1: 40 Points

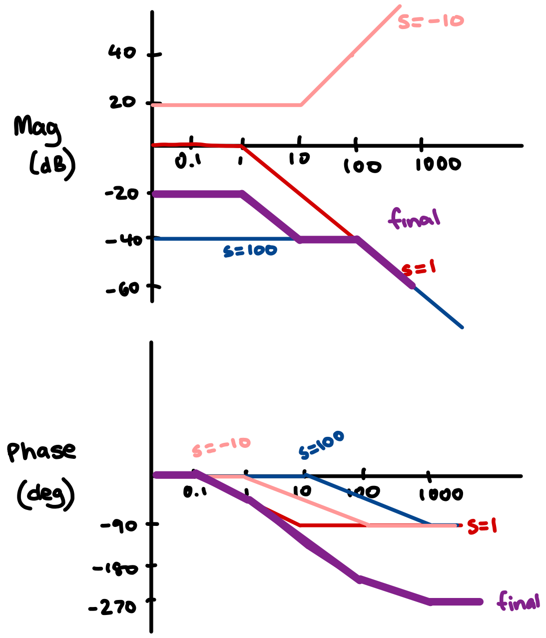
Given the system $G(s) = \frac{(-s+10)}{(s+1)(s+100)}$:

pole $s=-1$
pole $s=-100$
zero $s=10$

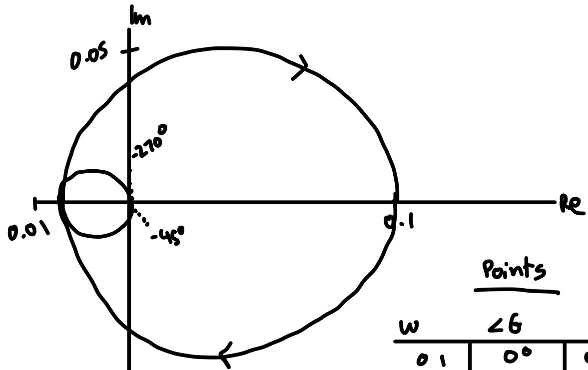
(a) Sketch the Bode plot using asymptotic approximations and the Nyquist plot based on your Bode plot.

*final answer is purple

Bode Plot



Nyquist Plot



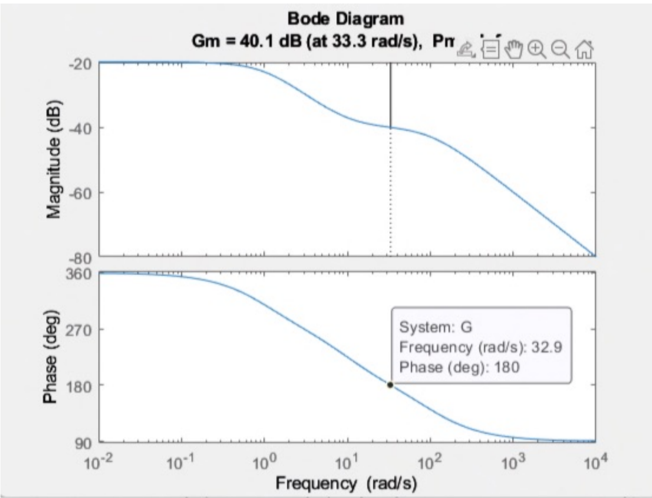
Points

w	∠G	|G|
0.1	0°	0.1
1	-45°	0.1
10	~-140°	0.01
100	~-210°	0.01

$-20 = 20\log_{10}(x)$

(b) The classical Ziegler-Nichols tuning rules for PI controller design of the form $K(s) = K_c\left(1 + \frac{1}{\tau_I s}\right)$ choose $K_c = \frac{K_u}{2.2}$ and $\tau_I = \frac{P_u}{1.2}$ where K_u is the maximum proportional controller gain (at marginal stability) and P_u is the period of oscillations at marginal stability. These can be found experimentally, but can also be obtained from a Bode plot by noting $K_u = \frac{1}{|G(j\omega_u)|}$ and $P_u = \frac{2\pi}{\omega_u}$ with ω_u the frequency at which $\angle G(j\omega) = -180^o$.

