

ECE 365

Spring 2006

Matlab Project 3:

Bode and Nyquist  
Techniques

John O'Flaherty

### 1] Problem statement:

It is desired to use Nyquist and Bode plots to characterize this system over a range of gains  $k$ :

$G(s)H(s) = \frac{k}{s(s+1)(1+10s)}$ , specifically, to find the value of  $k$  that will make it a perfect oscillator, and

to further characterize it with a 0.2 s delay time incorporated,

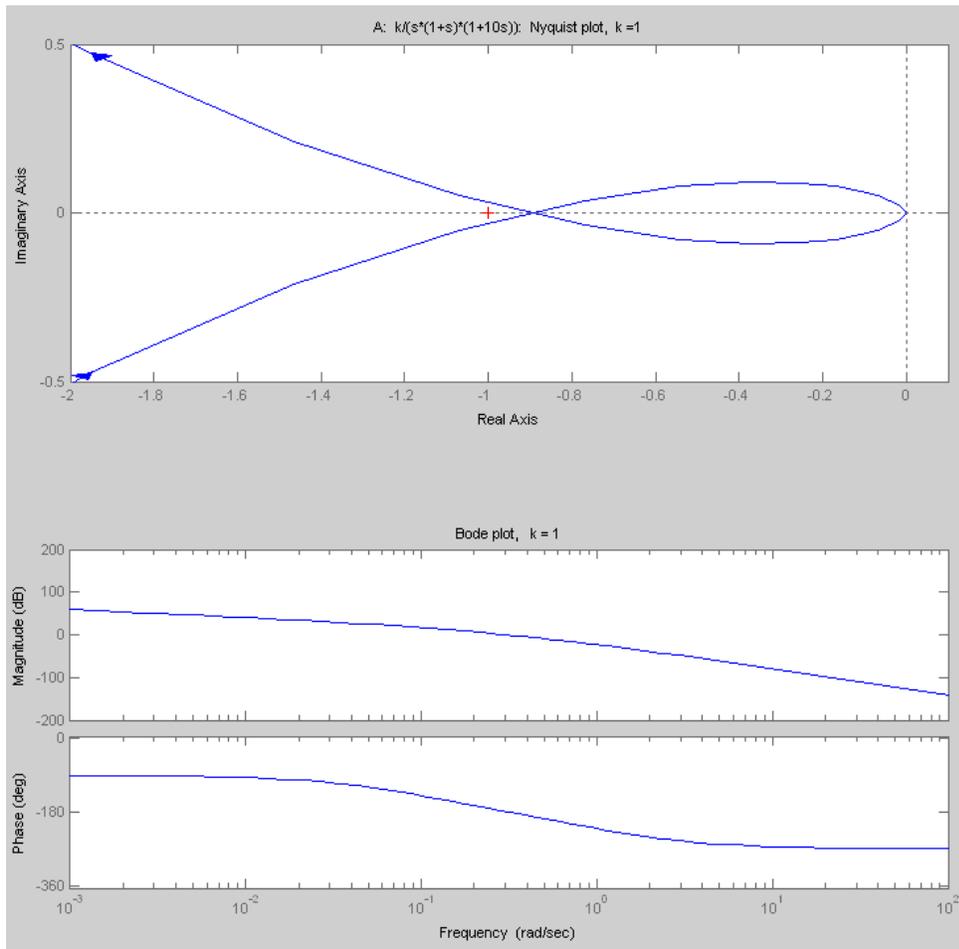
$G(s)H(s) = \frac{ke^{-0.5s}}{s(s+1)(1+10s)}$ , to see what new value of  $k$  will make it a perfect oscillator.

**2] Approach:** The Bode and Nyquist plots for  $k = 1$  are plotted (a). Then, Matlab is used to discover the value  $k = k_c$ , the value that will make it oscillate (b). This value is confirmed mathematically. The Bode and Nyquist plots are repeated for this critical value. Then plots for half the critical gain value are plotted (c), and the gain and phase crossover frequencies and gain and phase margin are found. Finally, the transport lag of 0.5 seconds delay ( $e^{-s/2}$ ) is included, and the critical value of gain to make the system oscillate is found again (d).

