The Magic of –1: A tortuous and torturous path to the Nyquist Criterion

What we try to explain is the Nyquist stability criterion, which leads to (1) why we can use the open-loop function, and (2) the gain and phase margins. This section is not a formal proof of the criterion, but rather tries to explain the ideas behind without more advanced complex variable techniques. The material follows the same approach as the explanation given in Phillips and Harbor (2000). More formal treatment can be found in, for example, Ogata (1997).

Step 1. Notation

We first need to clarify our slight change in terminology. For simplicity, we take $G_m = G_a = 1$, and the unity feedback system transfer function is

$$\frac{C}{R} = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \tag{1-1}$$

Stability is determined by the roots of the characteristic polynomial $1 + G_c(s)G_p(s) = 0$. The roots are the closed-loop poles of the system. From here on, we'll talk about the *zeros* (*i.e.*, roots) of an equation, $1 + G_c(s)G_p(s) = 0$, and that we do not want them on the right hand plane.

Step 2. Mapping to the [1 + $G_c(s)G_p(s)$]-plane and to the [$G_c(s)G_p(s)$]-plane

For a given transfer function G(s), we know that we can plot G(jω) in polar coordinates—the Nyquist plot. We can do the same with $[1 + G_c(s)G_p(s)]$ and plot $[1 + G_c(j\omega)G_p(j\omega)]$, again with ω as the parameter. This is what we refer to as mapping from the s-plane to the $[1 + G_c(s)G_p(s)]$-plane. In this respect, a Nyquist plot is a mapping of the imaginary axis ($-j\infty$ to $+j\infty$) from the s-plane to the G(jω)-plane.

We can go one step further and translate the coordinate system and shift the imaginary axis to –1. In other words, we can consider

$$G_c(j\omega)G_p(j\omega) = -1 \tag{1-2}$$

We only need to plot $G_c(j\omega)G_p(j\omega)$ on this translated G_cG_p-plane. Hence all of our analysis can be performed in terms of the open-loop function G_cG_p even though we are solving a closed-loop system. This idea is one reason why we use frequency response methods in controller design. We always have the open-loop function in any problem and we know all of the open-loop zeros and poles. The closed-loop equation could be messy, and now we do not have to do any extra work to address the closed-loop problem. Also, until a problem is "solved," we certainly do not know what the closed-loop poles are.

For example, consider the mapping of the imaginary axis $s = j\omega$, with $\omega = 0$ to ∞, by

$$G(s) = \frac{K}{s + 1}$$

The image is the half-circle below the real axis in the G-plane (Fig. 1). Likewise, the image of the negative imaginary axis would give the upper half-circle.

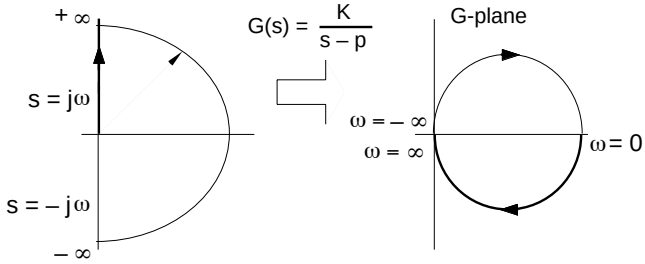

Figure 1.

To find the image of the entire RHP, we can consider the RHP as a semi-infinite half-circle, $s = Re^{j\theta}$, where $R \to \infty$ and θ varies from +90° to –90°. The resulting image is the origin in the G-plane. The imaginary axis and the semi-infinite half-circle in the RHP is called the Nyquist path.

Step 3. Mapping of pole-zero forms

To begin, consider a simple function $G(s) = s - z_1$, *i.e.*, z_1 is the zero of G(s). We arbitrarily pick *s* to be a unit circle clockwise around z_1 and we map *s* onto the G-plane, *i.e.*, the $(s - z_1)$-plane. The image is a unit circle about the origin in the G-plane (Fig. 2). The argument, $\arg(s - z_1)$, has changed by –360°. If we repeat with another odd shaped closed contour, the argument would still also be changed by –360°.

