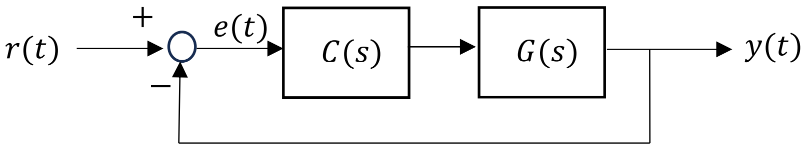
Week 6 Slides

Last time we looked at using feedback to change pole locations



For example, in a unity feeback system the "open-loop transfer function" consists of controller and plant (i.e. remove the feedback loop)

$$C(s)G(s) = \frac{N(s)}{D(s)}$$

and the closed-loop transfer function is

$$T = \frac{N/D}{1 + N/D} = \frac{N}{D + N}$$

We focused on where to put the (closed-loop) poles, and a little on the effect of transfer function zeros. Both affect the response of the system (open-loop or closed-loop).

In passing we noted that the steady-state error for the unity feedback is not always zero. And we have mostly considered the case where the reference input $r(t)$ is a unit step function. Why is this?

Of course, other references are possible: sinusoid, triangle, square, ramp, parabola, ….

For example we might sometimes want to track a ramp-type function, as when tracking a moving object.

Different signals lead to different types of error in the steady-state. And for sinusoidal references we need to understand frequency response functions in more detail, and the use of **bode plots**.

Steady-State Error

We previously mentioned the **order** of a transfer function
e.g.

$$G(s) = \frac{N(s)}{D(s)} = \frac{3s+1}{s^4+3s^3+s} = \frac{1}{s} \cdot \frac{3s+1}{s^3+3s^2+1}$$

This is a fourth order system, being the highest power of s in the numerator.

For steady-state error the **type** of the transfer function is also important – it's the number of integrator terms $\frac{1}{s}$ in the formula. The above example is type 1, having a single integrator.

Looking at the denominator polynomial

- order = highest power of s
- type = lowest power of s

A transfer function is called **proper** if the order of $N(s)$ is no more than the order of $D(s)$. Otherwise it is “top heavy”, or “improper” with higher powers on top.

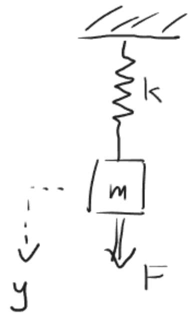
Real-world systems always have proper transfer functions, but sometimes simple models can look top-heavy: for example a PID controller …

$$C(s) = K_p + K_d s + \frac{K_i}{s} = \frac{K_p s + K_d s^2 + K_i}{s}$$

This is order 1 and type 1 (but also, weirdly, improper)

An electric motor with voltage input and angle output is a type 1 system (why?).

For a mass-spring damper,



$$F - ky = m\ddot{y}$$

so

$$m\frac{d^2y}{dt^2} + ky = F$$

What are the order and type of the corresponding transfer function, where force is input and deflection is the output?

PID control converts a type 0 plant $G(s)$ into a type 1 open-loop system due to the integrator.

Steady-state error

To understand this we need the **final value theorem** for Laplace transforms:

For a signal $y(t)$ with Laplace transform $Y(s)$, the steady-state value (final value) is given by

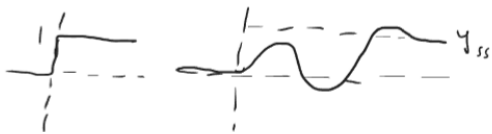
$$y_{ss} = \lim_{t \to \infty} y(t) = \lim_{s \to 0} sY(s)$$

Proving this is a tutorial question.

We can check that it makes sense from the fact that $G(0)$ is the DC gain.

$$U \rightarrow \boxed{G} \rightarrow Y$$

So if $u(t) = 1(t)$, $y_{ss} = G(0)$



In Laplace terms, $Y(s) = G(s) \times 1/s$, so the final value theorem gives

$$y_{ss} = \lim_{s \to 0} sY(s) = \lim_{s \to 0} sG(s) \times 1/s = G(0)$$

The correct answer!

