Root Locus and Frequency Analysis

Second-order systems are commonly encountered in practice, and are the simplest type of dynamic system to exhibit oscillations. Examples include mass-spring-damper systems and RLC circuits. In fact, many true higher-order systems may be approximated as second-order in order to facilitate analysis.

The canonical form of the second-order differential equation is as follows

$m\ddot{y} + b\dot{y} + ky = f(t)$ or $\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = k_{dc}\omega_n^2\, u(t)$

The canonical second-order transfer function has the following form, in which it has two poles and no zeros.

$$G(s) = \frac{1}{ms^2 + bs + k} = \frac{k_{dc}\omega_n^2}{\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y}$$

The parameters k_{dc}, ζ, and ω_n characterize the behavior of a canonical second-order system.

DC Gain

The DC gain, k_{dc}, again is the ratio of the magnitude of the steady-state step response to the magnitude of the step input, and for stable systems it is the value of the transfer function when $s = 0$. For the forms given,

$k_{dc} = \frac{1}{k}$

Damping Ratio

The damping ratio ζ is a dimensionless quantity characterizing the rate at which an oscillation in the system's response decays due to effects such as viscous friction or electrical resistance. From the above definitions,

$\zeta = \frac{b}{2\sqrt{km}}$

Natural Frequency

The natural frequency ω_n is the frequency (in rad/s) that the system will oscillate at when there is no damping, $\zeta = 0$.

$\omega_n = \sqrt{\frac{k}{m}}$

Poles / Zeros

The canonical second-order transfer function has two poles at:

$s_p = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$

Underdamped Systems

If $\zeta < 1$, then the system is **underdamped**. In this case, both poles are complex-valued with negative real parts; therefore, the system is stable but oscillates while approaching the steady-state value. Specifically, the natural response oscillates with the damped natural frequency, ω_d=$\omega_n\sqrt{1-\zeta^2}$ (in rad/sec).

```
k_dc = 1;
w_n = 10;
zeta = 0.2;

num = [k_dc*w_n^2];
den = [1 2*zeta*w_n w_n^2];

G1 = tf(num,den)

figure
pzmap(G1)
axis([-3 1 -15 15])

figure
step(G1)
```

Overdamped Systems

If $\zeta > 1$, then the system is **overdamped**. Both poles are real and negative; therefore, the system is stable and does not oscillate. The step response and a pole-zero map of an overdamped system are calculated below:

```
k_dc = 1;
w_n = 10;
zeta = 1.2;

num = [k_dc*w_n^2];
den = [1 2*zeta*w_n w_n^2];

G2 = tf(num,den)

figure
pzmap(G2)
axis([-20 1 -1 1])
```

```
figure
step(G2)
```

Critically-Damped Systems

If $\zeta = 1$, then the system is **critically damped**. Both poles are real and have the same magnitude, $s_p = -\zeta\omega_n$. For a canonical second-order system, the quickest settling time is achieved when the system is critically damped. Now change the value of the damping ratio to 1, and re-plot the step response and pole-zero map.

```
k_dc = 1;
w_n = 10;
zeta = 1;

num = [k_dc*w_n^2];
den = [1 2*zeta*w_n w_n^2];

G3 = tf(num,den)

figure
pzmap(G3)
axis([-20 1 -1 1])

figure
step(G3)
```

Undamped Systems

If $\zeta = 0$, then the system is **undamped**. In this case, the poles are purely imaginary; therefore, the system is marginally stable and the step response oscillates indefinitely.

```
k_dc = 1;
w_n = 10;
zeta = 0;

num = [k_dc*w_n^2];
den = [1 2*zeta*w_n w_n^2];

G4 = tf(num,den)

figure
pzmap(G4)
axis([-20 1 -1 1])

figure
step(G4)
```

```
axis([0 5 -0.5 2.5])
```

Plotting the Root Locus of a Transfer Function

Consider an open-loop system which has a transfer function of:

$$H(s) = \frac{Y(s)}{U(s)} = \frac{(s+7)}{s(s+5)(s+15)(s+20)}$$

```
num = [1 7];
den = conv(conv([1 0],[1 5]),conv([1 15],[1 20]));

H = tf(num,den)

rlocus(H)
axis([-22 3 -15 15])
```

In our problem, we need an overshoot less than 5% (which means a damping ratio of greater than 0.7) and a rise time of 1 second (which means a natural frequency greater than 1.8). Enter the following in the MATLAB command window:

```
sgrid(0.7,1.8)
```

To make the overshoot less than 5%, the poles have to be in between the two angled dotted lines, and to make the rise time shorter than 1 second, the poles have to be outside of the dotted semicircle. So now we know what part of the root locus, which possible closed-loop pole locations, satisfy the given requirements. All the poles in this location are in the left-half plane, so the closed-loop system will be stable.

From the plot above we see that there is part of the root locus inside the desired region. Therefore, in this case, we need only a proportional controller to move the poles to the desired region. You can use the `rlocfind` command in MATLAB to choose the desired poles on the locus:

```
[k,poles] = rlocfind(H)
```

In order to verify the step response, you need to know the closed-loop transfer function.

```
sys_cl = feedback(k*H,1)
```