

Tips ☐Effects ☐

Search Control Tutorials



Carnegie Mellon

DETROIT MERCY

ABOUT

BASICS

HARDWARE

INDEX

NEXT ►



INTRODUCTION



CRUISE CONTROL



MOTOR SPEED

## SYSTEM ◀

MODELING

ANALYSIS

## CONTROL ◀

PID

ROOT LOCUS

FREQUENCY

STATE-SPACE

DIGITAL

## SIMULINK ◀

MODELING

CONTROL

SIMSCAPE

## Introduction: Frequency Domain Methods for Controller Design

The frequency response method of controller design may be less intuitive than other methods you have studied previously. However, it has certain advantages, especially in real-life situations such as modeling transfer functions from physical data. In this tutorial, we will see how we can use the open-loop frequency response of a system to predict its closed-loop time response behavior.

Key MATLAB commands used in this tutorial are: [bode](#) , [nyquist](#) , [margin](#) , [lsim](#) , [step](#) , [feedback](#) , [controlSystemDesigner](#)

### Related Tutorial Links

- [Freq Resp Activity](#)
- [Lead/Lag Freq Resp](#)

### Related External Links

- [Bode MATLAB Video](#)

- [Freq Resp Video](#)

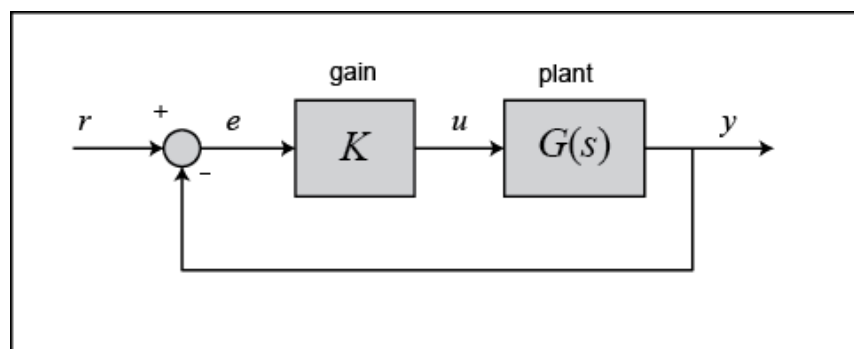
## Contents

- [Gain and Phase Margin](#)
- [Nyquist Diagram](#)
- [The Cauchy Criterion](#)
- [Closed-Loop Performance from Bode Plots](#)
- [Closed-Loop Stability from the Nyquist Diagram](#)

## Gain and Phase Margin

Recall from the [Introduction: System Analysis](#) page that the frequency response of a system consists of evaluating how a sinusoidal input to a system is scaled and shifted by the system. The manner in which the scaling and shifting of the sinusoidal output changes as a function of frequency provides useful information about the system's time response. One aspect, in particular, that a system's frequency response is used for determining is a system's "robustness." For example, how close is the system to becoming unstable? Here we use two quantities, **gain margin** and **phase margin** to indicate the margin the system has before it would go unstable.

Consider the following unity feedback system:

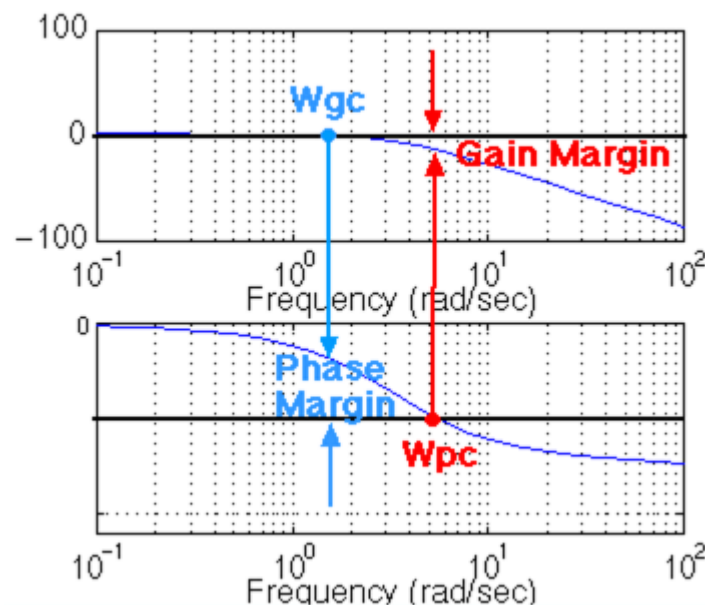


where  $K$  is a variable (constant) gain and  $G(s)$  is the plant under consideration. The gain margin is defined as the change in open-loop gain required to make the closed-loop system unstable. Systems with greater gain margins can withstand greater changes in system parameters before becoming unstable in closed-loop.

The phase margin is defined as the change in open-loop phase shift required to make the closed-loop system unstable. The phase margin also measures the system's tolerance to time delay. If there is a time delay greater than  $PM/\omega_{gc}$  in the loop (where  $\omega_{gc}$  is the frequency in rad/sec where the magnitude is 0 dB and PM is the phase margin converted to radians), the closed-loop system will become unstable. The time delay,  $\tau_d$ , can be thought of as an extra block in the forward path of the block diagram that adds phase lag to the system, but has no effect on the gain. That is, a time delay can be represented as a block with magnitude of 1 and phase  $-\omega\tau_d$ .

For now, we won't worry about where all this comes from and rather will concentrate on identifying the gain and phase margins on a Bode plot.

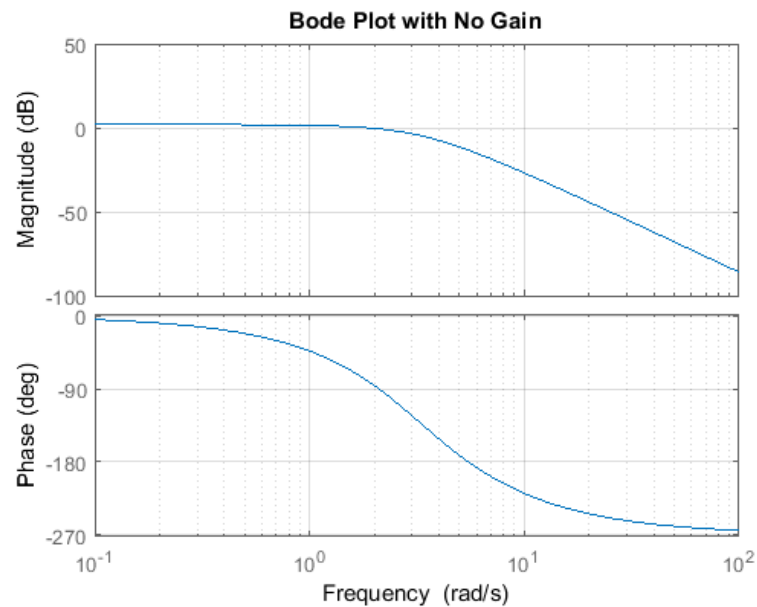
The closed-loop system's phase margin is the additional amount of phase lag that is required for the open-loop system's phase to reach -180 degrees at the frequency where the open-loop system's magnitude is 0 dB (the gain crossover frequency,  $\omega_{gc}$ ). Likewise, the gain margin is the additional amount of gain (usually in dB) required for the open-loop system's magnitude to reach 0 dB at the frequency where the open-loop system's phase equals -180 degrees (the phase crossover frequency,  $\omega_{pc}$ ).



