CDS 110/ChE 105: Lecture 7-1
Small Signal (Freq Domain) Analysis

Richard M. Murray
13 May 2024

Goals:

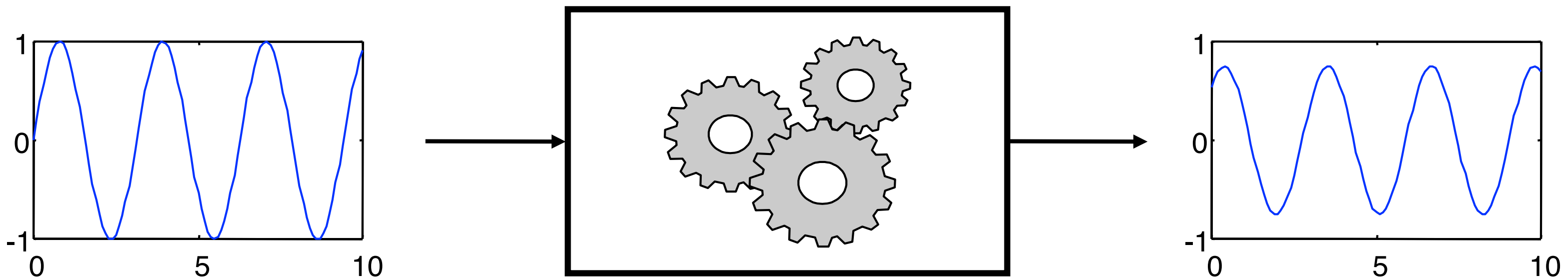
- Define the input/output transfer function of a linear system
- Describe how to use Bode plots to understand frequency response
- Understand relationships between frequency response, transfer function, and state-space model

Reading:

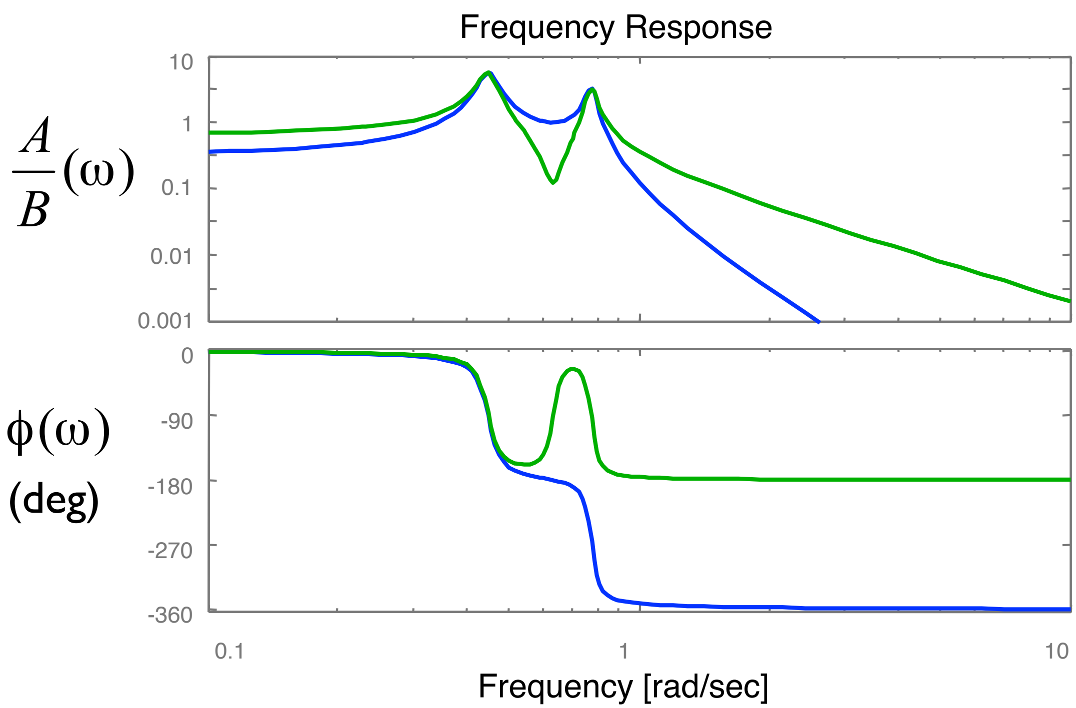
- Åström and Murray, *Feedback Systems*, Sec 9.1-9.4

Frequency Domain Modeling

Defn. The frequency response of a linear system is the relationship between the gain and phase of a sinusoidal input and the corresponding steady state (sinusoidal) output.



$u = A\sin(\omega t)$ $\qquad$ $y = B\sin(\omega t + \phi)$



Bode plot (1940; Hendrik Bode)

- Plot gain and phase vs input frequency
- Gain is plotting using log-log plot
- Phase is plotting with log-linear plot
- Can read off the system response to a sinusoid – in the lab or in simulations
- Linearity ⇒ can construct response to any input (via Fourier decomposition)
- Key idea: do all computations in terms of gain and phase (frequency domain)

Transfer Function and Frequency Response

Exponential response of a linear state space system

$$y(t) = Ce^{At}\Big(x(0) - (sI - A)^{-1}B\Big) + \Big(C(sI - A)^{-1}B + D\Big)e^{st}$$

transient steady state

Transfer function

- Steady state response is proportional to exponential input => look at input/output ratio
- $G(s) = C(sI - A)^{-1}B + D$ is the *transfer function* between input and output

Frequency response

$$u(t) = A\sin\omega t = \frac{A}{2i}(e^{i\omega t} - e^{-i\omega t})$$

$$y_{ss}(t) = \frac{A}{2i}\Big(G(i\omega)e^{i\omega t} - G(-i\omega)e^{-i\omega t}\Big)$$
$$= A \cdot |G(i\omega)| \sin(\omega t + \arg G(i\omega))$$