Generate the Bode plot using Matlab and use it to determine the controller $K(s)$ using these tuning rules.



$20\log_{10}(x) = -40.1\,dB$

$x = \frac{1}{101.158}$

$K_u = \frac{1}{\frac{1}{101.158}}$

$w_u = 32.9$

$P_u = \frac{2\pi}{32.9}$

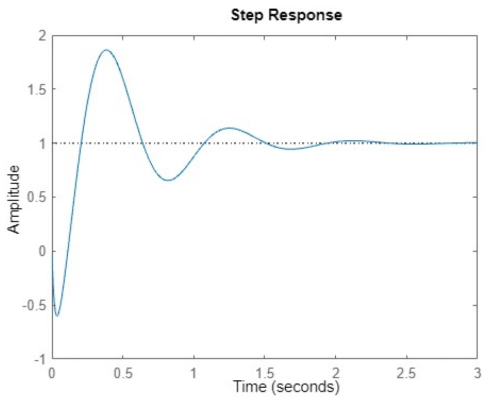
```
Ks =

  7.318 s + 45.98
  ---------------
     0.1591 s

Continuous-time transfer function.
```

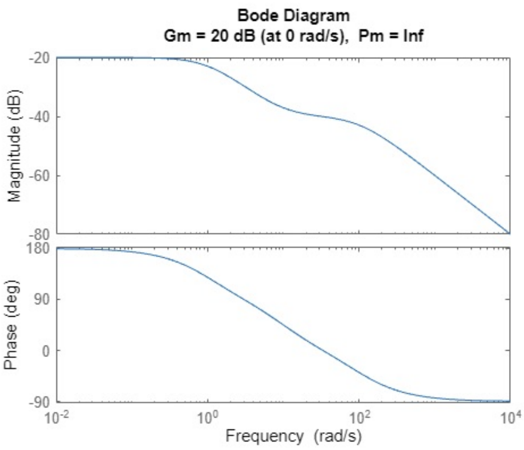
```
%Problem 1b
margin(G)
Ku = 1/(1/10^2.005);
Wu = 32.9;
Pu = (2*pi)/Wu;
s = tf('s');
Kc = Ku / 2.2;
tau = Pu/1.2;
Ks = Kc *(1 + (1/(tau*s)))
```

(c) Use Matlab to find the gain and phase margins of the system. Based on the phase margin, calculate the delay margin of the system. Simulate the step response of the closed loop system.



$DM = \frac{P_m}{\omega} = 0.0379$

(d) What if the numerator were written as $(s - 10)$, which is equivalent to multiplying the system by -1? Show that for the case with numerator $(s - 10)$ there does not exist a stabilizing PI controller (with positive coefficients / negative feedback), then design a proportional controller with 6 dB gain margin and simulate its performance.