One nice thing about the phase margin is that you don't need to replot the Bode diagram in order to find the new phase margin when changing the loop gain. If you recall, adding gain only shifts the magnitude plot up (or down). This is equivalent to changing the y-axis on the magnitude plot. Finding the phase margin is simply a matter of finding the new gain crossover frequency and

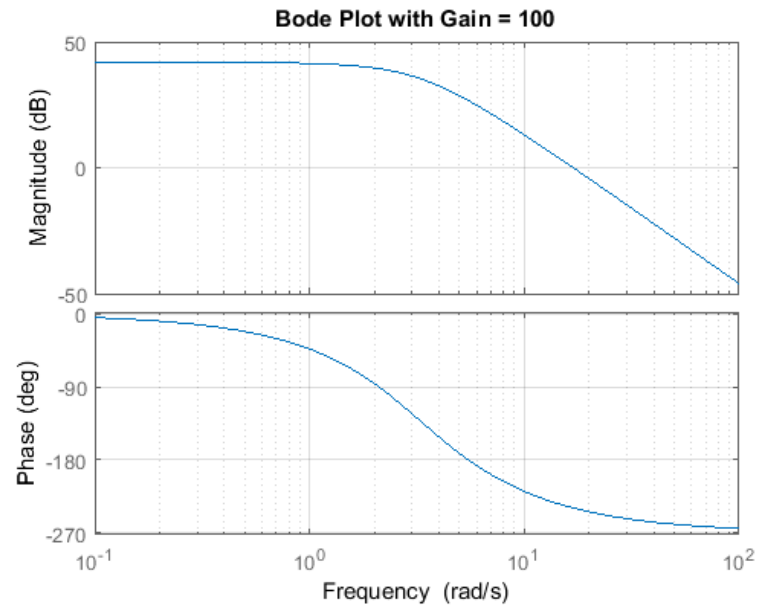
reading off the phase margin. For example, suppose you employed the following commands to generate the Bode plot shown below:

```
s = tf('s');  
G = 50/(s^3 + 9*s^2 + 30*s + 40);  
  
bode(G)  
grid on  
title('Bode Plot with No Gain')
```



You should see that the phase margin is about 100 degrees. Now suppose you added a gain of 100, by entering the command `bode(100*G)`. You should get the following plot:

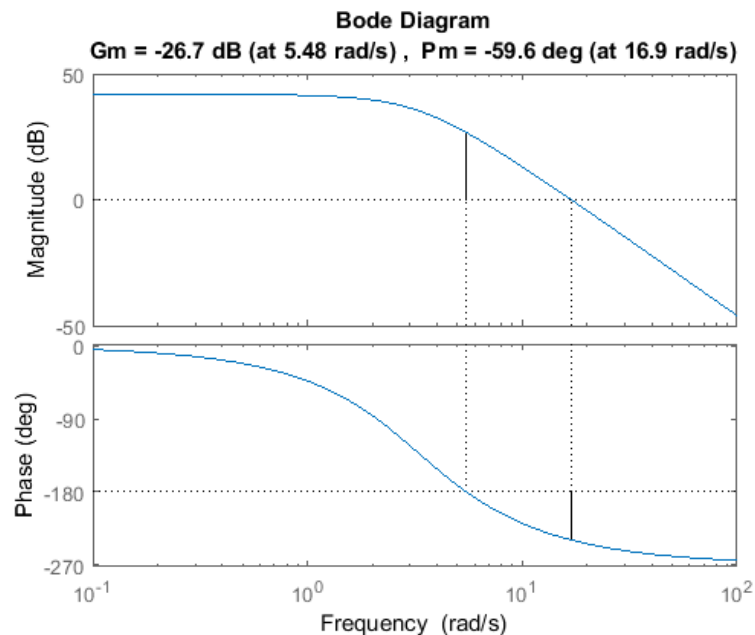
```
bode(100*G)  
grid on  
title('Bode Plot with Gain = 100')
```



As you can see, the phase plot is exactly the same as before, and the magnitude plot is shifted up by 40 dB (gain of 100). The phase margin is now about -60 degrees. This same result could be achieved if the y-axis of the magnitude plot was shifted down 40 dB. Try this, look at the first Bode plot, find where the curve crosses the -40 dB line, and read off the phase margin. It should be about -60 degrees, the same as the second Bode plot.

We can have MATLAB calculate and display the gain and phase margins using the `margin(G)` command. This command returns the gain and phase margins, the gain and phase crossover frequencies, and a graphical representation of these quantities on the Bode plot. See below, for an example:

```
margin(100*G)
```



### Bandwidth Frequency

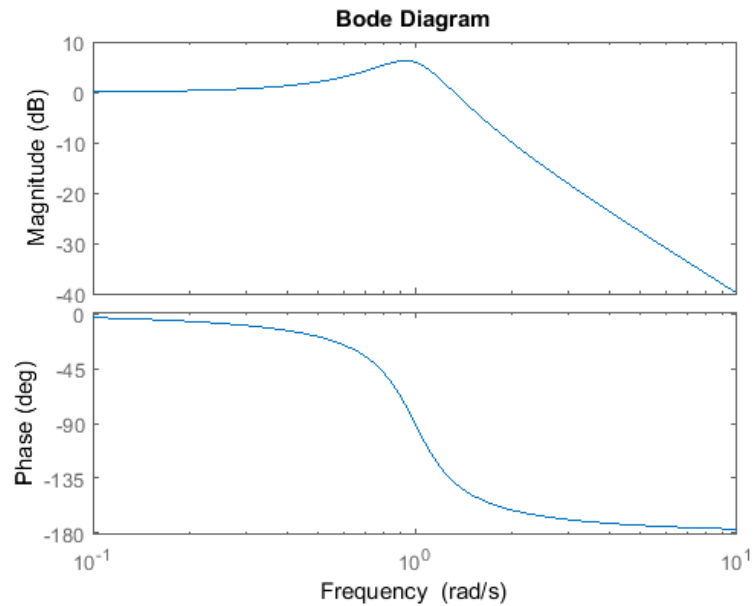
The bandwidth frequency is defined as the frequency at which the closed-loop magnitude drops 3 dB below its magnitude at DC (magnitude as the frequency approaches 0). However, when we design via frequency response, we are interested in predicting the closed-loop behavior from the open-loop response. Therefore, we will use a second-order system approximation and say that the bandwidth frequency equals the frequency at which the open-loop magnitude response is between -6 and -7.5 dB, assuming the open-loop phase response is between -135 deg and -225 deg. For a complete derivation of this approximation, consult your textbook.

In order to illustrate the importance of the bandwidth frequency, we will show how the output changes with different input frequencies. We will find that sinusoidal inputs with frequency less than  $\omega_{bw}$  (the bandwidth frequency) are tracked "reasonably well" by the system. Sinusoidal inputs with frequency greater than  $\omega_{bw}$  are attenuated (in magnitude) by a factor of 0.707 or greater (and are also shifted in phase).

Let's say we have the following closed-loop transfer function representing a system:

$$G(s) = \frac{1}{s^2 + 0.5s + 1} \quad (1)$$

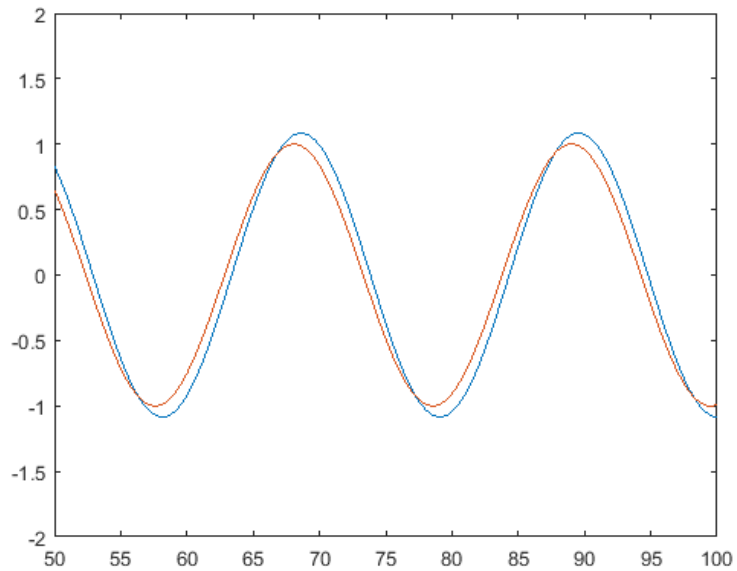
```
G = 1/(s^2 + 0.5*s + 1);
bode(G)
```



Since this is the closed-loop transfer function, our bandwidth frequency will be the frequency corresponding to a gain of -3 dB. Looking at the plot, we find that it is approximately 1.4 rad/s. We can also read off the plot that for an input frequency of 0.3 radians, the output sinusoid should have a magnitude about one and the phase should be shifted by perhaps a few degrees (behind the input). For an input frequency of 3 rad/sec, the output magnitude should be about -20 dB (or 1/10 as large as the input) and the phase should be nearly -180 (almost exactly out-of-phase). We can use the `lsim` command to simulate the response of the system to sinusoidal inputs.