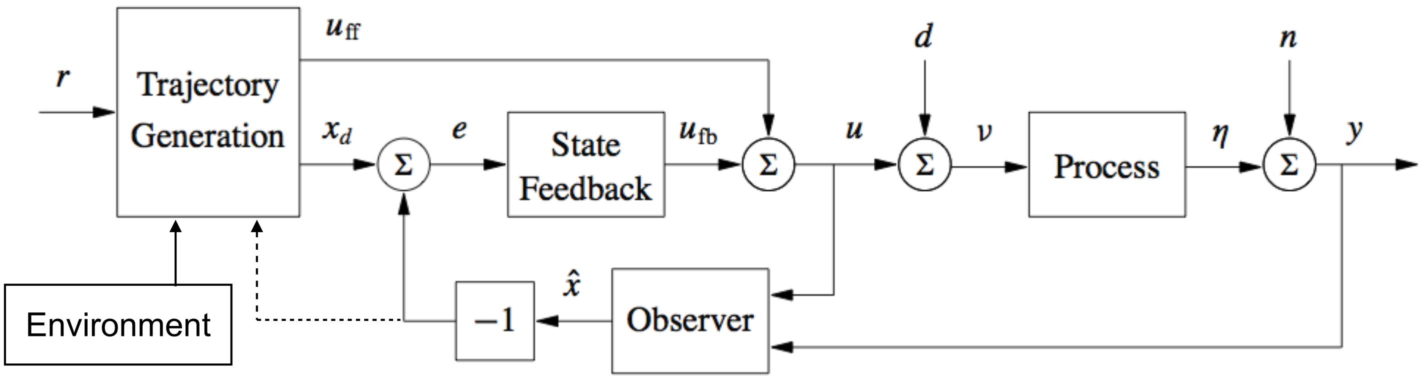
gain phase

Common transfer functions

$\dot{y} = u$	$\frac{1}{s}$
$y = \dot{u}$	s
$\dot{y} + ay = u$	$\frac{1}{s+a}$
$\ddot{y} = u$	$\frac{1}{s^2}$
$\ddot{y} + 2\zeta\omega_n\dot{y} + \omega_n^2 y = u$	$\frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2}$
$y = k_p u + k_d \dot{u} + k_i \int u$	$k_p + k_d s + \frac{k_i}{s}$
$y(t) = u(t - \tau)$	$e^{-\tau s}$

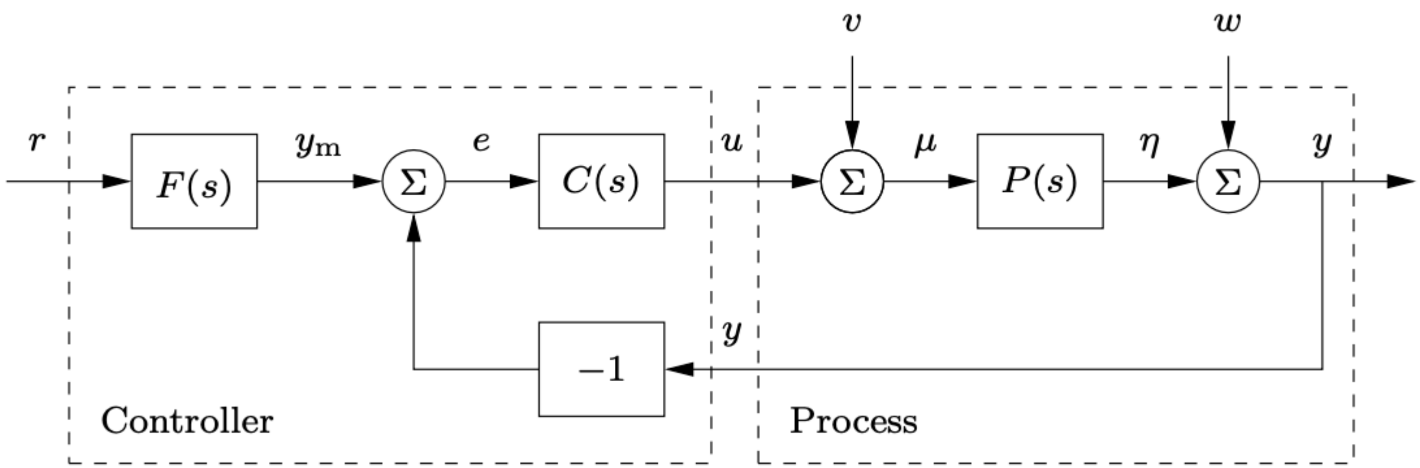
Small Signal Analysis Using Transfer functions

State feedback design pattern



- Allows (optimization-based) high performance trajectory generation and tracking
- Can be difficult to understand stability and robustness except via simulation…

Small signal approach: assume signals are small, analyze linear systems in frequency domain



- Focus on operation near equilibrium or along a trajectory
- Replace (nonlinear) input/output maps with transfer functions
- Enables robustness analysis via frequency bounds (next week)

- Early control theory *started* with this design pattern, driven by early advances in electronics
  - Use exponential response (Laplace transforms) to model dynamics and control
  - Use "loop shaping" (frequency domain design) for design: performance vs robustness

Transfer Function Properties

Def. The transfer function for a linear system Σ=(A,B,C,D) is given by

$$G(s) = C(sI - A)^{-1}B + D \qquad s \in \mathbb{C} \qquad\qquad G(s) = \frac{n(s)}{d(s)} \qquad d(s) = \det(sI - A)$$

Thm. The transfer function $G(s)$ corresponding to $\Sigma = (A,B,C,D)$ has the following properties:

- $G(s)$ is a ratio of polynomials $n(s)/d(s)$ where $d(s)$ is the characteristic equation for the matrix A and $n(s)$ has order less than or equal to $d(s)$.
- The steady state frequency response of Σ has gain $|G(j\omega)|$ and phase arg $G(j\omega)$:

$$u = A\sin(\omega t)$$
$$y = |G(i\omega)| A \sin(\omega t + \arg G(i\omega)) + \text{transients}$$

Remarks

- Formally, can show that $G(s)$ is the Laplace transform of the impulse response of Σ
- Often write "y = $G(s)$ u" for $Y(s) = G(s)\, U(s)$, where $Y(s)$ and $U(s)$ are Laplace transforms of $y(t)$ and $u(t)$. (Multiplication in Laplace domain corresponds to convolution.)
- Python: G = ct.ss2tf(A, B, C, D)

Series Interconnections

Q: what happens when we connect two systems together in series?