Next, consider a slightly more complex mapping

$$G(s) = \frac{(s - z_1)(s - z_2)(s - z_3)}{(s - p_1)(s - p_2)}$$

We pick an arbitrary closed contour, *s*, which encircles only z_1, z_2, and p_1 (Fig. 3). We use the fact that $\angle(G) = \angle(s - z_1) + \angle(s - z_2) + \angle(s - z_3) - \angle(s - p_1) - \angle(s - p_2)$. Now we go around the curve once, say clockwise. Both z_1 and z_2 contribute –360° each while p_1 contributes +360°. The effective angle contribution of both z_3 and p_2 outside the contour is zero, even though they may affect the shape of the image. (Try it with a string.) Hence, the argument of the image in the G-plane should have changed by 2(–360°) + 360° = –360°, *i.e.*, one clockwise encirclement of the origin in the G-plane.

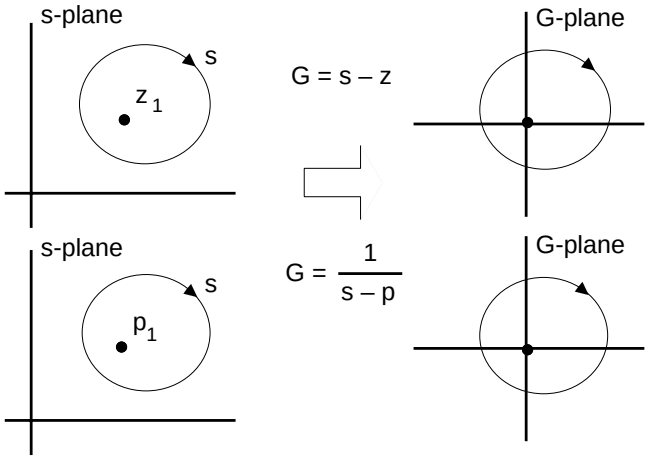

Figure 2.

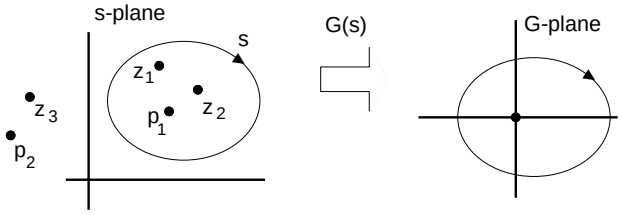

Figure 3. The shape of the curves is arbitrary. It is unlikely that the transformed curve can retain the original shape.

We can write for all z_i and p_i within a chosen contour *s*

$$N = Z - P$$

where N is the number of encirclement in the G-plane, and Z and P are the number of zeros and poles of G(s) = 0.

Finally, we pick our arbitrary contour *s* to be the right hand plane (*i.e.*, a semi-infinite half circle). The number of encirclement (in the G-plane) after the G(s) mapping is

$$N = Z - P \quad \text{for all } z_i,\ p_i \text{ in the right hand } s\text{-plane}$$

This relation tells us the net difference of zeros and poles of the function G(s) in the RHP; N = 1 may mean 1 zero with 0 pole, or 2 zeros with 1 pole, or 3 zeros with 2 poles, etc. Note that if N = 0, it only means Z = P. We may have Z = P = 0, or we may have Z = P = 1.

Step 4. Back to the 1 + G_cG_p Mapping

We take $G = 1 + G_cG_p$ and rearrange it into its pole-zero form. The object to be mapped is the Nyquist path as mentioned in Step 2. The net number of encirclements around the origin in the right hand plane is (Z – P). Recall Step 1. Zeros in the right hand plane are really the poles of our closed-loop characteristic polynomial $1 + G_cG_p$. Hence even if there is just one zero (one RHP root) of $1 + G_cG_p$, the system is unstable. The important point is that we *do not want any*

encirclement of the origin in the $[1 + G_cG_p]$-plane at all.