First, consider a sinusoidal input with a **frequency lower than  $\omega_{bw}$** . We must also keep in mind that we want to view the steady state response. Therefore, we will modify the axes in order to see the steady state response clearly (ignoring the transient response).

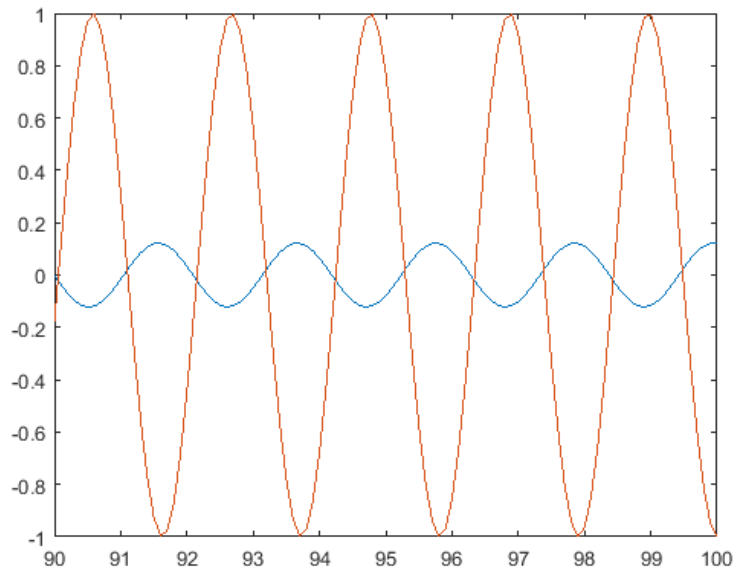
```
G = 1/(s^2 + 0.5*s + 1);
w = 0.3;
t = 0:0.1:100;
u = sin(w*t);
[y,t] = lsim(G,u,t);
plot(t,y,t,u)
axis([50 100 -2 2])
```



Note that the output (blue) tracks the input (red) fairly well; it is perhaps a few degrees behind the input as expected. However, if we set the frequency of the input **higher than the bandwidth frequency** for the system, we get a very distorted response (with respect to the input):

```
G = 1/(s^2 + 0.5*s + 1);  
w = 3;  
t = 0:0.1:100;  
u = sin(w*t);  
[y,t] = lsim(G,u,t);  
plot(t,y,t,u)  
axis([90 100 -1 1])
```





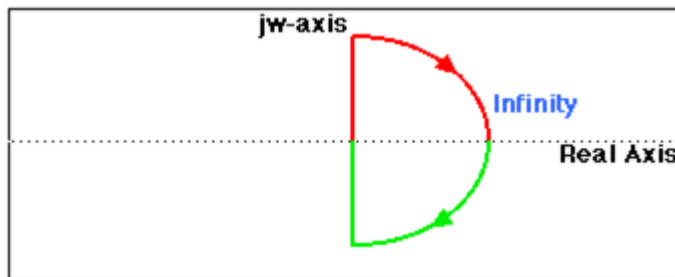
Again, note that the magnitude is about 1/10 that of the input, as predicted, and that it is almost exactly out of phase (180 degrees behind) the input. Feel free to experiment and view the response for several different frequencies  $\omega$ , and see if they match the Bode plot.

## Nyquist Diagram

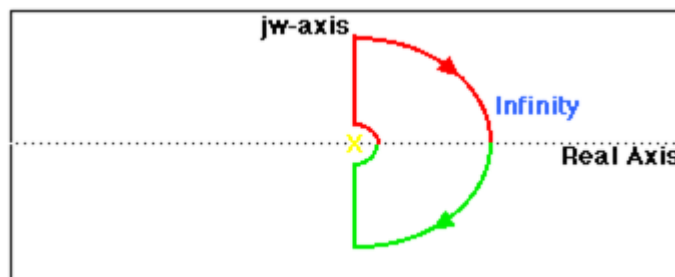
The Nyquist diagram allows us to predict the stability and performance of a closed-loop system by observing its open-loop behavior. The Nyquist criterion can be used for design purposes regardless of open-loop stability (remember that the Bode design methods assume that the system is stable in open-loop). Therefore, we use this criterion to determine closed-loop stability when the Bode plots display confusing information.

**Note: The MATLAB `nyquist` command does not provide an adequate representation for systems that have open-loop poles on the imaginary axis.** Therefore, we suggest that you copy the [nyquist1.m](#) file as a new m-file. This m-file creates more accurate Nyquist plots, since it correctly deals with poles and zeros on the imaginary axis.

The Nyquist diagram is basically a plot of  $G(j\omega)$  where  $G(s)$  is the open-loop transfer function and  $\omega$  is a vector of frequencies which encloses the entire right-half plane. In drawing the Nyquist diagram, both positive (from zero to infinity) and negative frequencies (from negative infinity to zero) are taken into account. We will represent positive frequencies in red and negative frequencies in green. The frequency vector used in plotting the Nyquist diagram usually looks like this (if you can imagine the plot stretching out to infinity):



However, if we have open-loop poles or zeros on the imaginary axis,  $G(s)$  will not be defined at those points and we must loop around them when we are plotting the contour. An example contour would appear as follows:



Please note that the contour loops around the pole on the imaginary axis. As we mentioned before, the MATLAB `nyquist` command does not take poles or zeros on the imaginary axis into account and, therefore, produces an incorrect plot. To correct this, please download and use [nyquist1.m](#). If we have a pole on the imaginary axis, we have to use `nyquist1`. If there are no poles or zeros on the imaginary axis, or if we have pole-zero cancellation, we can use either the `nyquist` command or `nyquist1.m`.

## The Cauchy Criterion

The Cauchy criterion (from complex analysis) states that when taking a closed contour in the complex plane, and mapping it through a complex function  $G(s)$ , the number of times that the plot of  $G(s)$  encircles the origin is equal to the number of zeros of  $G(s)$  enclosed by the frequency contour minus the number of poles of  $G(s)$  enclosed by the frequency contour. Encirclements of the origin are counted as positive if they are in the same direction as the original closed contour or negative if they are in the opposite direction.

When studying feedback control, we are not as interested in  $G(s)$  and, rather, are more concerned with the closed-loop transfer function:

$$\frac{G(s)}{1 + G(s)} \quad (2)$$

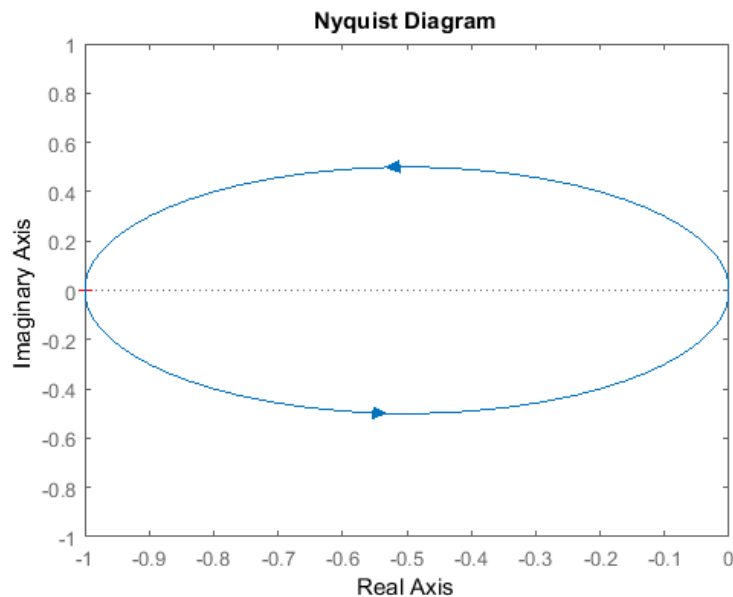
If  $1 + G(s)$  encircles the origin, then  $G(s)$  will enclose the point -1. Since we are interested in the closed-loop stability, we want to know if there are any closed-loop poles (zeros of  $1 + G(s)$ ) in the right-half plane. More details on how to determine this will come later.