$20\log_{10}(x) = 20\,dB$

$x = 10$

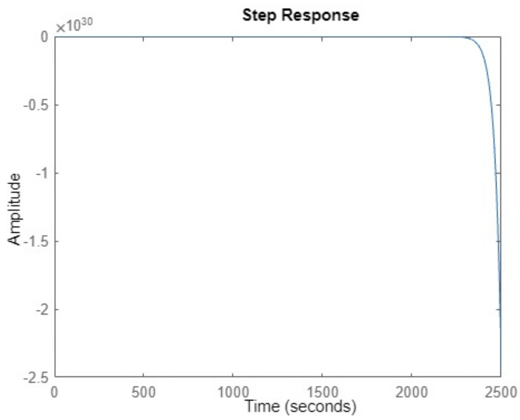
$K_u = \frac{1}{10}$

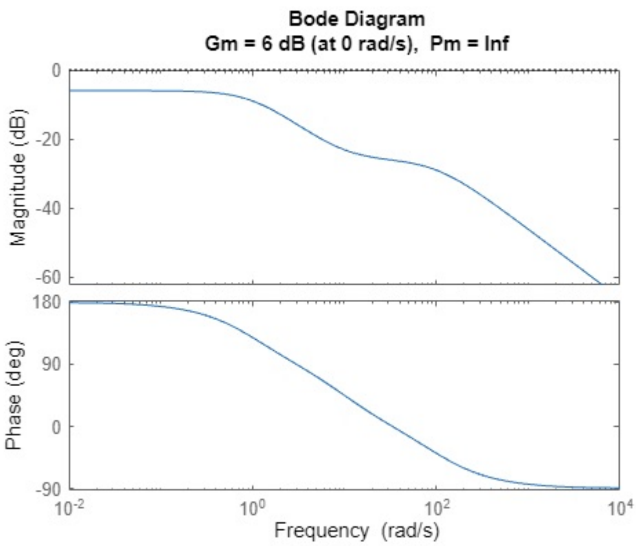
$\omega_u = 33.1$ rad/s

$P_u = \frac{2\pi}{\omega_u}$

We can see that there is no stabilizing PI through the fact that the step response doesn't stabilize.

P controller






```
%Problem 1d
new_sys = tf([1 -10], [1 101 100]);
margin(new_sys)

Ku = 1/10;
Wu = 33.1;
Pu = (2*pi)/Wu;
s = tf('s');
Kc = Ku / 2.2;
tau = Pu/1.2;
Ks = Kc *(1 + (1/(tau*s)))

T = feedback(new_sys * Ks,1);
step(T)
margin(T)


Ks = 5.01;
margin(new_sys * Ks)
```

$K_p = 5.01$

2: 40 points

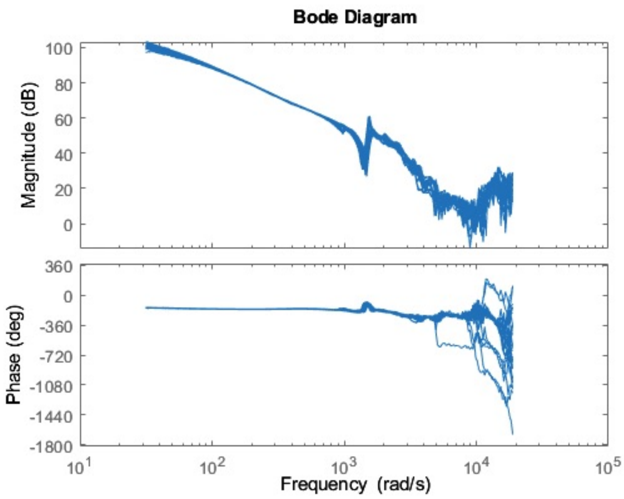
A local company has seen issues with some of its plant models changing over the device's operating temperature. The file *respdata.mat* contains experimental frequency response data for a precision positioning system for various operating conditions. This question will ask you to solve a more open ended (and realistic) design problem, where use of loopshaping tools can help design a controller that works for ALL measured plants. A "good" controller for this system will achieve an open loop crossover frequency of 800 Hz, a phase margin of at least 40 degrees, and a gain margin of at least 6 dB.

(a) Design a feedback controller that includes integral action and meets the specifications. Use margin to demonstrate that the goals were achieved (margin with output arguments can give you the margin for each measured plant).

GM	PM	Wc
3.471	65.229	2383.991
3.235	66.377	2370.208
3.674	60.598	2395.300
3.251	57.498	2379.465
3.207	58.301	2377.838
3.351	57.619	2381.374
3.239	55.279	2329.677
3.254	62.363	2338.411
3.350	62.286	2344.724
4.146	Inf	2530.330
4.171	Inf	2537.897
4.359	Inf	2580.080
4.122	Inf	2572.594
4.098	Inf	2616.597
4.024	Inf	2615.760
4.081	Inf	2551.986
3.874	Inf	2577.725
4.504	Inf	2659.338
3.180	Inf	2631.552
3.126	Inf	2631.673
3.211	Inf	2618.499
3.664	Inf	2635.192
3.680	Inf	2616.372
3.447	Inf	2643.969
3.539	Inf	2618.051
3.705	Inf	2561.711
3.302	Inf	2622.218
Min Value:	Min Value:	Min Value:
3.126	55.279	2329.677
9.899		