Note that steady-state gain applies to a <u>system</u> with an input that settles to a constant value, while the final value theorem applies to a signal that settles to a constant value.

If $y(t)$ signal does not settle is this way the answer can be wrong or meaningless. For example $y(t) = \sin \omega t$, the FVT gives

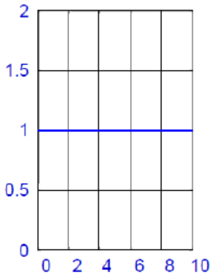
$$y_{ss} = \lim_{s \to 0} s \cdot \frac{\omega}{s^2 + \omega^2} = 0$$

which is clearly wrong, the signal does not settle.

For a closed-loop control system we often consider three particular test signals for the reference (apart from the sinusoid, which comes later)

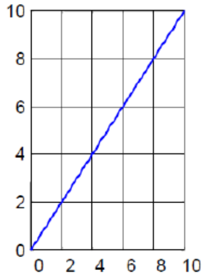
- unit step: $r(t) = 1(t)$ $\qquad R(s) = 1/s$
- unit ramp: $r(t) = t \times 1(t)$ $\qquad R(s) = 1/s^2$
- unit acceleration: $r(t) = \frac{1}{2}t^2 \times 1(t)$ $\qquad R(s) = 1/s^3$

$r(t) = 1u(t)$



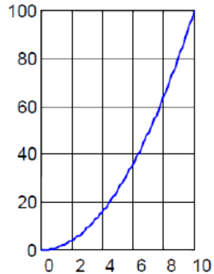
Unit Step Input

$r(t) = t$



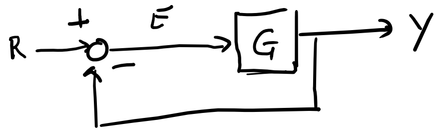
Unit Ramp Input

$r(t) = \frac{1}{2}t^2$



Unit Parabolic Input

Consider the error transfer function. For convenience we lump the controller and plant together as a single transfer function $G(s)$. In the forward path from R to E the transfer function is just 1.



$$\frac{E}{R} = \frac{1}{1+G}$$

Hence, for a step input in the reference,

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} s \cdot \frac{1}{1+G(s)} \cdot \frac{1}{s} = \frac{1}{1+G(0)}$$

Define the '**position error constant**'

$$K_p = \lim_{s\to 0} G(s)$$

(this is also the DC gain of the open-loop system). Then

$$\boldsymbol{e_{ss} = \frac{1}{1 + K_p}}$$

For a **ramp** reference

$$e_{ss} = \lim_{s\to 0} s \cdot \frac{1}{1 + G(s)} \cdot \frac{1}{s^2} = \lim_{s\to 0} \frac{1}{s + sG(s)} = \lim_{s\to 0} \frac{1}{sG(s)}$$

So, defining the **velocity error constant**

$$K_v = \lim_{s\to 0} sG(s)$$

the steady-state error for a ramp is

$$\boldsymbol{e_{ss} = \frac{1}{K_v}}$$

In the same way, for the unit acceleration (parabolic) reference, we have the **acceleration error constant**

$$K_a = \lim_{s\to 0} s^2 G(s)$$

and

$$\boldsymbol{e_{ss} = \frac{1}{K_a}}$$

The error constants may turn out to be zero or infinite, or possibly a finite value. For these simple references we can interpret the results in all cases.

The result mainly depends on how many poles there are at $s = 0$, i.e. the **system type**.

For a type 0 system, K_p is finite, but K_v and K_a are both zero, because of the extra s or s^2 factor.

For type 1 K_v is finite, for type 2 is K_a is finite

SUMMARY TABLE

System Type \ Input	Step	Ramp	Parabolic
0	$\frac{1}{1+K_p}$	∞	∞
1	0	$\frac{1}{K_v}$	∞
2	0	0	$\frac{1}{K_a}$

Notes

- calculate the K's from the open-loop TF
- errors apply to the closed-loop TF
- "zero means zero"
- ∞ means the error simply grows
- don't confuse these K's with controller gains!!!!!