Therefore, the behavior of the Nyquist diagram around the -1 point in the real axis is very important; however, the axis on the standard nyquist diagram might make it hard to see what's happening around this point. To correct this, you can add the `lnyquist.m` function to your files. The `lnyquist.m` command plots the Nyquist diagram using a logarithmic scale and preserves the characteristics of the -1 point.

To view a simple Nyquist plot using MATLAB, we will define the following transfer function and view the Nyquist plot:

$$G(s) = \frac{0.5}{s - 0.5} \quad (3)$$

```
s = tf('s');
G = 0.5/(s - 0.5);
nyquist(G)
axis([-1 0 -1 1])
```

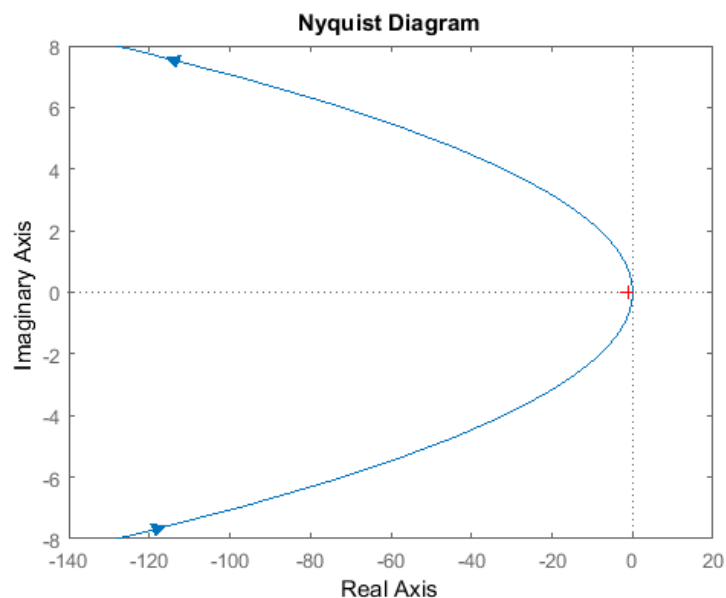


Now we will look at the Nyquist diagram for the following transfer function:

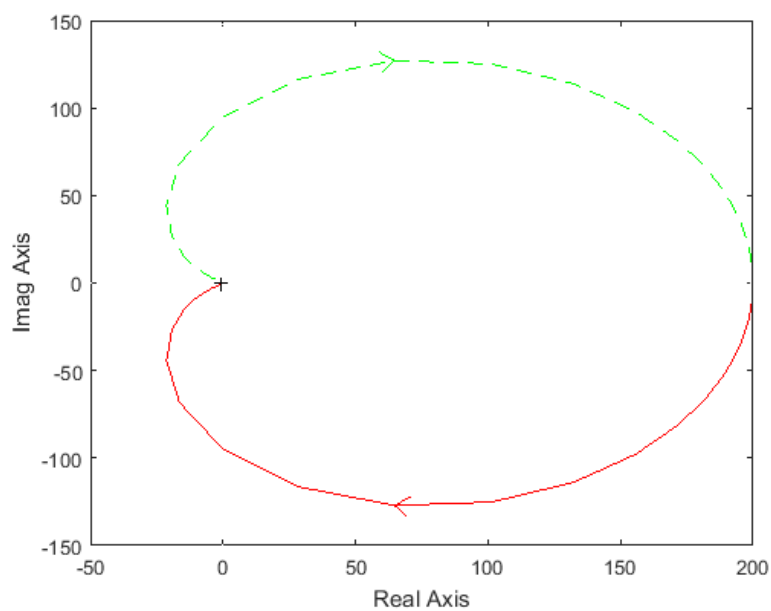
$$G(s) = \frac{s + 2}{s^2} \quad (4)$$

Note that this function has a pole at the origin. We will see the difference between using the `nyquist`, `nyquist1`, and `lNyquist` commands with this particular function.

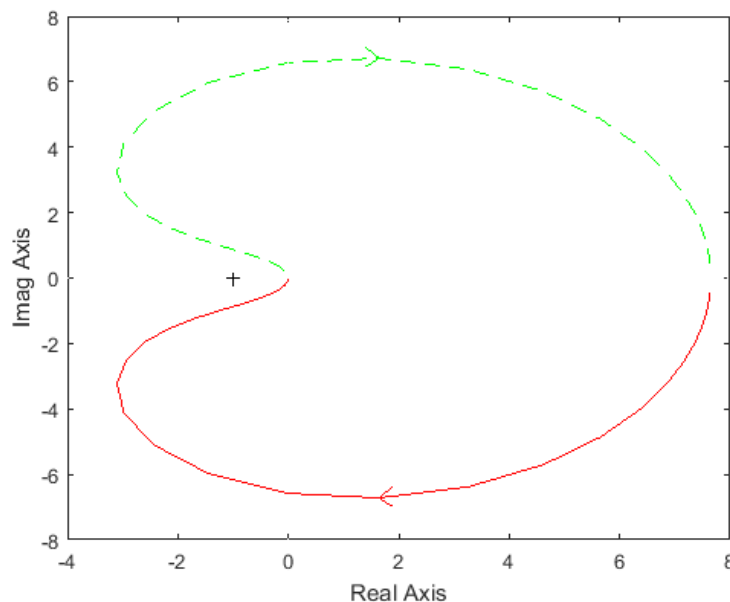
```
G = (s + 2)/(s^2);  
nyquist(G)
```



```
nyquist1(G)
```



lNyquist(G)



Note that the nyquist plot is not correct. In particular, it does not capture the behavior of the Nyquist plot for a radius approaching infinity ( $s$  approaching 0). The `nyquist1` plot has the correct shape, which allows us to evaluate encirclements of the -1 point and apply the Nyquist criterion, but it's difficult to see what happens close to the -1 point. The `lNyquist` function employs a modified scale that maintains the correct behavior of the plot near the -1 point and keeps the correct basic shape.

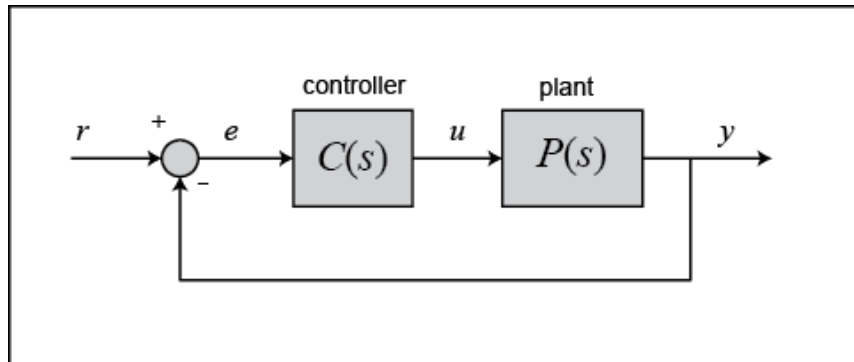
### Closed-Loop Performance from Bode Plots

In order to predict closed-loop performance from open-loop frequency response, we need to make a few assumptions:

- The system must be stable in open-loop if we are going to use Bode plots for our design.
- For canonical second-order systems, the closed-loop damping ratio is approximately equal to the phase margin divided by 100 if the phase margin is between 0 and 60 degrees. We can use this concept with caution if the phase margin is greater than 60 degrees.
- For canonical second-order systems, a relationship between damping ratio, bandwidth frequency, and settling time is given by an equation described on the [Extras: Bandwidth](#) page.

- A very rough estimate that you can use, is that the bandwidth is approximately equal to the natural frequency.