In most introductory problems, the process is open-loop stable, and P = 0.[1] We essentially are looking at the roots of a polynomial. The condition that we really are testing is N = Z, and whether N is equal to 0 or not.

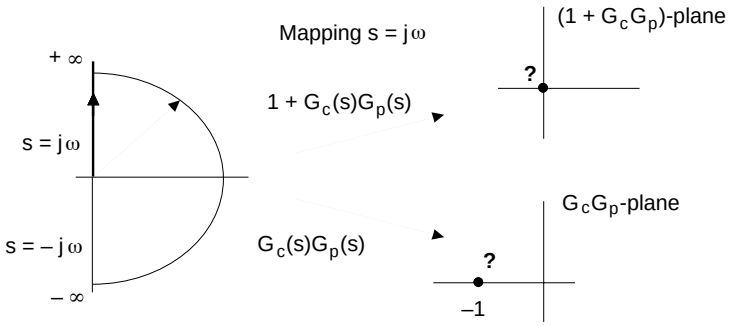

Figure 4.

Step 5. Shift the Imaginary Axis

We repeat Step 4 except that the coordinate system is translated such that the imaginary axis is at –1. As explained in Step 2, we now look for encirclement about the point (–1,0) after a mapping of the RHP. If there is a net encirclement, the system is unstable.

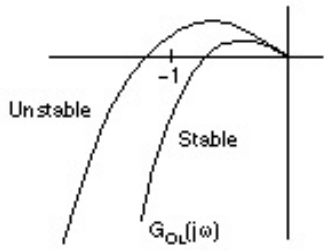

Figure 5

Now we can finally state the *Nyquist Stability Criterion*: A closed-loop system with characteristic polynomial $1 + G_cG_p$ is stable if and only if the net number of clockwise encirclement of (–1,0) in a Nyquist plot of the open-loop function is zero. In simpler terms, any encirclement of (–1,0) by the open-loop function on the polar plot means instability. In Fig. 5, we have shown the mapping of only the positive imaginary axis, and thus there is only a half encirclement of (–1,0) when the system is unstable.

Relation to the Bode plot. The point (–1, 0) has a magnitude of 1 and angle –180°. This is the

---

[1] Consider the closed-loop characteristic equation,

$$G(s) = 1 + K_c \frac{K_p(\tau_D s + 1)}{(\tau_p s + 1)} = \frac{(\tau_p s + 1) + K_c K_p(\tau_D s + 1)}{(\tau_p s + 1)} = 0 \cdot$$

For open-loop stable processes, the number of poles in the RHP is zero (*i.e.*, P = 0). If there are no roots in the RHP, Z and hence N are also zero.

What if I use a PI controller and there is an open-loop pole at the origin? In this case, when we draw the Nyquist path in Step 2, we use a tiny semicircle "detour" to bypass the origin along the imaginary axis so that this pole will not be mapped into the $1 + G_cG_p$-plane. All our results so far are then applicable.

What if the system has an open-loop pole that is in the RHP? Try the following example in MATLAB: $G(s) = 1 + \frac{K}{(s-1)(s+2)(s+4)} = 0$, where we have one open-loop pole in the RHP, P = 1. First, we can use a Routh array to show that the system is stable for $8 < K < 18$. When the system is stable, we need Z = 0; thus we must have N = –1. This means a *counterclockwise* encirclement of (–1, 0), and you can see that by doing a Nyquist plot with MATLAB.

basis of the stability criterion in the magnitude and phase lag plots of the open-loop function. To be stable, the magnitude cannot be larger than 1 at a phase lag of –180°, the gain crossover frequency.