### 3] Analysis and results:

**a) Nyquist and Bode plots for  $k = 1$ .** The transfer function is entered into Matlab:

```
s=tf('s'); gh=1/(s*(1+s)*(1+10*s)); The Bode and Nyquist plots are here.
```



**b)** Find  $k_c$ , the value of  $k$  to make the system a perfect oscillator. Then use  $k_c/2$  to find the gain margin. Oscillation will occur at a frequency where the transfer function magnitude is -1 and the phase is  $-180^\circ$ . This is found by setting the imaginary part of the system response to zero and solving for  $\omega$ . Then the magnitude of the response at that frequency is found, and the gain to produced oscillation is the inverse

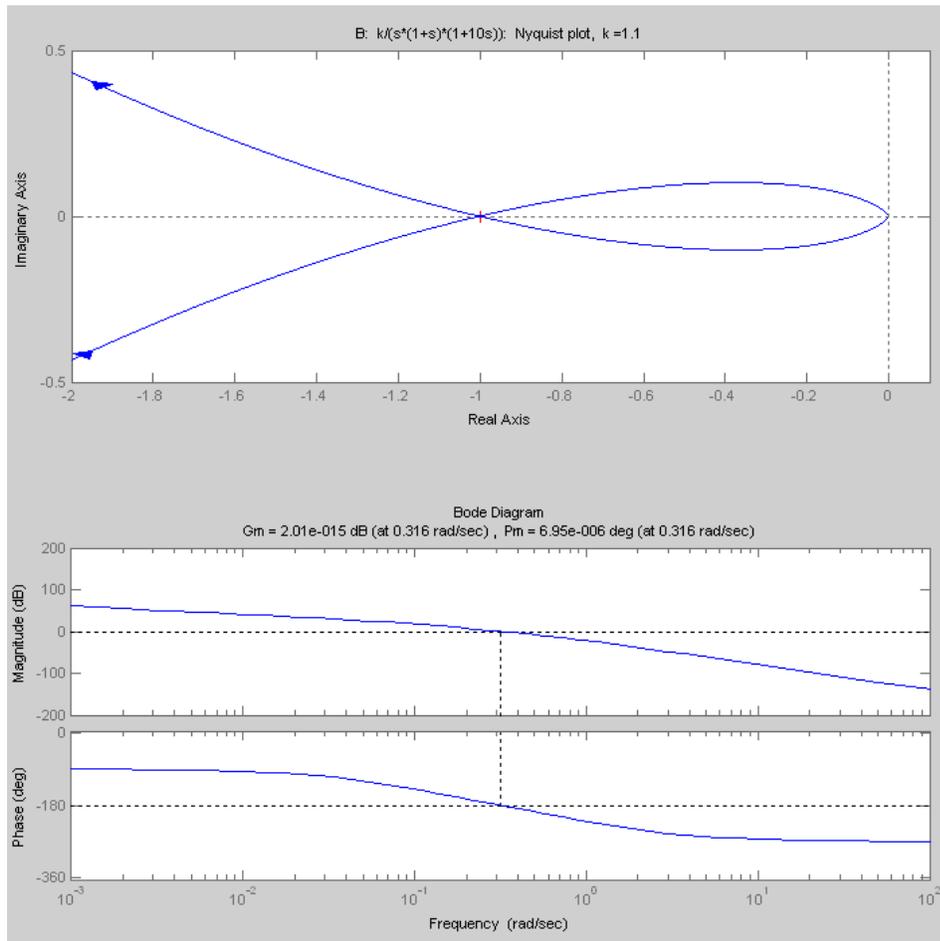
of that value. 
$$\frac{1}{s(s+1)(1+10s)} = \frac{1}{j\omega(j\omega+1)(1+10j\omega)}$$

$$\text{Im}(j\omega(j\omega+1)(1+10j\omega)) = 0 \Rightarrow -10\omega^2 + 1 = 0 \Rightarrow \omega = \pm \frac{\sqrt{10}}{10} = \pm 0.31623\dots$$

Finding the magnitude of the transfer function with this value for  $j\omega$  gives 0.9090909..., the inverse of which is **1.1**. This is the value of  $k_c$ . This value is confirmed by use of the Matlab nyquist function:

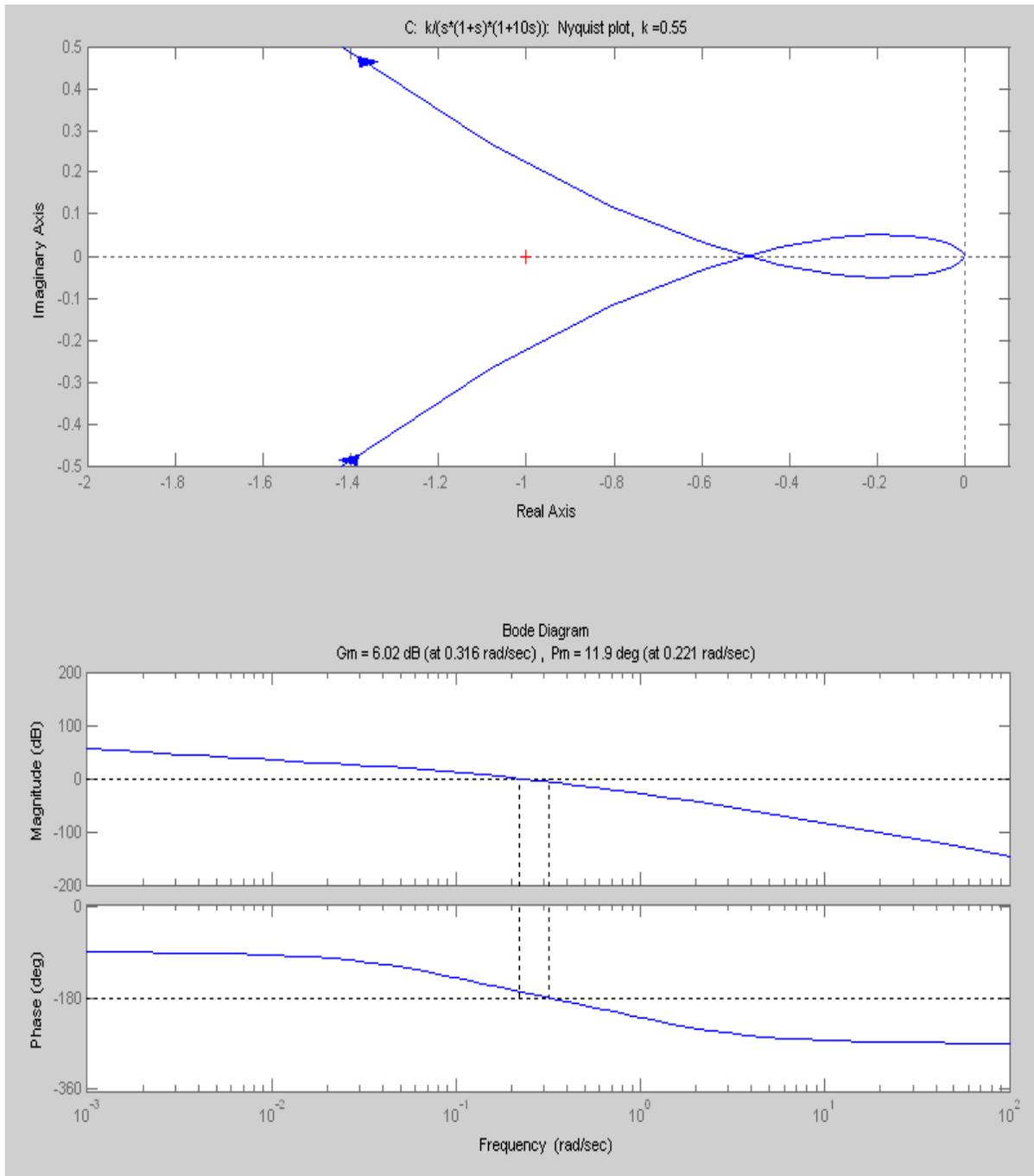
```
[re, im, w]=nyquist(gh, .31:.000001:.33);
w(find(abs(im)<0.000001))           ans = 0.31
kc=abs(1/re(find(abs(im)<0.000001)))  kc = 1.1000
```

These are the Bode and Nyquist plots for  $k_c = 1.1$ :



At the critical value, the gain and phase margins are both zero. Now, setting the gain to  $k_c/2$  means  $k$  is 0.55, and the **gain margin** is obviously **6.02 db**, since that is a factor of two.

**c)** *Verify the findings in part b) by Bode and Nyquist plots with  $k_c/2$ , finding the gain and phase crossover frequencies and the gain and phase margins of the system.*



The gain and phase margins and frequencies are shown in the caption of the drawing, produced by using the command 'margin' instead of 'bode'. They can also be found from the command

```
[Gm, Pm, Wcg, Wcp] = margin(gh*.55)
```

```
Gm = 2.0000 Pm = 11.8522 Wcg = 0.3162 Wcp = 0.2212
```

That is, **gain margin = 6.02 db**, **phase margin = 11.8522°**, **gain crossover frequency =  $\omega_{cg} = 0.3162$  rad/s**, and **phase crossover frequency =  $\omega_{cp} = 0.2212$  rad/s**.