$u_1 \rightarrow$

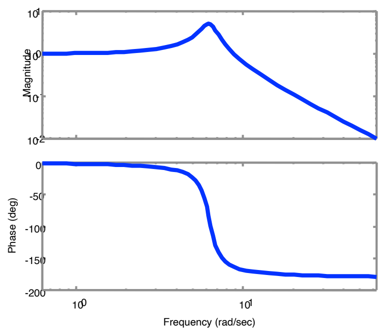
$$\dot{x}_1 = A_1 x_1 + B_1 u_1 \qquad y_1 = C_1 x_1 + D_1 u_1$$

$\rightarrow y_1 \quad u_2 \rightarrow$

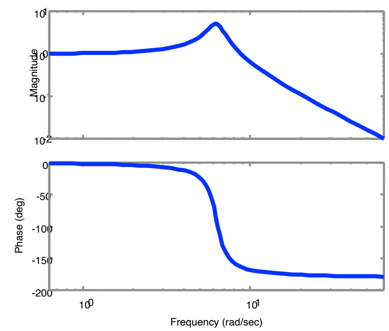
$$\dot{x}_2 = A_2 x_2 + B_2 u_2 \qquad y_2 = C_2 x_2 + D_2 u_2$$

$\rightarrow y_2$

$$u = A\sin(\omega t) \rightarrow$$



$$y = g_1 A \sin(\omega t + \phi_1) \rightarrow$$



$$y_2 = g_1 g_2 A \times \sin(\omega t + \phi_1 + \phi_2) \rightarrow$$

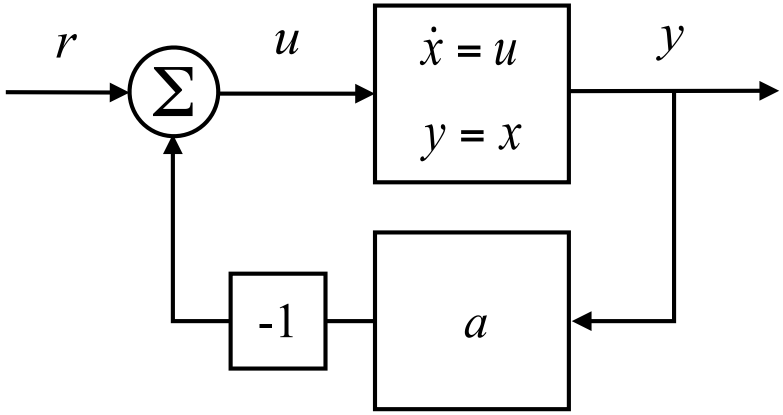
A: Transfer functions *multiply*

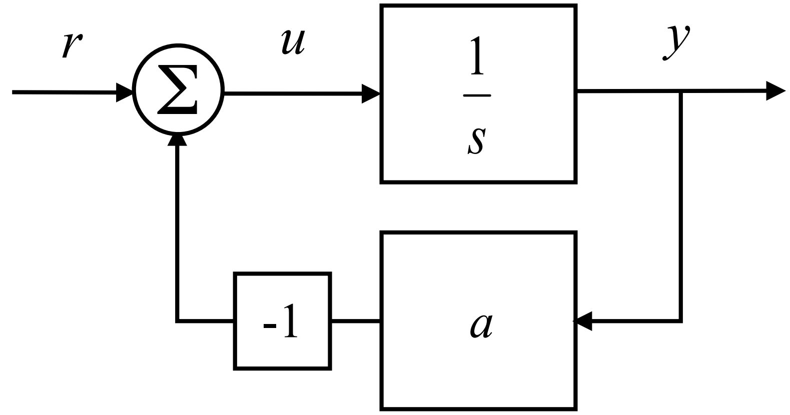
- Gains multiply, phases add
- Generally: transfer functions well formulated for frequency domain interconnections
  - Convolution → multiplication
- Python: G = series(G1, G2) or G = G2 * G1

$$u_1 \rightarrow \boxed{G_1(s)} \xrightarrow{u_2 = y_1} \boxed{G_2(s)} \xrightarrow{y_2}$$

$$u_1 \rightarrow \boxed{G_2(s)G_1(s)} \rightarrow y_2 = G_2(s)G_1(s)u_1$$

Feedback Interconnection





State space derivation

$$\dot{x} = u = r - ay = -ax + r$$
$$y = x$$

Frequency response $\quad r = A\sin(\omega t)$

$$y = \left|\frac{1}{\sqrt{a^2 + \omega^2}}\right| \sin\left(\omega t - \tan^{-1}(\frac{\omega}{a})\right)$$

Transfer function derivation

$$y = \frac{u}{s} = \frac{r - ay}{s}$$
$$y = \frac{r}{s + a} = G(s)r$$

Frequency response

$$y = |G(i\omega)| \sin(\omega t + \angle G(i\omega))$$

- Python: G = ct.feedback(sys, a) $\leftarrow$ works for either state space or transfer functions

Poles and Zeros

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

$$G(s) = \frac{n(s)}{d(s)}$$
$$d(s) = \det(sI - A)$$

- Roots of $d(s)$ are called *poles* of $G(s)$
- Roots of $n(s)$ are called *zeros* of $G(s)$

Poles of G(s) determine the stability of the (closed loop) system

- Denominator of transfer function = characteristic polynomial of state space system
- Provides easy method for computing stability of systems
- Right half plane (RHP) poles (Re > 0) correspond to unstable systems