Pitfall in applying the Nyquist Criterion to a Bode plot

We have to be careful when we use the Bode plot to interpret the Nyquist Stability Criterion. Using the Bode plot is not as straightforward with problems where there are multiple crossover points. We'll illustrate this point with an example.[1]

Consider a simple unity feedback system with the characteristic equation $1 + G_cG_p = 0$, with

$$G_p = \frac{-s+1}{s(\tau s+1)} e^{-\theta s}, \quad \text{and} \quad G_c = K_c \frac{\tau_D s+1}{\alpha \tau_D s+1}.$$

The numerical values to be used are $\tau = 0.1$ min, $\theta = 0.4$ min, $\tau_D = 1$ min, and $\alpha = 0.05$. This is a tricky problem. Not only is there a positive open-loop zero, but the dead time is extremely large.

The first thing to do is to find the ultimate gain of this system. Following Section 8.4 in the text, we make a Bode plot with $K_c = 1$. The MATLAB statements are:

```
Kc=1;
tau=0.1;  tdead=0.4;
taud=1;   alf=0.05;

Gp=tf([-1 1],[tau 1 0]);
Gc=tf(Kc*[taud 1],[alf*taud 1]);
G=Gc*Gp;
ezbo
subplot(212) %Need to limit the scale of the phase angle axis
axis([0.1 100 -600 0])
```

Here, `ezbo.m` is our M-file posted on the *Web Support* that does a Bode plot with dead time. In this Bode plot (not shown), the magnitude curve is always above 1. At –180°, the magnitude is 0.327 (what MATLAB thought is the "gain margin"). So we need to reduce the magnitude to achieve a gain margin of 1 by using $K_c = 0.327$. So we set

```
Kc=0.327;
%and repeat all the other statements listed above
```

The result is shown in Fig. 6. There is a "hump" near frequency 10 rad/min, arising from the fact that the corner frequency of the lag term is much larger than the lead term in G_p. The magnitude is 1 at –180°, but not so at –540° (–180° –360°). The system is still unstable if we use $K_c = 0.327$; the ultimate gain is lower than that. From Fig. 6, the magnitude is approximately 2.18 at –540°.[2] So the actual ultimate gain should be

$$K_{cu} = 0.327/2.18 = 0.15$$

---

[1] This example is taken from "A Note on Stability Analysis using Bode Plots," J. Hahn, T. Edison, and T. F. Edgar, *Chem. Eng. Edu.*, 208-211 (Summer, 2001). And of course, we consider our explanation here much more sensible that what you find in this article.

[2] Our M-file `ezbo.m` stores the values of the Bode plot in an array, and it is from this array that we make the interpolation.

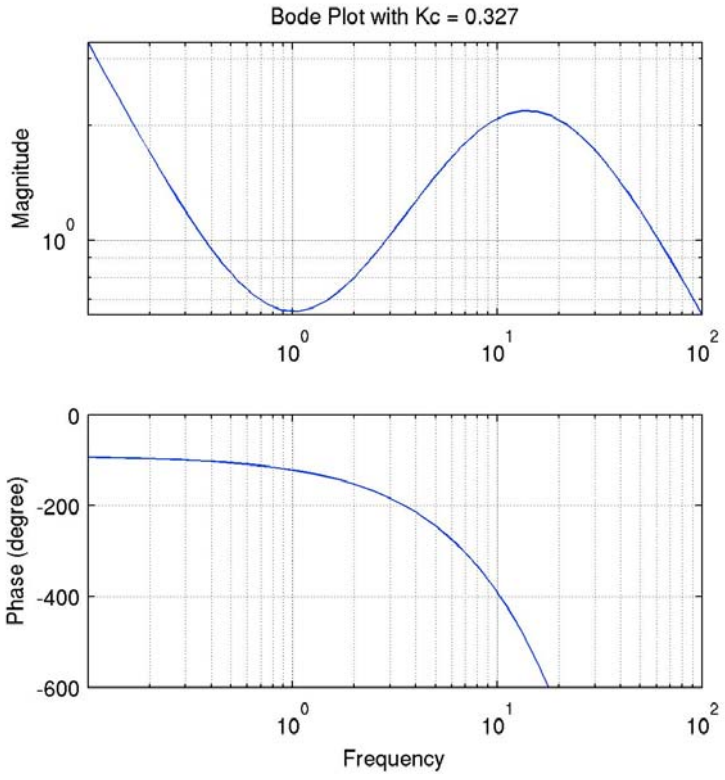

Figure 6.