Let's use these concepts to design a controller for the following system:



Where  $C(s)$  is the controller and  $P(s)$  is:

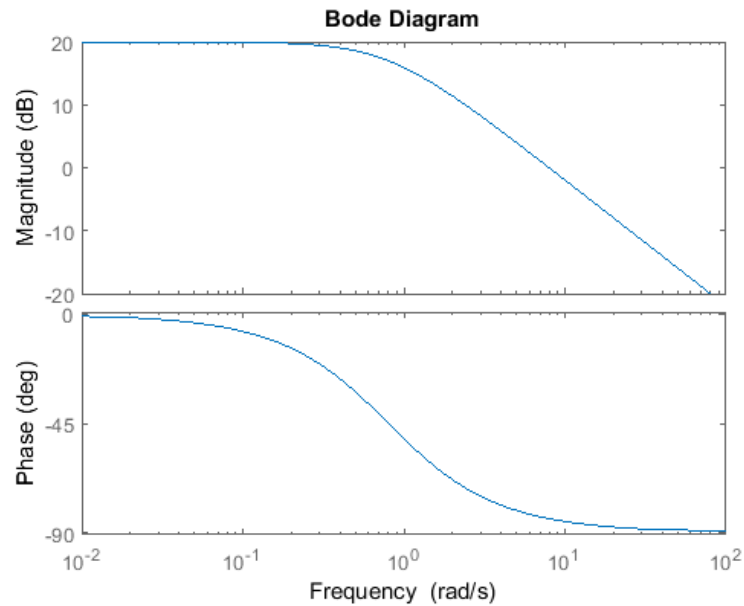
$$P(s) = \frac{10}{1.25s + 1} \quad (5)$$

The design must meet the following specifications:

- Zero steady-state error
- Maximum overshoot must be less than 40%
- Settling time must be less than 2 seconds

There are two ways of solving this problem: one is graphical and the other is numerical. Within MATLAB, the graphical approach is best, so that is the approach we will use. First, let's look at the Bode plot. Create an m-file with the following code:

```
P = 10/(1.25*s + 1);  
bode(P)
```

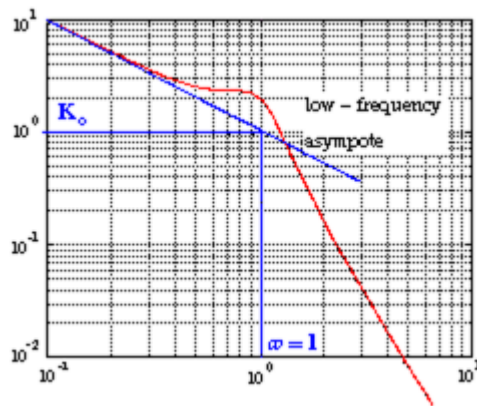


There are several characteristics of the system that can be read directly from this Bode plot. First of all, we can see that the bandwidth frequency is around 10 rad/sec. Since the bandwidth frequency is roughly the same as the natural frequency (for a first-order system of this type), the rise time is  $1.8/BW = 1.8/10 = 0.18$  seconds. This is a rough estimate, so we will say the rise time is about 0.2 seconds.

The phase margin for this system is approximately 95 degrees. The relation, damping ratio =  $PM/100$ , only holds for  $PM < 60$ . Since the system is first-order, there should be no overshoot.

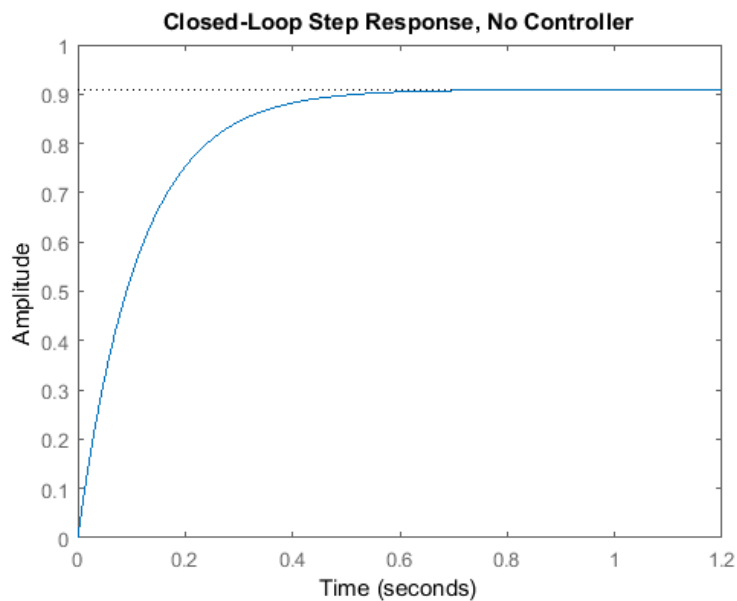
The last major point of interest is **steady-state error**. The steady-state error can be read directly off the Bode plot as well. The error constant ( $K_p$ ,  $K_v$ , or  $K_a$ ) is found from the intersection of the low frequency asymptote with the  $\omega = 1$  rad/sec line. Just extend the low frequency line to the  $\omega = 1$  line. The magnitude at this point is the error constant. Since the Bode plot of this system is a horizontal line at low frequencies (slope = 0), we know this system in unity feedback is type zero. Therefore, the intersection is easy to find. The gain is 20 dB (magnitude 10). What this means is that the constant for the error function is 10. The steady-state error is  $1/(1 + K_p) = 1/(1+10) = 0.091$ .

If our system was type one instead of type zero, the constant for the steady-state error would be found in a manner similar to the following.



Let's check our predictions by looking at a step response plot. This can be done by adding the following two lines of code into the MATLAB command window.

```
sys_cl = feedback(P,1);
step(sys_cl)
title('Closed-Loop Step Response, No Controller')
```



As you can see, our predictions were very good. The system has a rise time of about 0.2 seconds, has no overshoot, and has a steady-state error of about 9%. Now we need to choose a controller that will allow us to meet the design criteria. We choose a PI controller because it will yield zero steady-state error for a step input. Also, the PI controller has a zero, which we can place. This gives us additional design flexibility to help us meet our criteria. Recall that a PI controller is has the following form:



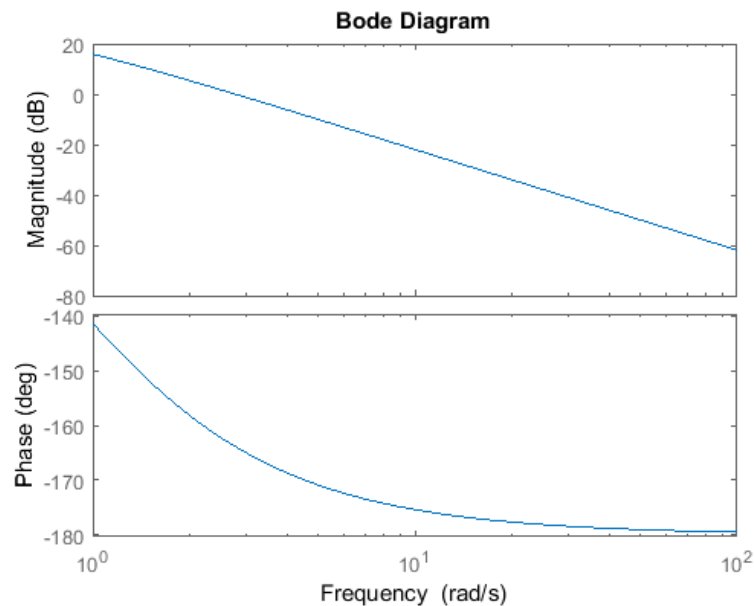
$$C(s) = \frac{K(s+a)}{s} = K + \frac{Ka}{s} \quad (6)$$

The first thing we need to find is the damping ratio corresponding to a percent overshoot of 40%. Plugging in this value into the equation relating overshoot and damping ratio (or consulting a plot of this relation), we find that the damping ratio corresponding to this overshoot is approximately 0.28. Therefore, our phase margin should be at least 30 degrees. We must have a bandwidth frequency greater than or equal to 12 if we want our settling time to be less than 1.75 seconds which meets the design specifications.

Now that we know our desired phase margin and bandwidth frequency, we can start our design. Remember that we are looking at the open-loop Bode plots. Therefore, our bandwidth frequency will be the frequency corresponding to a gain of approximately -7 dB.

