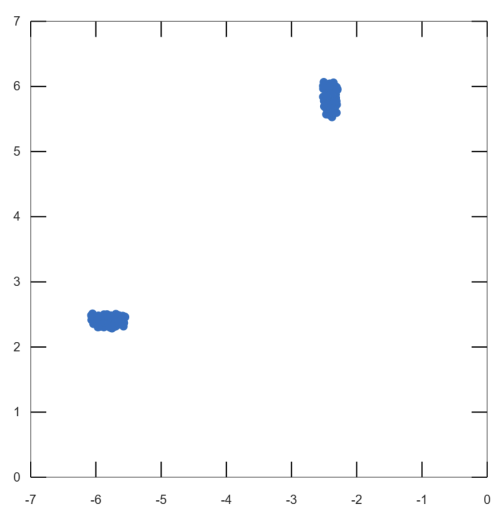
f) Plot the first 200 of the 1000 calculated values of p2 as dots on the complex plane:

```
plot(P2(1:200),'.')
axis([real(p2)+[-0.6,0.6],imag(p2)+[-0.6,0.6]])
axis square
```



g) Plot the first 200 of the 1000 calculated {p1, p2} pairs on the complex plane:

```
Poles1 = P1(1:200);
Poles2 = P2(1:200);
plot([Poles1(:);Poles2(:)],'.')
axis([-7,0,0,7])
axis square
```



h) From the list of perturbed pole values calculated in (d), estimate the mean values and standard deviations of Re(p1), Im(p1), Re(p2), and Im(p2).

```
f = logspace(-1,1,1e2);
[p1,p2,SF] = Poles(R1,R2,C1,C2,L1,L2);

fc = 5/100;
poles = [];
for n = 1:1000
        r1 = R1;
        r2 = R2;
        c1 = C1*(1+fc*(2*rand-1));
        c2 = c2*(1+fc*(2*rand-1));
         l1 = L1*(1+fc*(2*rand-1));
         l2 = L2*(1+fc*(2*rand-1));
        [P1,P2,SF] = Poles(r1,r2,c1,c2,l1,l2);
        poles = [poles; [P1,P2]];

                end

rP1 = real(poles(:,1));
iP1 = imag(poles(:,1));
rP2 = real(poles(:,2));
iP2 = imag(poles(:,2));

mean_real_P1 = mean(rP1)
sdev_real_P1 = std(rP1)

mean_imag_P1 = mean(iP1)
sdev_imag_P1 = std(iP1)

mean_real_P2 = mean(rP2)
sdev_real_P2 = std(rP2)

mean_imag_P2 = mean(iP2)
sdec_imag_P2 = std(iP2)
```

Answers:

mean_real_P1 = -2.4943
sdev_real_P1 = 0.3987

mean_imag_P1 = 5.5685
sdev_imag_P1 = 0.9440

mean_real_P2 = -5.7355
sdev_real_P2 = 0.3968

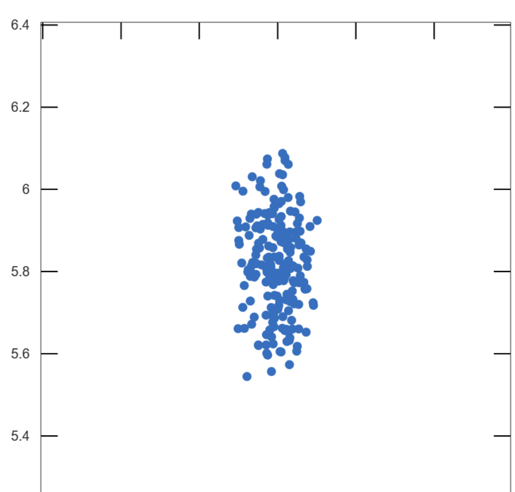
mean_imag_P2 = 2.2254
sdec_imag_P2 = 0.5813

i) From the statistics calculated in (h), generate 200 samples of the following "synthetic perturbed poles" whose PDF's are intended to mimic those of the actual perturbed poles found in (d)

```
X1 = randn(200, 1);
X2 = randn(200, 1);
X3 = randn(200, 1);
X4 = randn(200, 1);

pertp1 = (mrP1 + stdrP1 + X1) + j*(miP1 + stdrP1 + X2);
pertp2 = (mrP2 + stdrP2 + X4) + j*(miP2 + stdrP2 + X4);
```