We can confirm the result with time domain simulation. With such a large dead time, we need to use Simulink. (The Simulink file named `largelag.mdl` for this example is provided among the list of samples on our Simulink page on the *Web Support*.) Running simulations with different values of K_c, we should find that the system is indeed unstable when $K_c > 0.15$. This problem, of course, can be visualized with a Nyquist plot. The statements that we can use are

```
Kc=1; %Repeated with 0.2 and 0.15 in Fig. 7

%Other values and Gp remain the same as before
Gc=tf(Kc*[taud 1],[alf*taud 1]);
G=Gc*Gp;

freq=logspace(-1,2,400); %400 points for a smoother curve
[Mag,Phase]=bode(G,freq);
Mag=Mag(1,:);
Phase=Phase(1,:);
Phase=Phase - ((180/pi)*tdead*freq);
polar(Phase*pi/180,Mag) %Do the polar plot
hold on
polar(pi,1,'rx') %Add the –1 point
hold off
```

The Nyquist plots using three different values of K_c are shown in Fig. 7. When $K_c = 1$, the Nyquist curve always encircles the –1 point, consistent with the fact that the magnitude is always larger than 1.

When we choose, say, $K_c = 0.2$, a value that is less than 0.327, the system appears to be stable at lower frequencies, but eventually the Nyquist plot encircles the –1 point at higher frequencies, meaning that the system is still unstable. Finally, at the ultimate gain $K_{cu} = 0.15$, the plot only "touches" the –1 point. The system should be stable when $K_c < 0.15$.

The crux of this example is that when the magnitude does not decrease monotonously, we need to assess the stability situation at higher frequencies, or in other words, crossover at phase angles

larger than –180°. Theoretically, we need to look at all multiples of –180°. In more practical situations, this problem is as tricky as it gets.

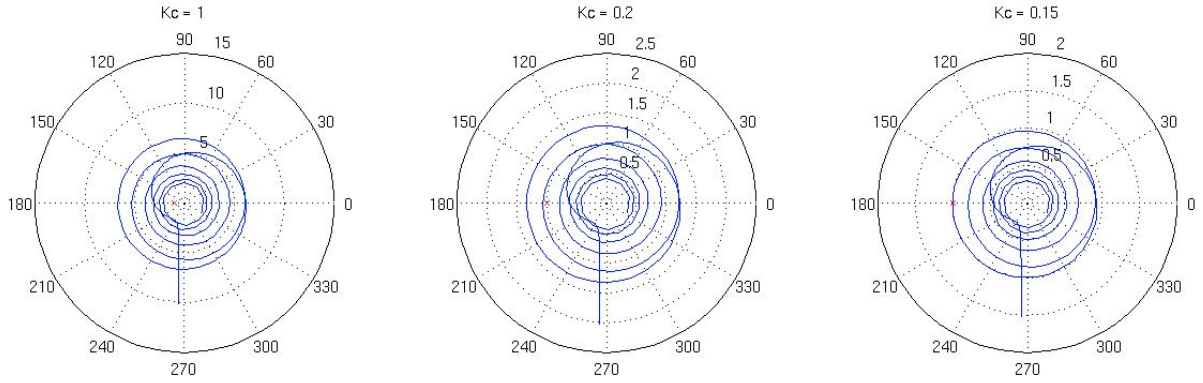

Figure 7. From left to right, the values of K_c are 1, 0.2, and 0.15. The tiny (red) cross is the point (–1, 0).