Example 1 – for a plant with transfer function

$$G(s) = \frac{3s + 1}{s^3 + 3s^2 + 1}$$

(a) what is the system type? What is the system order?

(b) Assume it is controlled by proportional (P) control, $u(t) = k_p e(t)$ Sketch the block diagram and find the steady-state error for a step input. How does the value of k_p affect the result?

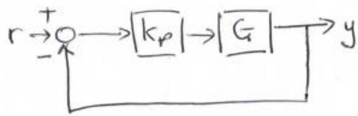
Solution

(a) The type is 0 and the order is 3

(b)

$$CG = \frac{k_p(3s+1)}{s^3+3s^2+1}$$

no change in order or type



$$K_p = \lim_{s\to 0} CG = \frac{k_p}{1} = k_p$$

$$\therefore e_{ss} = \frac{1}{1+K_p} = \frac{1}{1+k_p}$$

As k_p increases, e_{ss} decreases.

As mentioned, don’t confuse the error constant with the feedback gain, they are not always equal!

Tutorial Questions

1. Find the order and type of the following transfer functions:
   (i) $G_1(s) = \frac{s^2+1}{s^2}$ (ii) $G_2(s) = \frac{s+1}{s^2+s+1}$ (iii) $G_3(s) = G_1(s)G_2(s)$.
   Which of these transfer funtions are proper, and which are "strictly proper" – where the order of D is greater than the order of N.

2. Find the order and type of the following
   (i) $C(s) = \frac{K(1+0.1s)}{s+5s^2}$ (ii) $G(s) = \frac{s+1}{s^2+1}$ (iii) $C(s)G(s)$

3. For question 2, find the appropriate (finite) error constant for the OLTF = CG. For what input is the steady-state error finite? Obtain its value in terms of K.

4. Write down a type 1, second order transfer function $C(s)$ with a single zero at $s = -1$, and one of its poles at $s = -5$. Given that the velocity error constant is 100, determine $C(s)$.

5. With the controller of question 4, the plant transfer function $G(s)$ is type zero and has a DC gain $G(0) = 0.2$. Determine the closed-loop steady-state error for a unit ramp reference.
6. For the system in Example 1(b), suppose we use PI control instead of proportional control:

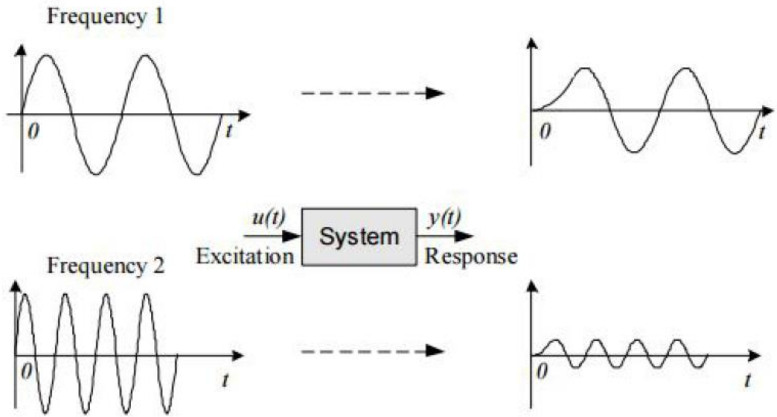
$$u(t) = k_p e(t) + k_i \int e(t)dt$$

   What is the system type for the OLTF and what is the steady-state error in closed loop, assuming (i) a step input, (ii) a ramp input?
7. Test out your result in question 4 by inventing a suitable plant transfer function and running a simulation in Matlab or Simulink.
8. Prove the final value theorem! Consider the Laplace transform for the derivative of $f(t)$: $\int_0^T \dot{f}\, e^{-st}dt$, where T is very very large and now s is very very small. Simplify this integral and compare it to the equation for $\mathcal{L}[\frac{df}{dt}]$.

Frequency Response Function and Plotting
Another “test signal” is the sinusoid. It’s an important concept and we can apply sinusoids to both open-loop and closed-loop signals.