**d)** With a transport lag of 0.5 seconds, find the new critical value of  $k_c$  to make the system oscillate.

First, the transport lag is set for the system by the Matlab statement

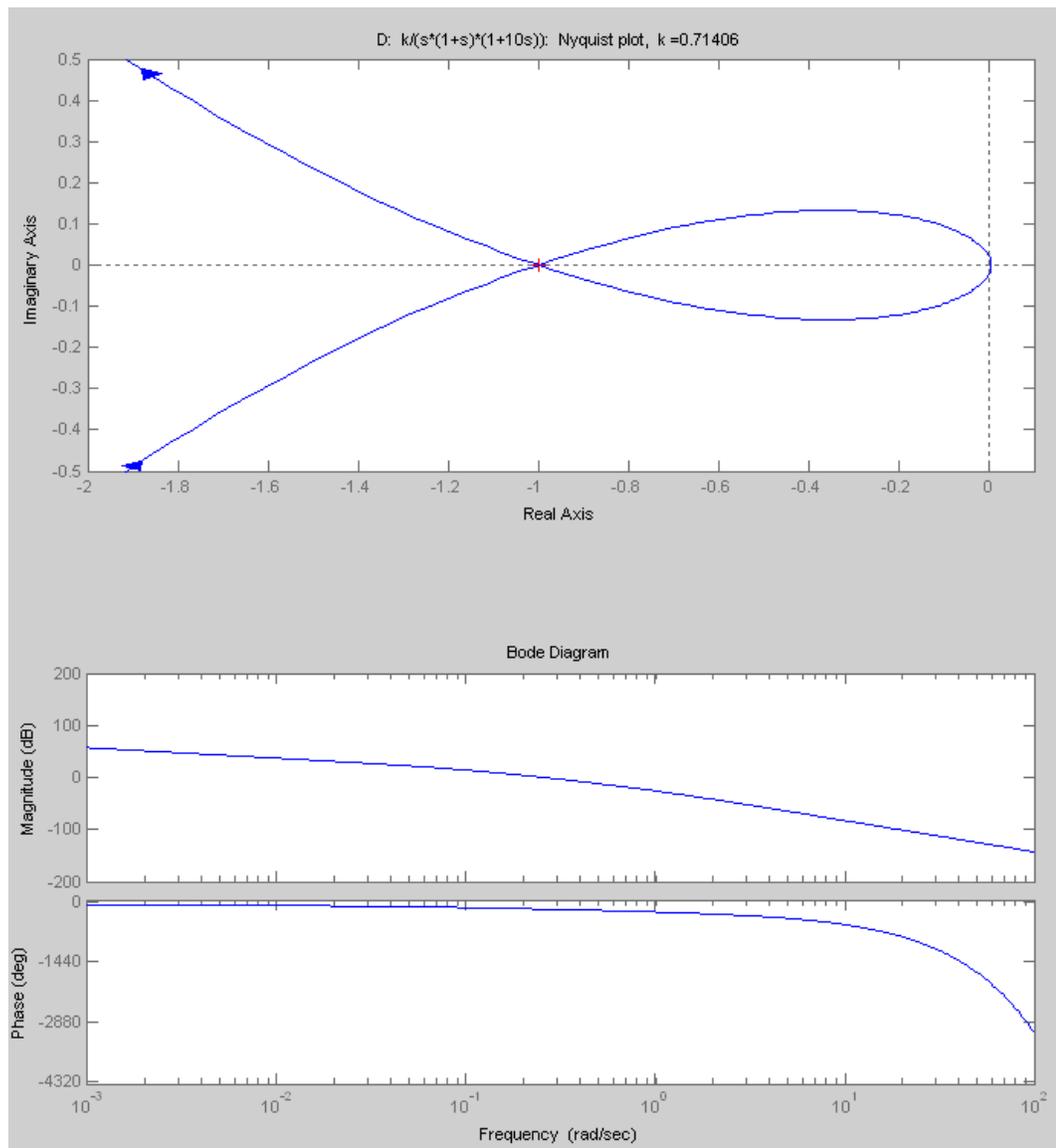
```
set(gh, 'iodelay', 0.5); which produces the output
```

$$\exp(-0.5*s) * \frac{1}{10 s^3 + 11 s^2 + s}$$

A numerical solution in a CAS is done, and the **critical frequency** is found to be **0.2537572 rad/s** The value of  $k_c$  found from this frequency is **0.7140598**. These values are confirmed by Matlab:

```
[re,im,w]=nyquist(ght, .253:.000001:.254);
w(find(abs(im)<0.000001))           ans = 0.2538
kt=abs(1/re(find(abs(im)<0.000001))) ans = 0.7141
```