Generation of Bode plots with experimental data

Theoretically speaking, we can do a frequency response experiment by varying the input frequency and thus generating a Bode plot with data. In reality, it only works for electronics or electrical systems. However, we can still generate a Bode plot with a **pulse response** experiment if we know **Fourier transform**. Let's consider a transfer function and the definition of Laplace transform on both the input and output:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{\int_0^\infty y(t)\, e^{-st}\, dt}{\int_0^\infty x(t)\, e^{-st}\, dt} . \qquad (2\text{-}1)$$

Follow the common technique of frequency response analysis, we make the substitution $s = j\omega$ to obtain

$$G(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{\int_0^\infty y(t)\, e^{-j\omega t}\, dt}{\int_0^\infty x(t)\, e^{-j\omega t}\, dt} . \qquad (2\text{-}2)$$

The idea (not that we actually do it) is that we can pick a value of ω, substitute it in Eq. (2-2) and make use of the experimental data for x(t) and y(t) to calculate the magnitude and argument of G(jω). We pick another value of ω again, and repeat the calculation. Before long, we'd have our Bode plot.

For the integral to be convergent, it is important that the input x(t) starts at zero (we work with deviation variables) and decays back to zero within a finite time span. Likewise, the process that we study must be stable such that the response y(t) also decays eventually back to zero.

Because of the requirement on x(t), we cannot use a step input. Instead, the input is usually some kind of a pulse, and the experiment hence is called pulse testing. Two common input pulses are a rectangular pulse and a triangular pulse in which we ramp up the input value and then back down to zero. (Of course, we know that an impulse experiment cannot be real.)

Note that the Fourier transform of a function f(t) is defined as

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-j\omega t}\, dt \ . \tag{2-3}$$

Since we work with deviation variables with the condition of f(t) = 0 for t ≤ 0, the lower integral limit of the Fourier transform can be changed to zero. It is now apparent that we can analyze Eq. (2) with numerical algorithms using Fourier transform—namely fast Fourier transform.

We can apply fast Fourier transform to sequences of input x(t) and response y(t), and from the transforms, calculate:

$$|G(j\omega)| = \frac{|Y(j\omega)|}{|X(j\omega)|}\ , \text{ and } \quad \arg\,(G(j\omega)) = \arg\,(Y(j\omega)) - \arg\,(X(j\omega)) \tag{1-4}$$

We use the term sequence because virtually all computer implementations of fast Fourier transform are based on discrete-time analysis.

What follows are equations which serve as a math review. It is sort of an idea that even if we do not know Fourier transform, we could still analyze our pulse data to generate a Bode plot. All we need is a numerical quadrature procedure and a big loop for different values of the frequency.

Substitute the Euler identity $e^{-j\omega t} = \cos \omega t - j \sin \omega t$ in Eq. (2-2), and we should obtain

$$G(j\omega) = \frac{A_r(\omega) - j\, A_i(\omega)}{B_r(\omega) - j\, B_i(\omega)} \tag{2-5}$$

where $A_r(\omega) = \int_0^{T_y} y(t) \cos \omega t\, dt$, $A_i(\omega) = \int_0^{T_y} y(t) \sin \omega t\, dt$,

and $B_r(\omega) = \int_0^{T_x} x(t) \cos \omega t\, dt$, $B_i(\omega) = \int_0^{T_x} x(t) \sin \omega t\, dt$.

We have changed the infinity upper integral limit to times when the data presumably have decayed back to zero.

We can also rewrite Eq. (2-5) as

$$G(j\omega) = \left[\frac{A_r B_r + A_i B_i}{B_r^2 + B_i^2}\right] + j\left[\frac{A_r B_i - A_i B_r}{B_r^2 + B_i^2}\right] \tag{2-6}$$

where we have omitted the explicit frequency dependence of the functions A's and B's. With Eq. (2-6), we can calculate the magnitude and argument of G(jω), with frequency ω as the parameter.

This is a good place to stop. It is risky to perform the analysis without learning more about the properties of discrete fast Fourier transform.