As with the steady-state error equations, analysis of the open-loop system also tells us useful information about the closed-loop system.

We won’t do that analysis here, but just focus on basics and see a strong link between transfer functions and frequency plots.



Comparing input to output determines the **amplitude gain** and the **phase lag** once the sinusoids settle into a steady-state relationship.

Using the complex sinusoid we can relate frequency response to the TF. E.g.,

$$G(s) = \frac{0.1s + 1}{s^2 + s + 10}$$

Then $Y(s) = \frac{0.1s+1}{s^2+s+10} U(s)$, and we can find the differential equation:

$$(s^2 + s + 10)Y(s) = (0.1s + 1)U(s) \Rightarrow \ddot{y} + \dot{y} + 10y = 0.1\dot{u} + u$$

Now assume $u(t) = e^{j\omega t}$, and (in steady-state) $y(t) = He^{j\omega t}$

$$(j\omega)^2 He^{j\omega t} + (j\omega)He^{j\omega t} + 10He^{j\omega t} = 0.1(j\omega)e^{j\omega t} + e^{j\omega t}$$

so

$$[(j\omega)^2 + (j\omega) + 10]H = 0.1(j\omega) + 1$$

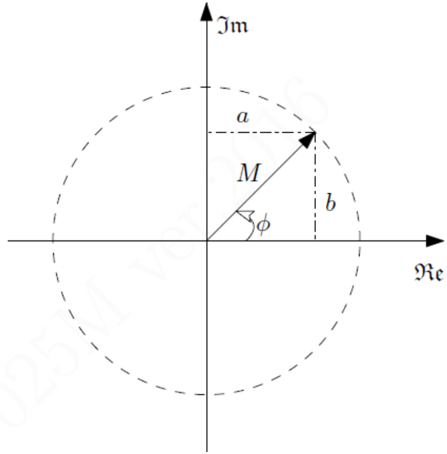
$$H(\omega) = \frac{0.1(j\omega) + 1}{(j\omega)^2 + (j\omega) + 10} = G(j\omega)$$

Once we assume a sinusoidal input, settling to a sinusoidal output, it is obvious that $j\omega$ is now a 'derivative operator'. Here ω is the excitation frequency in radians per second.

We can find the frequency response function by setting $s = j\omega$.

And while $G(s)$ is a theoretical concept, $H(\omega)$ is a very practical one.

The frequency response can be found from *measurement*, from differential equations, or from the *transfer function*.



Typically H is a complex number.

For any given frequency it scales the amplitude and phase from input to output

$$H(\omega) = a + jb = Me^{j\phi}$$

With reference input

$$u(t) = \cos \omega t = Re\{e^{j\omega t}\}$$

the output (after settling) is

$$y(t) = Hu(t) = Me^{j\phi}e^{j\omega t} = Me^{j(\omega t+\phi)}$$

Example

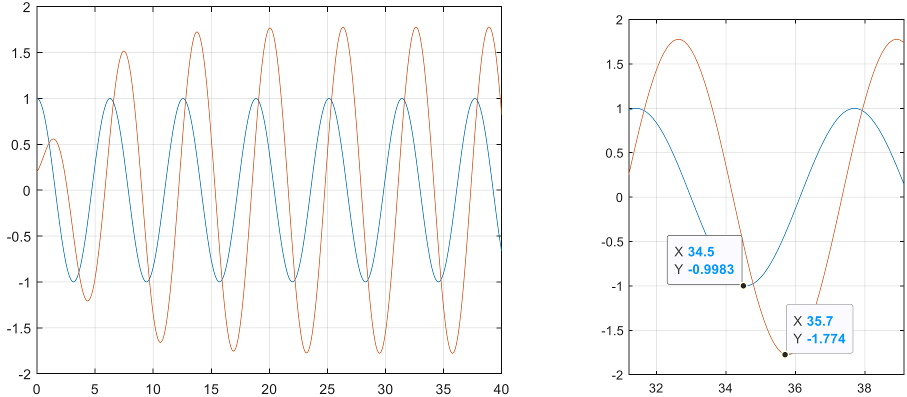
Consider a system with transfer function

$$G(s) = \frac{0.2s^2 + 0.3s + 1}{s^2 + 0.4s + 1}$$

We can use Matlab to find the frequency response at any required frequency. Let us choos $\omega = 1$ rad/sec.