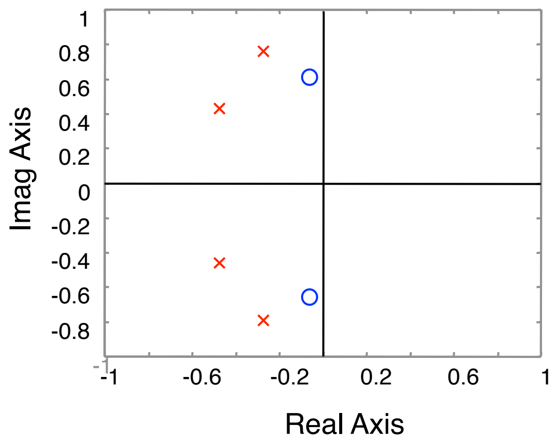
Zeros of G(s) related to frequency ranges with limited transmission

- A pure imaginary zero at s = iω blocks any output at that frequency (G(iω) = 0)
- Zeros provide limits on performance, especially RHP zeros (more on this later)

Python: G.poles(), G.zeros(), ct.pole_zero_plot(G)

$$G(s) = k\frac{s^2 + b_1 s + b_2}{s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4}$$

pole_zero_plot(G) →



Sketching the Bode Plot for a Transfer Function (1/2)

Evaluate transfer function on imaginary axis

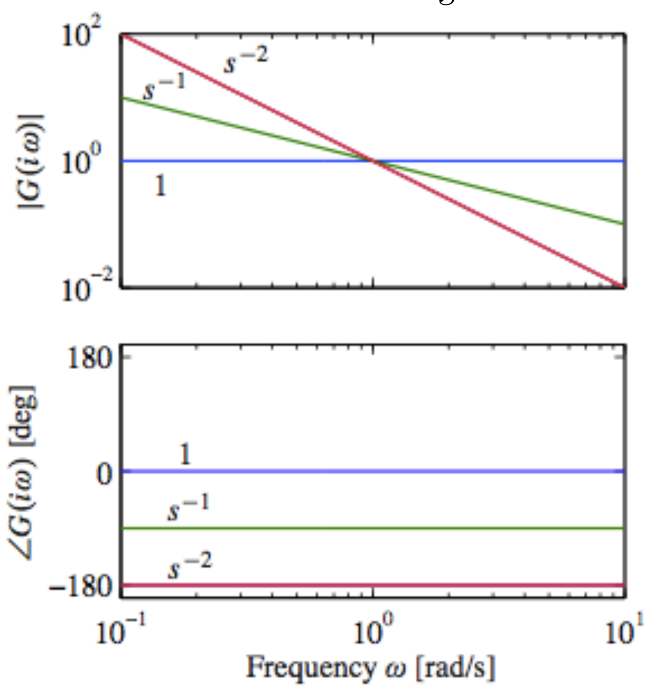
$$M = |G(i\omega)|, \qquad \varphi = \arctan \frac{\operatorname{Im} G(i\omega)}{\operatorname{Re} G(i\omega)}$$

- Plot gain (M) on log/log scale
- Plot phase (φ) on log/linear scale
- Piecewise linear approximations available

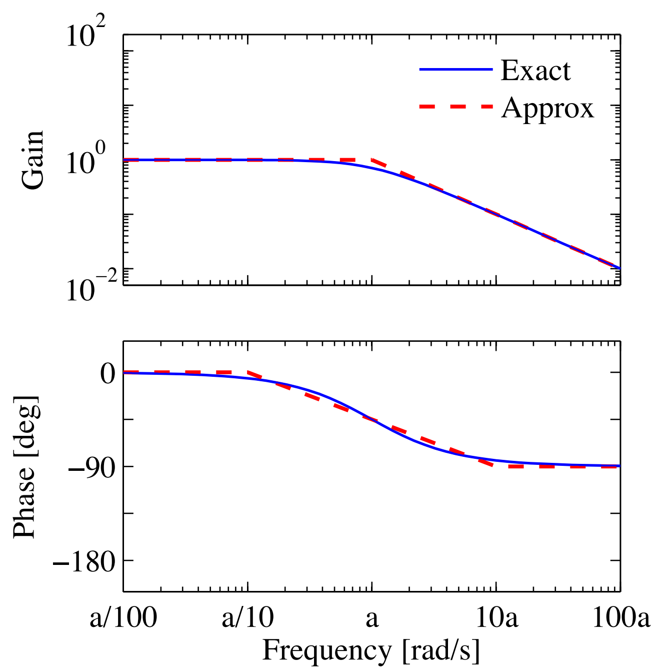
$$\log |G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega < a \\ \log a - \log \omega & \text{if } \omega > a, \end{cases}$$

$$\angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega < a/10 \\ -45 - 45(\log \omega - \log a) & a/10 < \omega < 10a \\ -90 & \text{if } \omega > 10a. \end{cases}$$

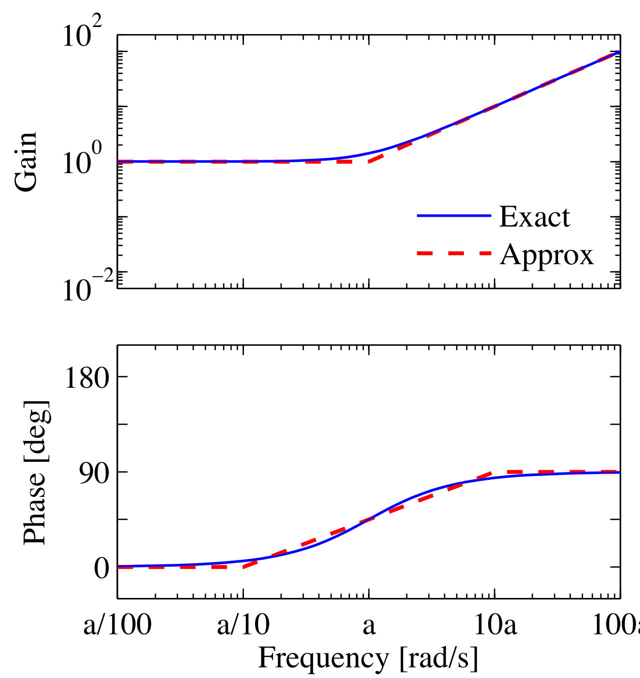
$$G(s) = \frac{1}{s^k}$$



$$G(s) = \frac{a}{s+a}$$



$$G(s) = \frac{s+a}{a}$$



Sketching the Bode Plot for a Transfer Function (2/2)