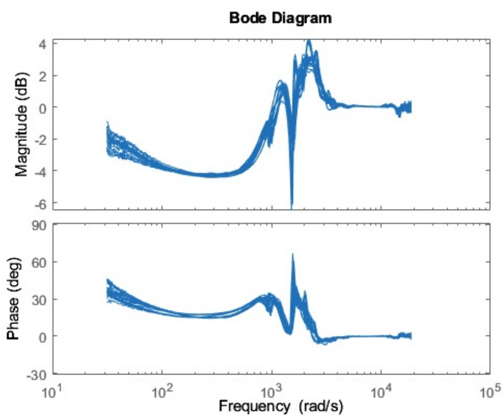
dB

```
wc = 800*2*pi;
s = tf('s');
controller_pi = (s + wc/10)/s;
bode(sys * controller_pi)
PM = 40;
[Gm, Pm, Wc] = margin(sys*controller_pi)
phase = -240;
new_angle = PM + (-180 - phase);
z = 50;
p = 1000;
compensator_lead = tf([1 z],[1 p]);
K = 1/600;
bode((sys * controller_pi * compensator_lead * compensator_lead* K))
[Gm, Pm, Wc] = margin(sys * controller_pi * compensator_lead * compensator_lead* K)
```



(b) Plot the sensitivity and complementary sensitivity function for your design. What are their peak values over frequency?