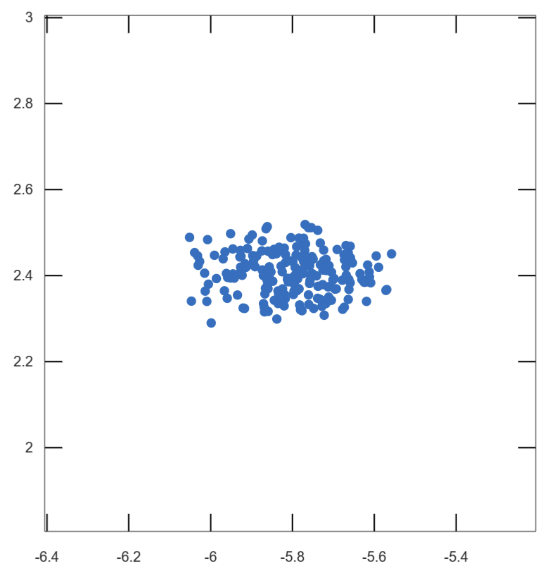
j) Repeat pole plots (e)-(g) for the 1000 synthetic perturbed poles $\{\tilde{p}1, \tilde{p}2\}$ that were generated in (i).

```
plot(pertpl(1:200), '.')
axis([real(p1)+[-0.6,0.6],imag(p1)+[-0.6,0.6]])
axis square
```



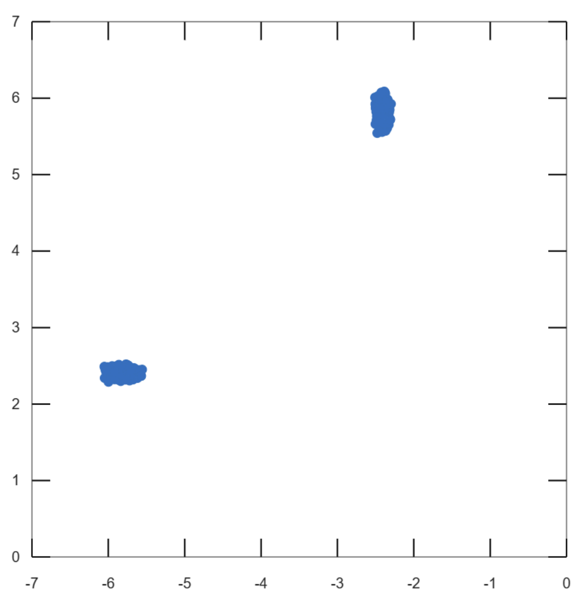
k)

```
plot(pertpl(1:200), '.')
axis([real(p2)+[-0.6,0.6],imag(p2)+[-0.6,0.6]])
axis square
```



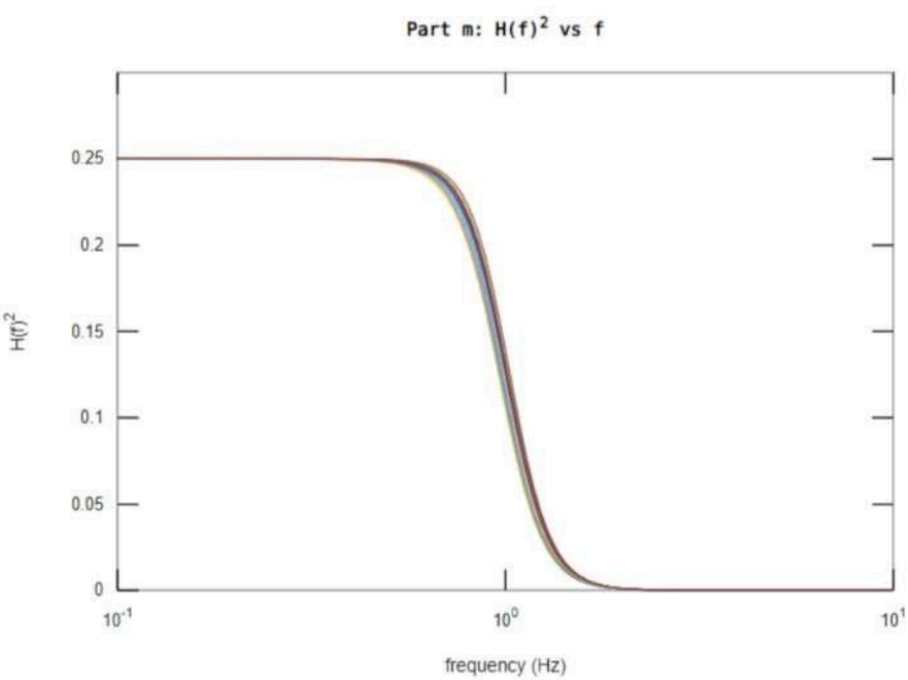
l)

```
plot([pertp1(:); pertp2(:)],'.')
axis([-7,0,0,7])
axis square
```