Complex poles $G(s) = \dfrac{\omega_0^2}{s^2 + 2\omega_0\zeta s + \omega_0^2}$

$$\log|G(i\omega)| \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0 \\ 2\log\omega_0 - 2\log\omega & \text{if } \omega \gg \omega_0, \end{cases}$$

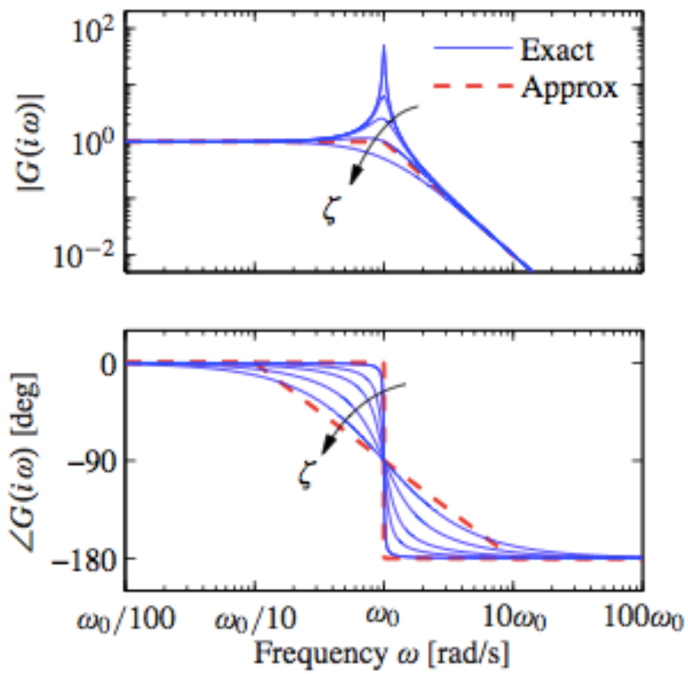
$$\angle G(i\omega) \approx \begin{cases} 0 & \text{if } \omega \ll \omega_0 \\ -180 & \text{if } \omega \gg \omega_0. \end{cases}$$

Ratios of products $G(s) = \dfrac{b_1(s)b_2(s)}{a_1(s)a_2(s)}$

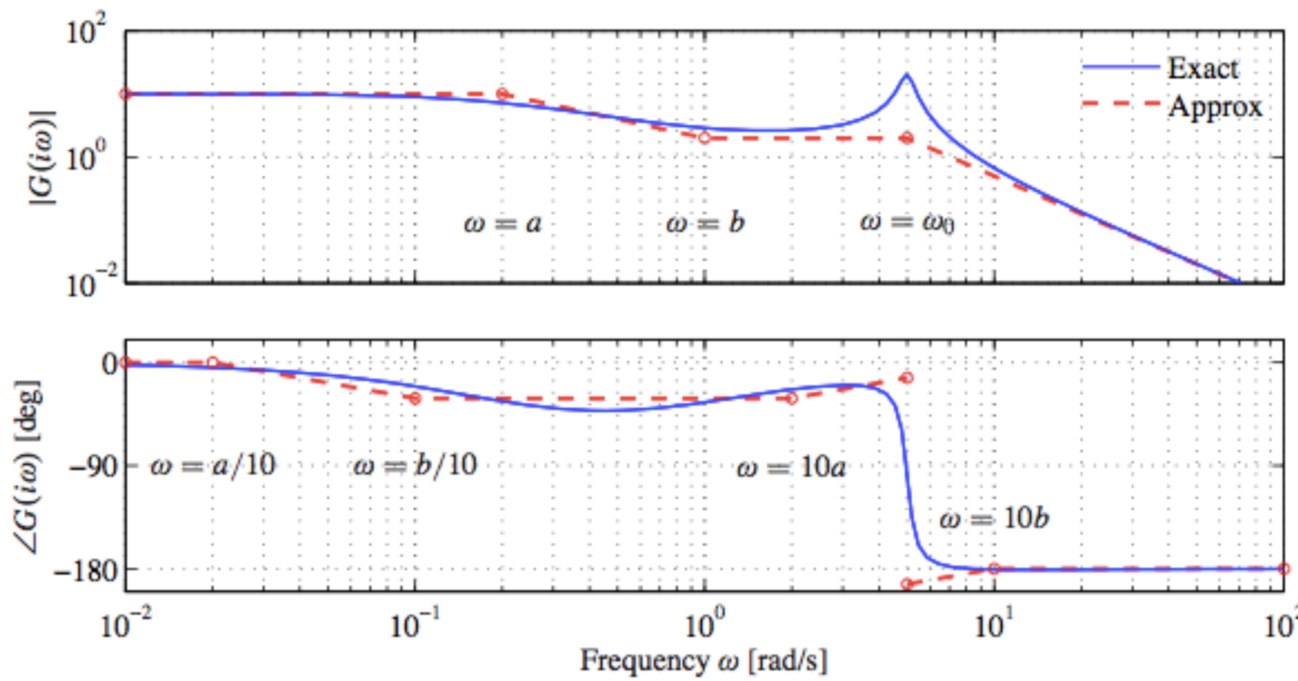
$$\log|G(s)| = \log|b_1(s)| + \log|b_2(s)| - \log|a_1(s)| - \log|a_2(s)|$$

$$\angle G(s) = \angle b_1(s) + \angle b_2(s) - \angle a_1(s) - \angle a_2(s)$$

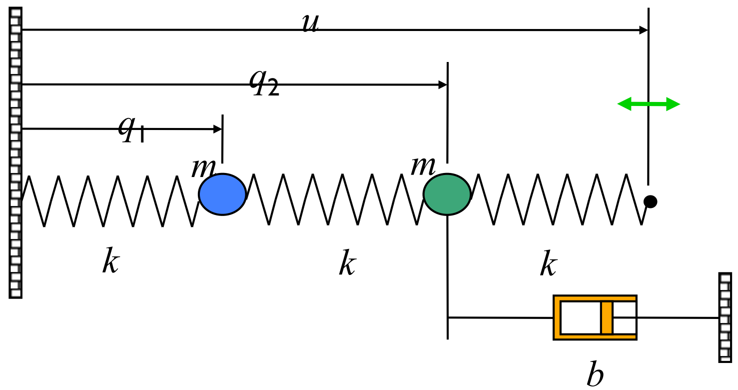
$$G(s) = \frac{\omega_0^2}{s^2 + 2\omega_0\zeta s + \omega_0^2}$$



$$G(s) = \frac{k(s+b)}{(s+a)(s^2 + 2\zeta\omega_0 s + \omega_0^2)}, \qquad a \ll b \ll \omega_0.$$