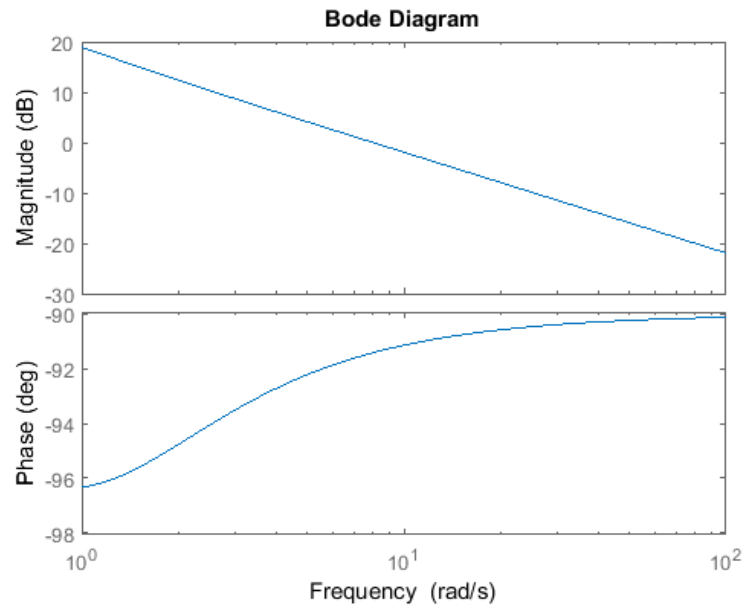
Let's see how the integrator portion of the PI controller affects our response. Change your m-file to the following (this adds an integral term but no proportional term):

```
P = 10/(1.25*s + 1);
C = 1/s;
bode(C*P, logspace(0,2))
```



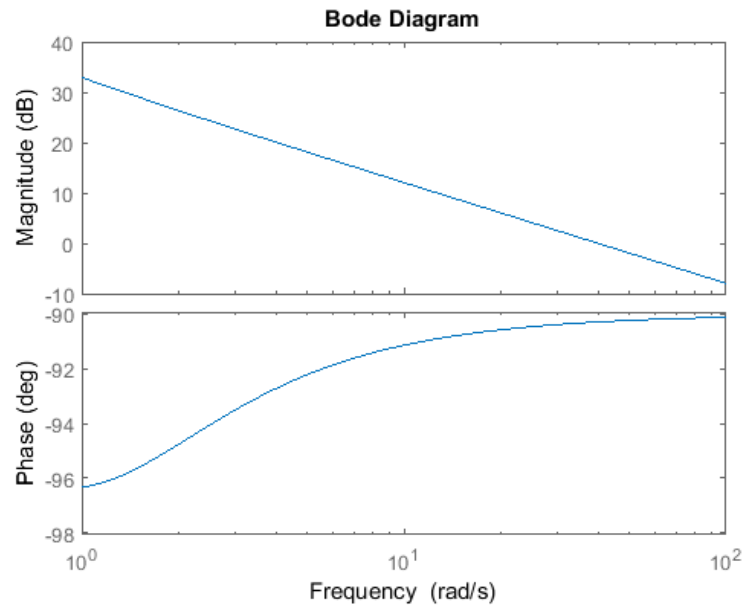
Our phase margin and bandwidth frequency are too small. We will add gain and phase with a zero. Let's place the zero at -1 for now and see what happens. Change your m-file to look like the following:

```
P = 10/(1.25*s + 1);  
C = (s + 1)/s;  
bode(C*P, logspace(0,2))
```



It turns out that the zero at  $-1$  with a unit gain gives us a satisfactory answer. Our phase margin is greater than 60 degrees (even less overshoot than expected) and our bandwidth frequency is approximately 11 rad/s, which will give us a satisfactory response. Although satisfactory, the response is not quite as good as we would like. Therefore, let's try to get a higher bandwidth frequency without changing the phase margin too much. Let's try to increase the gain to 5 and see what happens. This will make the gain shift and the phase will remain the same.

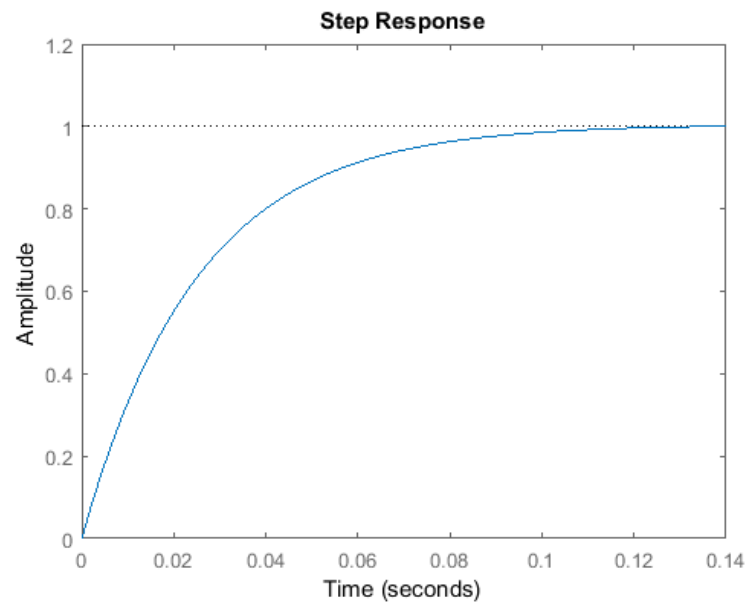
```
P = 10/(1.25*s + 1);  
C = 5*((s + 1)/s);  
bode(C*P, logspace(0,2))
```



That looks really good. Let's look at our step response and verify our results.

Add the following two lines to your m-file:

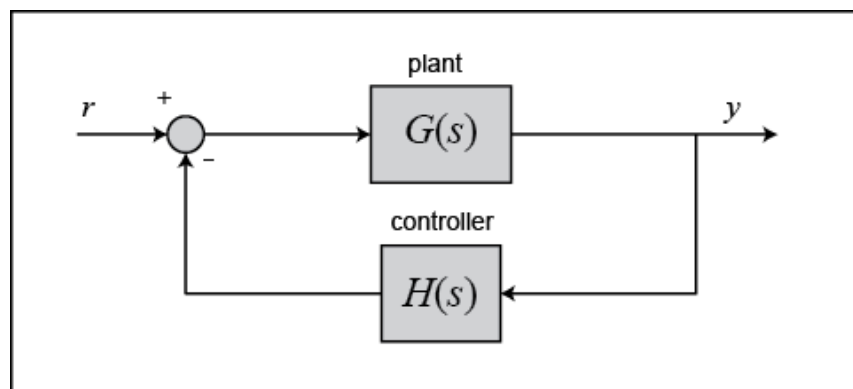
```
sys_cl = feedback(C*P,1);  
step(sys_cl)
```



As you can see, our response is better than we had hoped for. However, we are not always quite as lucky and usually have to play around with the gain and the position of the poles and/or zeros in order to achieve our design requirements.

## Closed-Loop Stability from the Nyquist Diagram

Consider the negative feedback system:



Remember from the Cauchy criterion that the number (N) of times that the plot of  $G(s)H(s)$  encircles  $-1+j.0$  is equal to the number (Z) of zeros of  $1 + G(s)H(s)$  enclosed by the frequency contour minus the number (P) of poles of  $1 + G(s)H(s)$  enclosed by the frequency contour ( $N = Z - P$ ). Keeping careful track of the open- and closed-loop transfer functions, as well as numerators and denominators, you should convince yourself that:

- The zeros of  $1 + G(s)H(s)$  are the poles of the closed-loop transfer function.
- The poles of  $1 + G(s)H(s)$  are the poles of the open-loop transfer function.

Based on the above definitions:

- $P$  = the number of open-loop (unstable) poles of  $G(s)H(s)$ .
- $N$  = the number of times the Nyquist diagram encircles -1.
- clockwise encirclements of -1 count as positive encirclements.
- counter-clockwise encirclements of -1 count as negative encirclements.
- $Z$  = the number of right-half-plane (positive, real) poles of the closed-loop system.

The Nyquist Stability Criterion that relates these three quantities is then:

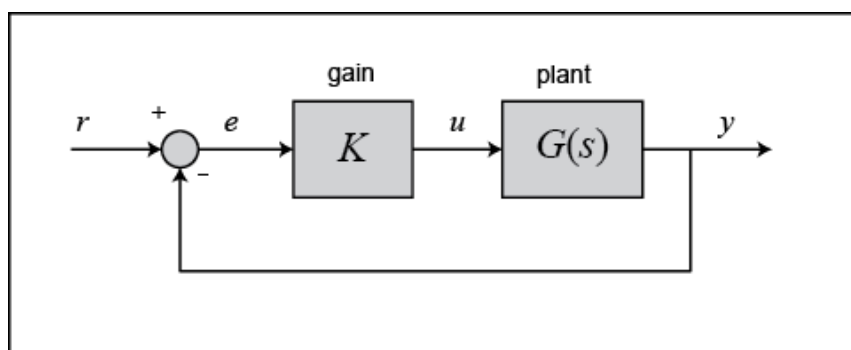
$$Z = P + N \quad (7)$$

**Note:** This is only one convention for the Nyquist criterion. Another convention states that a positive N counts the counterclockwise or anti-clockwise encirclements of -1. The P and Z variables remain the same. In this case the equation becomes  $Z = P - N$ . Throughout these tutorials, we will use a positive sign for clockwise encirclements.