Here we use the Matlab function **lsim** to perform the simulation.

```
G=(0.2*s^2+0.3*s+1)/(s^2+0.48*s+1);
w=1; %input frequency
t=0:0.1:40;
u=cos(w*t);
y=lsim(G,u,t);
figure(1)
plot(t,u),grid,hold on
plot(t,y)
```



The amplitude scaling in steady state is around 1.8, while the phase lag is a around 70 degrees. [check: $\phi \approx \omega(35.7 - 34.5) = 1.2$ radians = 69 deg]

The complex frequency response should be around
$M \approx 1.8$, $\phi \approx -1.21.57$

$$H = 1.8e^{-1.2j} = 0.6522 \; - \; 1.6777i$$

For comparison we can use the transfer function formula

```
S=1j*w;
H=(0.2*S^2+0.3*S+1)/(S^2+0.48*S+1);
M=abs(H);
phi=angle(H);
```

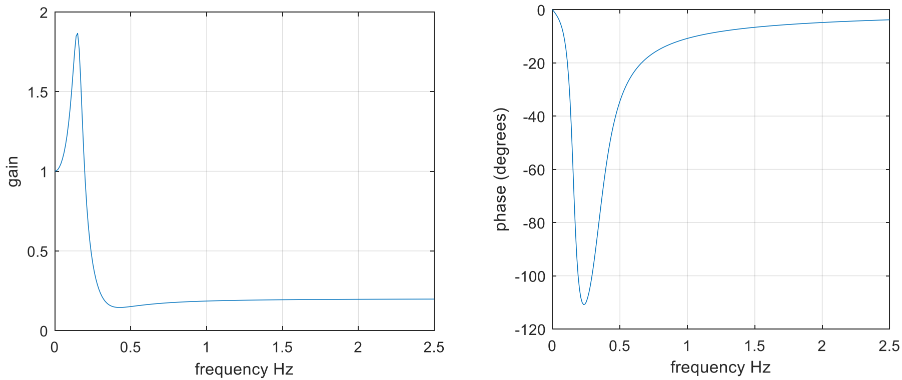
which gives M=1.78, phi= –1.212.

Note

- a phase lag gives a negative value for phi
- using complex numbers is convenient – a single value of H gives both the amplitude scaling and the phase shift
- the gain is found from **abs** in Matlab
- the phase is found from angle

This is just for one frequency, $\omega = 1$ rad/sec. We can easily change this in the above code, but it's more informative to see the overall picture as ω varies.

```
%bode plot done manually
f=0:0.01:2.5; %frequency range in Hz
SS=1j*2*pi*f; %array of S-values to put in the TF formula
H=zeros(size(f)); %empty array for the results
for i=1:length(f)
    S=SS(i);
    H(i)=(0.2*S^2+0.3*S+1)/(S^2+0.48*S+1);
end
figure(2)
plot(f,abs(H)),grid
figure(3)
plot(f,angle(H)*180/pi),grid
```



From this we can see

- the DC gain is 1 (as expected) – when f=0 Hz
- There is a resonance peak around 0.16 Hz (1 radian per second, gain around 1.8 as we checked before)
- The phase lag ranges from 0 to 11 degrees
- At high frequencies the gain and phase become nearly constant

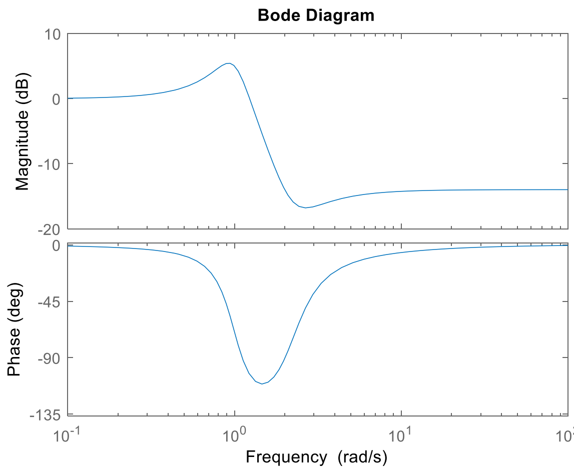
Matlab also has the built-in function **bode** to determine this this. However the gain plot has a default scale of decibels (dB) which is a logarithmic scale:

$$dB = 20 \log_{10} (|H|)$$

The command is simply **bode(G)**, once the transfer function has been defined

frequency is presented on a log scale also

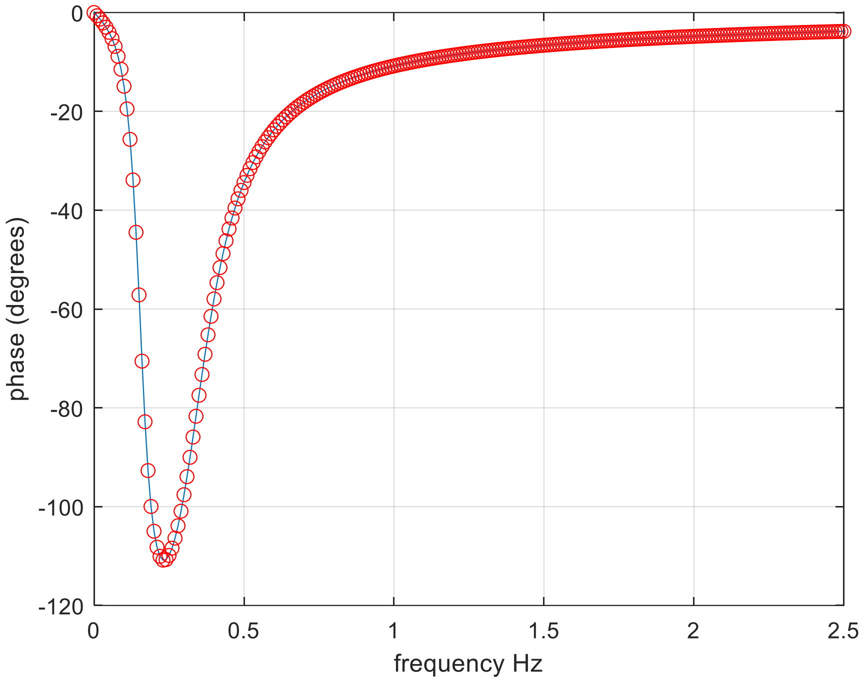
the result usually looks nice but it may be harder to read!



bode can be used with input and output arguments to 'take control' of the plotting. He we plot the phase on top of the previous phase figure.

The command **squeeze** is used – you can check out why by using **doc** or just deleting them and see what goes wrong!

```
%bode with input and output arguments
W=2*pi*f; %array of frequencies in rad/sec
[mag,phase]=bode(G,W);
mag=squeeze(mag);phase=squeeze(phase);
figure(3), hold on
plot(f,phase,'ro')
```


0
-20
-40
-60
-80
-100
-120
phase (degrees)
0
0.5
1
1.5
2
2.5
frequency Hz

Summary

This week we have studied a more general version of ‘steady state’ response, including steady-state error and steady-state gain/phase shift.

- not just step inputs
- system order and type
- final value theorem for signals
- **error constants from the OLTF**
- **closed-loop steady-state error from error constants**
- sinusoidal signals
- $s = j\omega$
- $H = Me^{j\phi}$
- Bode plots
- DC gain from $\omega = 0$ (same as $s = 0$ in the transfer function)

Tutorial Questions

9. What are the order, type and steady-state gain for the transfer functions:

$$G_1(s) = \frac{0.2s + 1}{s^3 + 0.4s} \quad , \qquad G_2(s) = \frac{0.2s^2 + 1}{1 + 0.4s}$$

10. Use Matlab to give the Bode plots of the above transfer functions. Comment on what you see. Determine the gain and phase values by hand when ω = 1 rad/sec, and check they agree with the bode plots (you can use the data tip tool).