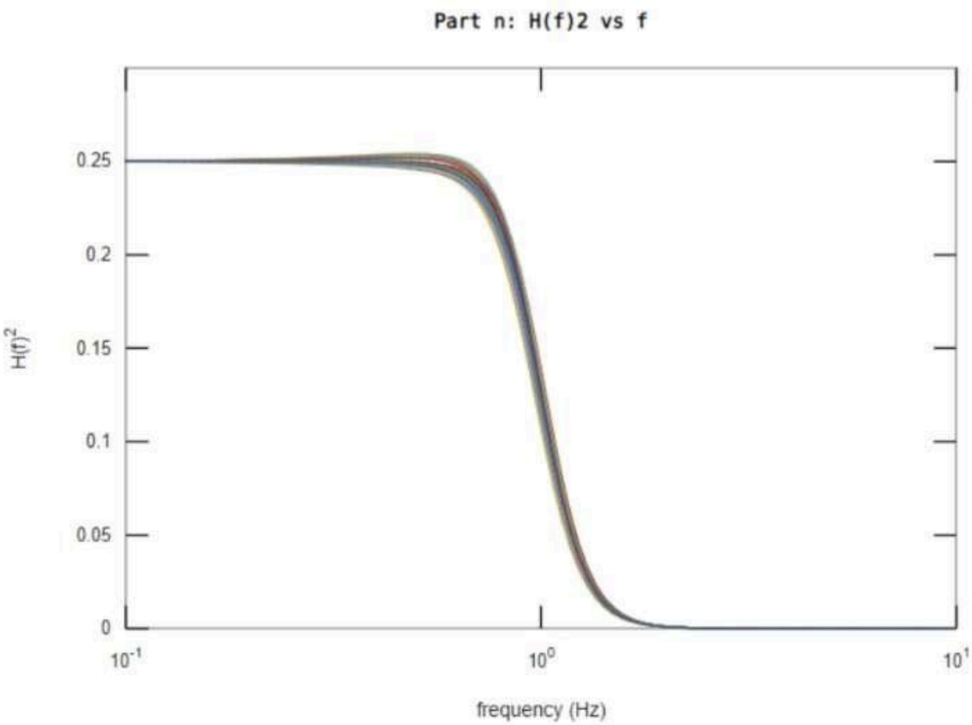
m) Create 25 superimposed plots of |H(f)|^2 vs. f on a single graph, using the provided function Frequency_response_from_poles and the first 25 out of the 1000 perturbed pole values that you found in (d).

```
    [p1, p2, SF] = Poles(R1, R2, C1, C2, L1, L2);

    percent = 25/1000;
    poles = [];
    for n = 1:25
    r1 = R1;
    r2 = R2;
    c1 = 0.1218*(1+percent*(2*rand-1));
    c2 = 0.2940*(1+percent*(2*rand-1));
    l1 = 0.2940*(1+percent*(2*rand-1));
    l2 = 0.1218*(1+percent*(2*rand-1));

    [P1, P2, SF] = Poles(r1, r2, c1, c2, l1, l2);
    Hmag2m = Frequency_response_from_poles(P1, P2, SF, f);
    semilogx(f, Hmag2m);
    hold on
end

    title('Part m: H(f)^2 vs f')
    xlabel('frequency (Hz)')
    ylabel('H(f)^2')
    axis([0.1, 10.0, 0.3])
```



n) Create 25 superimposed plots of |H(f)|^2 vs. f on a single graph, using the provided function Frequency_response_from_poles and the first 25 out of the 1000 synthetic pole values that you found in (i). (Since the scale factor was not calculated for the synthetic poles, use SF = 1/2 |p˜1p˜2|^2 in each case.)

SF = 0.5*(abs(pertp1(1:25).*pertp2(1:25))).^2;
Hmag2n = Frequency_response_from_poles(pertp1(1:25), pertp2(1:25), SF, f);
semilogx(f, Hmag2n)
title('Part n: H(f)2 vs f')



Conclusion:

In parts m and n, the two methods resulted in slightly different |H(f)|^2 vs f graphs. For part m, there is very little difference due to it being a uniform distribution, which each outcome is equally likely. However in part n, there is a larger difference in the graph due to a normal distribution being used. In a normal distribution, each outcome has its own probability occurrence, which causes more variation in the graph. Pole values are closer together when using a uniform distribution rather than a normal distribution.