Example: Coupled Masses



$$H_{q_1 f} = \frac{0.04}{s^4 + 0.08s^3 + 0.8016s^2 + 0.032s + 0.12}$$

$$H_{q_2 f} = \frac{0.2s^2 + 0.008s + 0.08}{s^4 + 0.08s^3 + 0.8016s^2 + 0.032s + 0.12}$$

Poles (Hq1f and Hq2f)

- -0.0200 ± 0.7743j
- -0.0200 ± 0.4468j

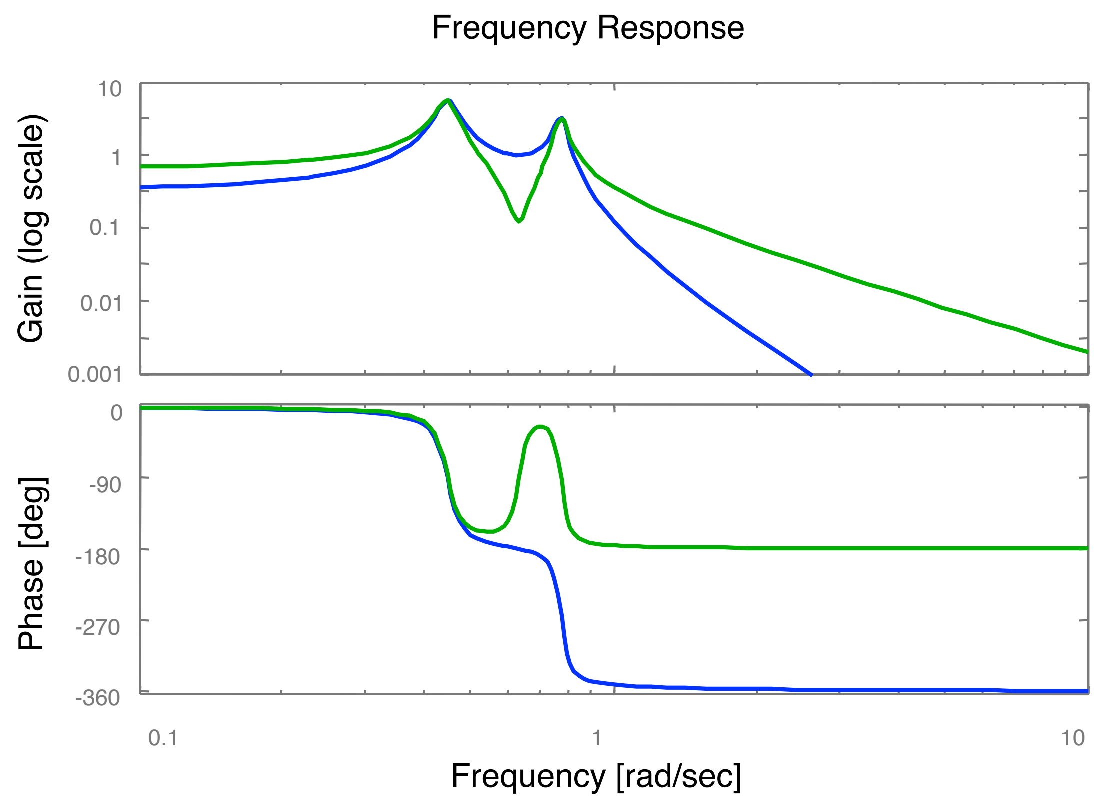
Zeros (Hq2f)

- -0.0200 ± 0.6321j

Interpretation

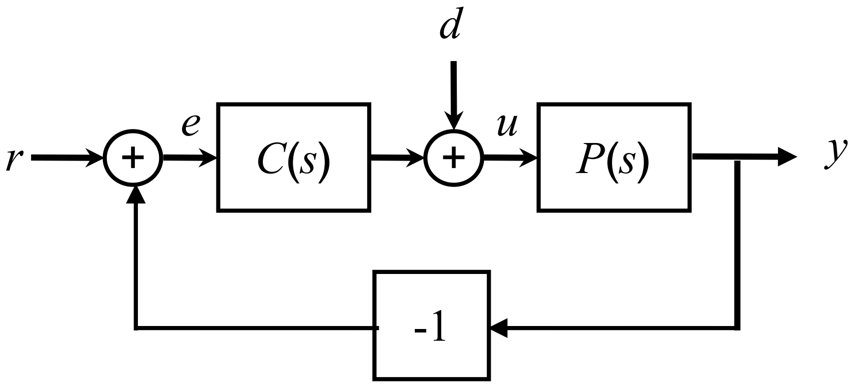
- Zeros in Hq2f give low response at ω ≈ 0.6321

Python: ct.bode_plot(H) or ct.frequency_response(H).plot()