It is very important (and somewhat tricky) to learn how to count the number of times that the Nyquist diagram encircles -1. Therefore, we will go into some detail to help you visualize this. One approach is to imagine you are standing on top of the -1 point and are following the diagram from beginning to end. Now ask yourself: How many times did I turn my head a full 360 degrees? Again, if the motion was clockwise,  $N$  is positive, and if the motion is counterclockwise,  $N$  is negative.

Knowing the number of right-half plane (unstable) poles of the open loop ( $P$ ), and the number of encirclements of -1 made by the Nyquist diagram ( $N$ ), we can determine the closed-loop stability of the system. If  $Z = P + N$  is a positive, nonzero number, the closed-loop system is unstable.

We can also use the Nyquist diagram to find the range of gains for a closed-loop unity feedback system to be stable. The system we will test looks like this:



where:

$$G(s) = \frac{s^2 + 10s + 24}{s^2 - 8s + 15} \quad (8)$$

This system has a gain  $K$  which can be varied in order to modify the response of the closed-loop system. However, we will see that we can only vary this gain within certain limits, since we have to make sure that our closed-loop system will be stable. This is what we will be looking for: the range of gains that will make this system stable in the closed-loop.

The first thing we need to do is find the number of positive real poles in our open-loop transfer function:

```
roots([1 -8 15])
```

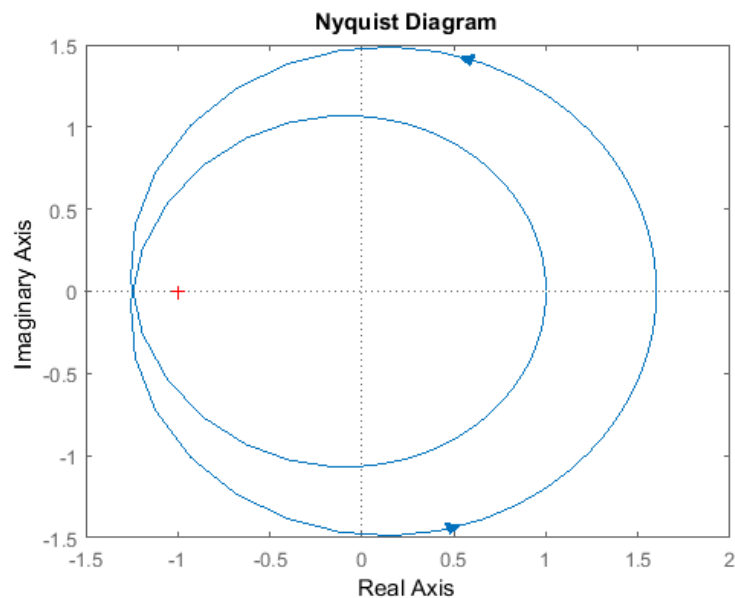
```
ans =
```

```
5
```

The poles of the open-loop transfer function are both positive. Therefore, we need two counterclockwise ( $N = -2$ ) encirclements of the Nyquist diagram in order to have a stable closed-loop system ( $Z = P + N$ ). If the number of encirclements is less than two or the encirclements are not counterclockwise, our system will be unstable.

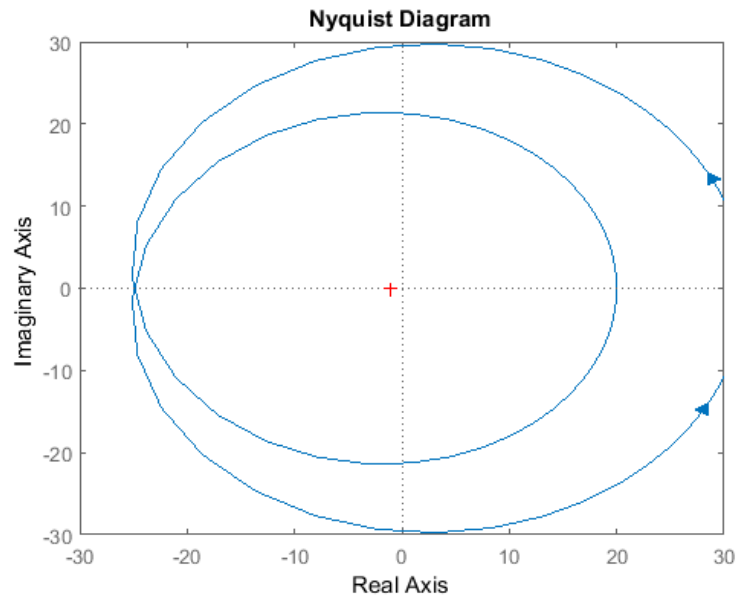
Let's look at our Nyquist diagram for a gain of 1:

```
G = (s^2 + 10*s + 24)/(s^2 - 8*s + 15);  
nyquist(G)
```



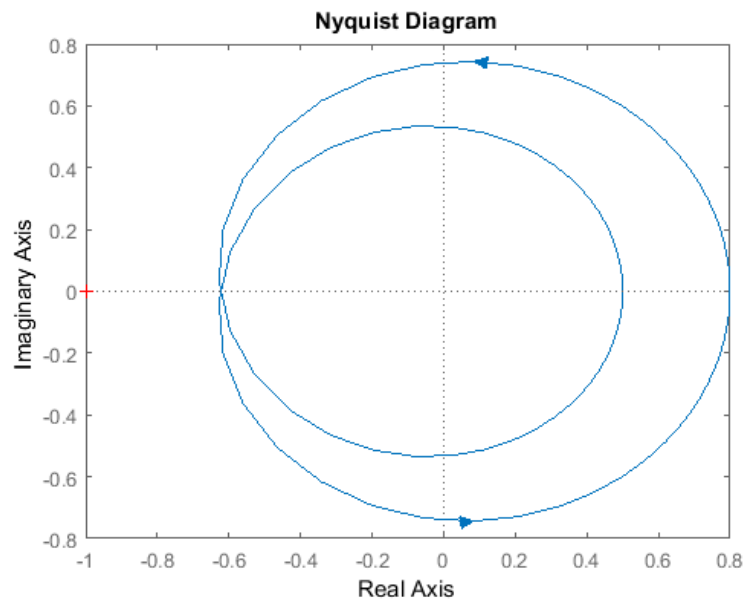
There are two counterclockwise encirclements of -1. Therefore, the system is stable for a gain of 1. Now we will see how the system behaves if we increase the gain to 20:

```
nyquist(20*G)
```



The diagram expanded. Therefore, we know that the closed-loop system will be stable no matter how much we increase the gain. However, if we decrease the gain, the diagram will contract and the system might become unstable. Let's see what happens for a gain of 0.5:

```
nyquist(0.5*G)
```



The system is now unstable. By trial and error we find that this system will become unstable for gains less than 0.80. We can verify our answers by

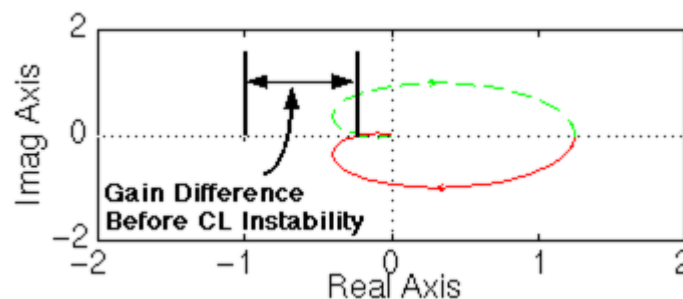
zooming in on the Nyquist plots as well as by looking at the closed-loop step responses for gains of 0.79, 0.80, and 0.81.