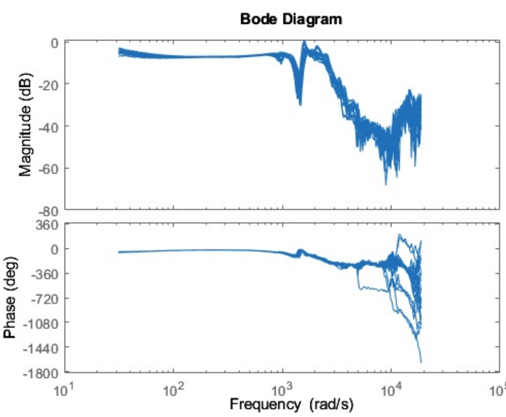
Sensitivity



Magnitude: 4.1 dB → 1.603

Phase: 66.6°

Complementary Sensitivity

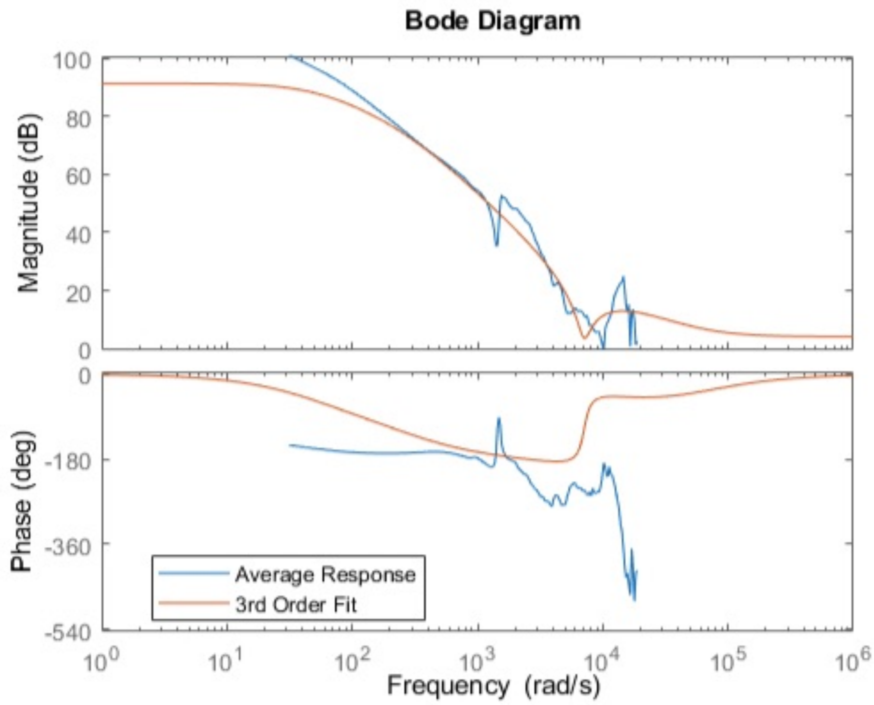


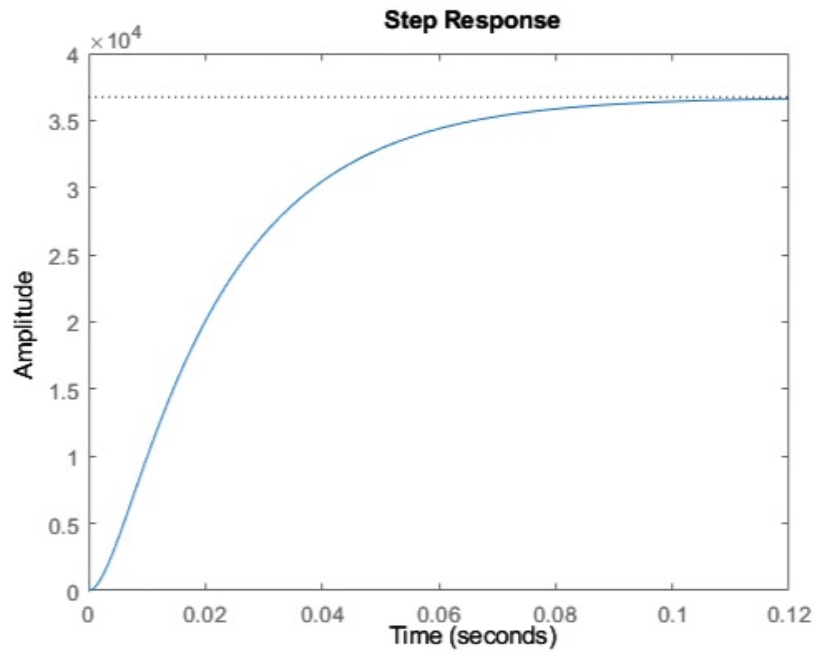
Magnitude: 1.125dB → 1.138

Phase: 162.46°

```
%Problem 2b
S = 1/(1+sys*control);
T = 1-S;
bode(S)
bode(T)
```

(c) FRD-based design can be quite effective in practice, but Matlab will not allow you to use an FRD model in simulink or to simulate performance in Matlab. For simulation we will need to find a state space approximation to the data. Read about the command *fitmagfrd*, and use that command to fit an approximation to *average* frequency response. Plot the Bode plot of the FRD and its approximation on the same axes, and use the approximation to plot the step response.






```
%Problem 2c
sumsys = 0;
for i = 1:27
    sumsys = sys(1,1,i,1) + sumsys;
end
avgfreqres = sumsys / 27;
approx = fitmagfrd(avgfreqres,3);
bode(avgfreqres, approx)
legend("Average Response", "3rd Order Fit", 'Location', 'best')

step(approx)
```

(d) Another alternative is to use LQR with an integral term to design a state space controller for the system. See this example from the Matlab documentation which will complete the design for you: `https://www.mathworks.com/help/control/getstart/design-an-lqg-servo-controller.html`. Unlike the example there you will want to use the '1dof' option in *lqgtrack* to match the design you have from Part a. When following that example choose $G = I$ and $H = 0$ of the appropriate dimensions. For this design we will let $Q = C^T C$ for the non-integrator states and $Q = 1000$ for the integrator state and you will change the level of R to adjust the crossover frequency. For the Kalman filter we will assume that the measurement is good, so choose $V = 1e-6$ and $W = I$. Perform the design for the nominal system, varying R until the crossover frequency is ≈ 800 Hz, and plot the step response against the response for Part (c). Will this stabilize *all* of the plant models shown?

R = 200000

Pm = 41.7657

W_{cp} = 824.92

GM = 7.33 dB