Block Diagram Algebra

Basic idea: treat transfer functions as multiplication, write down equations



$$y = P(s)u$$
$$u = d + C(s)e$$
$$e = r - y$$

Manipulate equations to compute desired signals

$$\begin{aligned} e &= r - y \\ &= r - P(s)u \\ &= r - P(s)(d + C(s)e) \end{aligned}$$

$$(1 + P(s)C(s))e = r - P(s)d$$

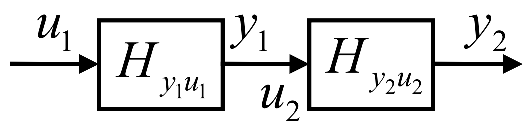
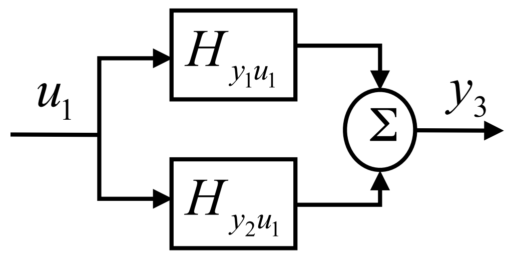
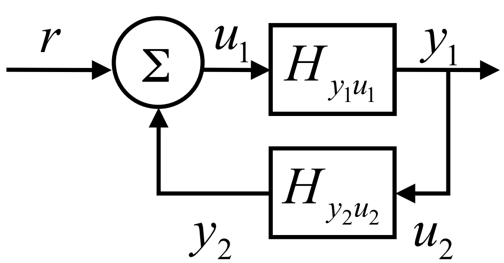
$$e = \underbrace{\frac{1}{1 + P(s)C(s)}}_{H_{er}} r - \underbrace{\frac{P(s)}{1 + P(s)C(s)}}_{H_{ed}} d$$

Note: linearity gives super-position of terms

Algebra works because we are working in frequency domain

- Time domain (ODE) representations are not as easy to work with
- Formally, all of this works because of Laplace transforms (see text)

Summary: Block Diagram Algebra

Type	Diagram	Transfer function
Series		$H_{y_2u_1} = H_{y_2u_2} H_{y_1u_1} = \dfrac{n_1 n_2}{d_1 d_2}$
Parallel		$H_{y_3u_1} = H_{y_2u_1} + H_{y_1u_1} = \dfrac{n_1 d_2 + n_2 d_1}{d_1 d_2}$
Feedback		$H_{y_1r} = \dfrac{H_{y_1u_1}}{1 + H_{y_1u_1} H_{y_2u_2}} = \dfrac{n_1 d_2}{n_1 n_2 + d_1 d_2}$

- These are the basic manipulations needed; some others are possible
- Formally, could work all of this out using the original ODEs (⇒ nothing really new)

Python (and MATLAB) Manipulation of Transfer Functions

Creating transfer functions

- [num, den] = ct.ss2tf(A, B, C, D)
- sys = ct.tf(num, den)
- num, den = [1, a, b] → s2 + a s + b

Interconnecting blocks

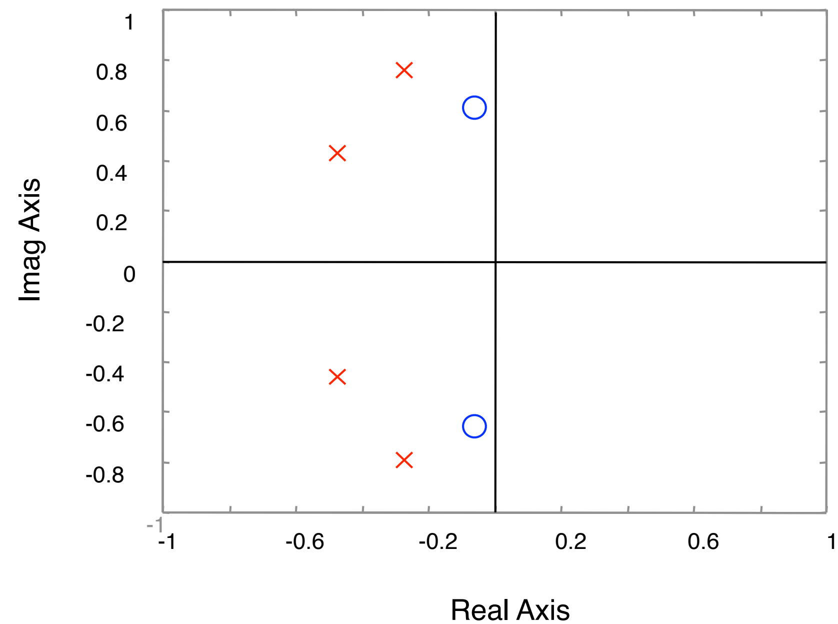
- sys = series(sys1, sys2), parallel, feedback
- Also: sys2 * sys1 (series), sys1 + sys2 (par)

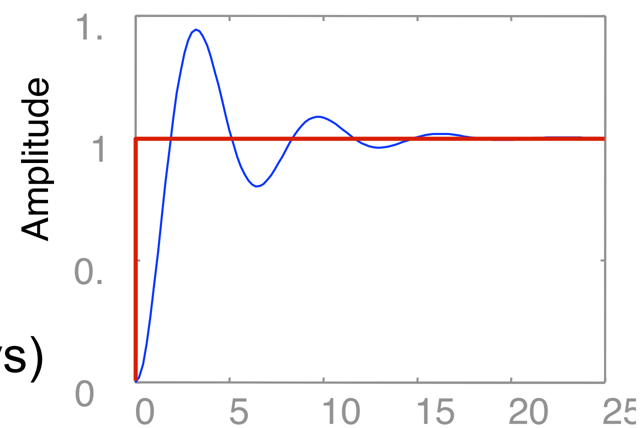
Computing poles and zeros

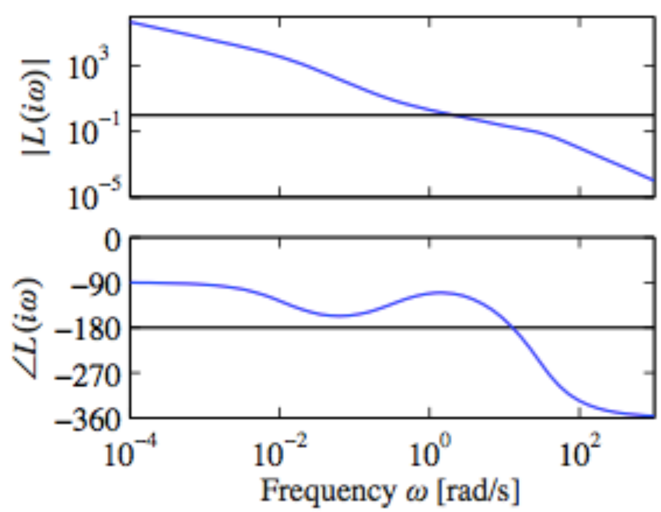
- sys.poles(), sys.zeros()
- pzmap(sys)