### Gain Margin

We already defined the gain margin as the change in open-loop gain expressed in decibels (dB), required at -180 degrees of phase shift to make the open-loop magnitude equal 0 dB. Now we are going to find out where this comes from. First of all, let's say that we have a system that is stable if there are no Nyquist encirclements of -1, such as:

$$G(s) = \frac{50}{s^3 + 9s^2 + 30s + 40} \quad (9)$$

Looking at the roots, we find that we have no open loop poles in the right-half plane. Therefore, there are no closed-loop poles in the right-half plane if there are no Nyquist encirclements of -1. Now, how much can we vary the gain before this system becomes unstable in closed-loop? Let's look at the following figure:



The open-loop system represented by this plot will become unstable in closed loop if the gain is increased past a certain boundary. The negative real axis area between  $-1/a$  (defined as the point where the open-loop has -180 degree of phase, that is, where the diagram crosses the real axis) and -1 represents the amount of increase in gain that can be tolerated before closed-loop instability.

If we think about it, we realize that if the added gain is equal to  $a$ , the diagram will touch the -1 point:

$$G(j\omega) = \frac{-1}{a} \quad (10)$$

or

$$aG(j\omega) = -1 \quad (11)$$

Therefore, we say that the gain margin is  $a$  units. However, we mentioned before that the gain margin is usually measured in decibels. Hence, the gain margin is:



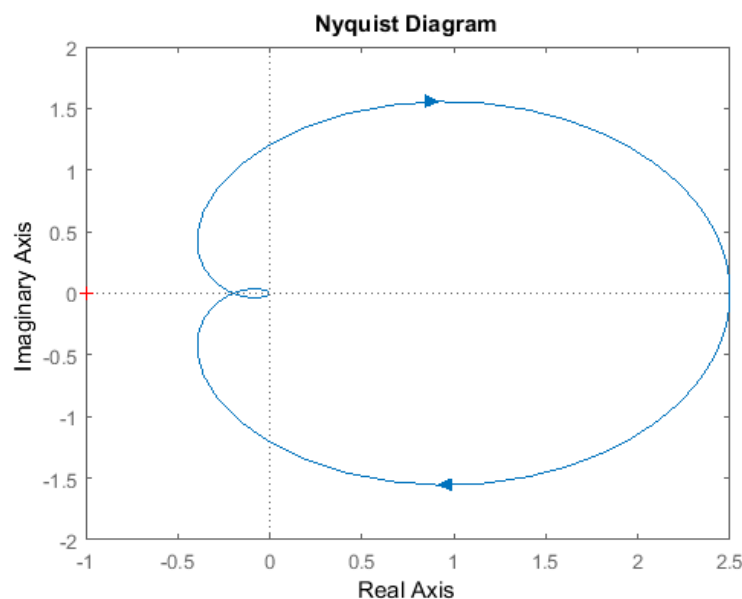
$$GM = 20 \log_{10}(a) \text{ dB} \quad (12)$$

We will now find the gain margin of the stable, open-loop transfer function we viewed before. Recall that the function is:

$$G(s) = \frac{50}{s^3 + 9s^2 + 30s + 40} \quad (13)$$

and that the Nyquist diagram can be viewed by typing:

```
G = 50/(s^3 + 9*s^2 + 30*s + 20);
nyquist(G)
```



As we discussed before, all that we need to do to find the gain margin is find  $a$ , as defined in the preceding figure. To do this, we need to find the point where there is exactly  $-180$  degrees of phase shift. This means that the transfer function at this point is real (has no imaginary part). The numerator is already real, so we just need to look at the denominator. When  $s = j\omega$ , the only terms in the denominator that will have imaginary parts are those which are odd powers of  $s$ . Therefore, for  $G(j\omega)$  to be real, we must have:

$$-j\omega^3 + 30j\omega = 0 \quad (14)$$

which means  $\omega = 0$  (this is the rightmost point in the Nyquist diagram) or  $\omega = \sqrt{30}$ . We can then find the value of  $G(j\omega)$  at this point using polyval:

```
w = sqrt(30);
polyval(50,j*w)/polyval([1 9 30 40],j*w)
```

```
ans =  
-0.2174
```

The answer is:  $-0.2174 + 0i$ . The imaginary part is zero, so we know that our answer is correct. We can also verify this by looking at the Nyquist plot again. The real part also makes sense. Now we can proceed to find the gain margin.

We found that the  $-180$  degrees phase shift occurs at  $-0.2174 + 0i$ . This point was previously defined as  $-1/a$ . Therefore, we now have  $a$ , which is the gain margin. However, we need to express the gain margin in decibels:

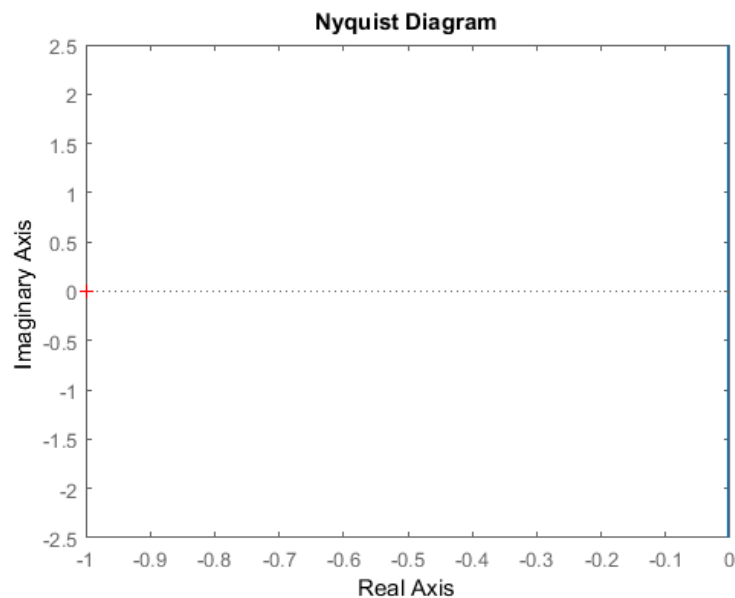
$$\frac{-1}{a} = -0.2174 \quad (15)$$

$$\Rightarrow a = 4.6 \quad (16)$$

$$\Rightarrow GM = 20 \log_{10}(4.6) = 13.26 \text{ dB} \quad (17)$$

We now have our gain margin. Let's see how accurate it is by using a gain of  $a = 4.6$  and zooming in on the Nyquist plot:

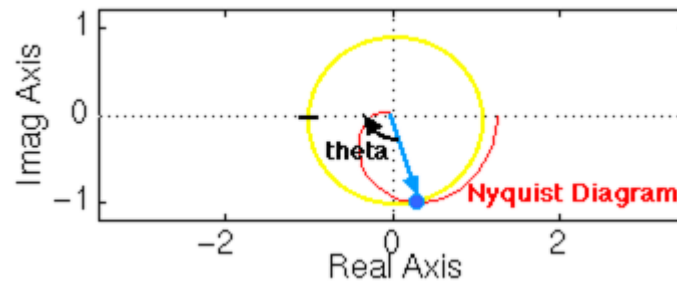
```
a = 4.6;  
nyquist(a*sys)
```



The plot appears to go right through the  $-1$  point. We will now verify the accuracy of our results by viewing the zoomed Nyquist diagrams and closed-loop step responses for gains of  $4.5$ ,  $4.6$ , and  $4.7$ .

### Phase Margin

We have already discussed the importance of the phase margin. Therefore, we will only talk about where this concept comes from. We have defined the phase margin as the change in open-loop phase shift required at unity gain to make a closed-loop system unstable. Let's look at the following graphical definition of this concept to get a better idea of what we are talking about.



Let's analyze the previous plot and think about what is happening. From our previous example we know that this particular system will be unstable in closed-loop if the Nyquist diagram encircles the -1 point. However, we must also realize that if the diagram is shifted by  $\theta$  degrees, it will then touch the -1 point at the negative real axis, making the system marginally stable in closed-loop. Therefore, the angle required to make this system marginally stable in closed-loop is called the phase margin (measured in degrees). In order to find the point we measure this angle from, we draw a circle with radius of 1, find the point in the Nyquist diagram with a magnitude of 1 (gain of 0 dB), and measure the phase shift needed for this point to be at an angle of -180 degrees.

---

*Published with MATLAB® 9.2*

All contents licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