The Nyquist and Bode plots are here. The notable thing about the Bode plot is the very high levels of phase lag at high frequencies. For a time delay, the phase lag increases without limit with frequency.



#### 4] Conclusions:

The Nyquist plot facility of Matlab is very convenient, but care must be taken when evaluating graphic results. A simple Nyquist plot of the original system would indicate a gain of  $k = 1.14$  to make the system oscillate, due to the relatively low resolution of the plot. Repeating the plot with a specific frequency range argument sharpens the results so the correct value of 1.1 is obtained.

The  $\frac{1}{2}$  second transport delay has a considerable effect on the system, reducing the gain to produce oscillation from 1.1 to 0.7. This means that steady-state errors for a stable system would be  $1.1 / 0.7$  or 1.5 times greater for the system with transport delay.

#### 5] Appendix:

##### Matlab code used for the plots and calculations:

```
s=tf('s');
gh=1/(s*(1+s)*(1+10*s));
k=1;

% Part a) Nyquist and Bode plots for k = 1
figure
subplot(2,2,[1 2]);
nyquist(gh);
title(['A: k/(s*(1+s)*(1+10s)): Nyquist plot, k =',num2str(k)])
axis([-2 .1 -.5 .5]);
subplot(2,2,[3 4]);
bode(gh);
title('Bode plot, k = 1')

% Part b) Use Nyquist function to isolate the frequency of the
% real axis crossing and the gain required.
[re,im,w]=nyquist(gh,.31:.000001:.33);
w(find(abs(im)<0.000001))
% the inverse of the real axis crossing is the value of k needed to
% get a crossing at exactly -1.000, the gain needed for oscillation.
kt=abs(1/re(find(abs(im)<0.000001)))

k=1.1;
% Nyquist and Bode plots for k = critical value
figure
subplot(2,2,[1 2]);
nyquist(gh*k,logspace(-1,.5,100000));
title(['B: k/(s*(1+s)*(1+10s)): Nyquist plot, k =',num2str(k)])
axis([-2 .1 -.5 .5]);
subplot(2,2,[3 4]);
margin(gh*k);
```

```

% Part c) Nyquist and Bode plots for k = (crit val/2)
k=k/2;
figure
subplot(2,2,[1 2]);
nyquist(gh*k);
title(['C: k/(s*(1+s)*(1+10s)): Nyquist plot, k =',num2str(k)])
axis([-2 .1 -.5 .5]);
subplot(2,2,[3 4]);
margin(gh*k);

% Part d) use the Nyquist function to isolate frequency of the
% real axis crossing and the gain needed for oscillation.
ght=gh; % use new variable for convenience
set(ght,'iodelay',0.5); % set the transport delay parameter to 0.2 seconds
% Find the critical frequency and gain.
[re,im,w]=nyquist(ght,.253:.000001:.254);
w(find(abs(im)<0.000001))
kt=abs(1/re(find(abs(im)<0.000001)))
% Nyquist and Bode plots for system with transport lag
figure
subplot(2,2,[1 2]);
nyquist(ght*kt,logspace(-2,2,500));
title(['D: k/(s*(1+s)*(1+10s)): Nyquist plot, k =',num2str(kt)])
axis([-2 .1 -.5 .5]);
subplot(2,2,[3 4]);
bode(ght*kt,logspace(-3,2,500));

kt2=kt/2
% Repeat Nyquist and Bode plots for system with transport lag but with
% k=kc/2
figure
subplot(2,2,[1 2]);
nyquist(ght*kt2);
title(['D-2: exp(-0.5s)*k/(s*(1+s)*(1+10s)): Nyquist plot, k =
',num2str(kt1)])
axis([-2 .1 -.5 .5]);
subplot(2,2,[3 4]);
margin(ght*kt2);

```