I/O response

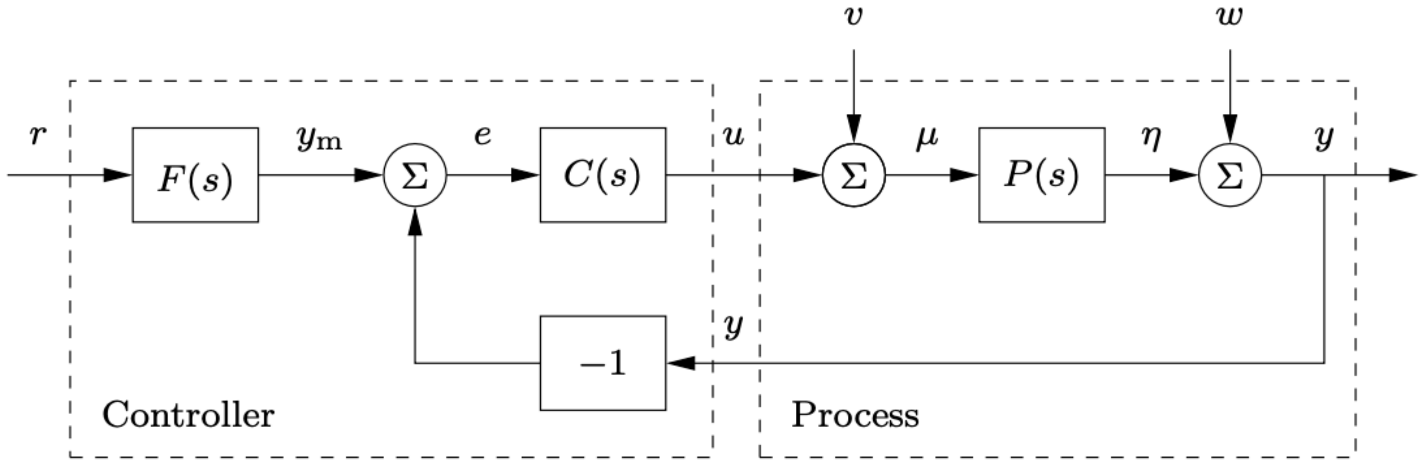
- initial_response(sys)
- step_response(sys)
- forced_response(sys)
- frequency_response(sys)







Control Analysis and Design Using Transfer Functions



Transfer functions provide a method for “block diagram algebra”

- Easy to compute transfer functions between various inputs and outputs
  - Her(s) is the transfer function between the reference and the error
  - Hed(s) is the transfer function between the disturbance and the error

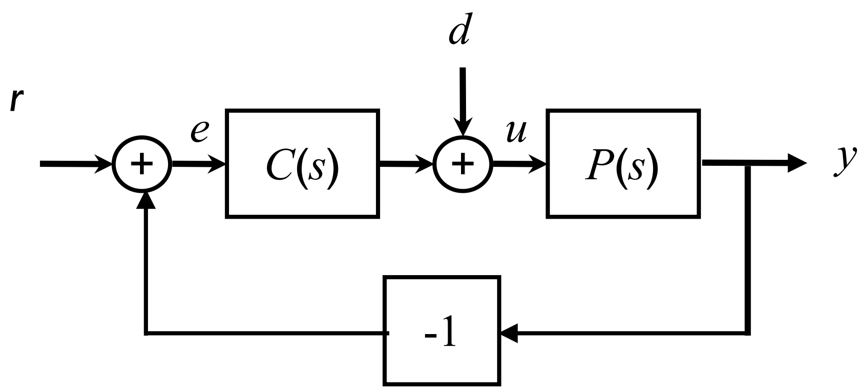
Transfer functions provide a method for performance specification

- Since transfer functions provide frequency response directly, it is convenient to work in the “frequency domain”
  - Her(s) should be small in the frequency range 0 to 10 Hz (good tracking)

Key idea: perform all analysis and design for linear systems in “frequency domain”

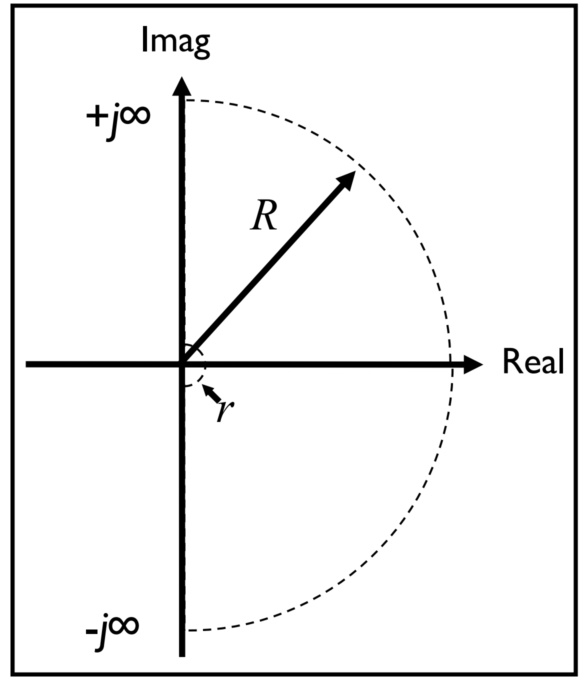
- Convert specifications on time response to specifications on frequency response
- “Shape” the frequency response by design of controller transfer function (Ch 10-13)

Wednesday: Stability via Nyquist Criterion



Determine stability from (open) loop transfer function, L(s) = P(s)C(s).

- Use "principle of the argument" from complex variable theory (see reading)



- Nyquist "D" contour
- Take limit as $r \to 0$, $R \to \infty$
- Trace from –1 to +1 along imaginary axis

Thm (Nyquist). Consider the Nyquist plot for loop transfer function $L(s)$. Let

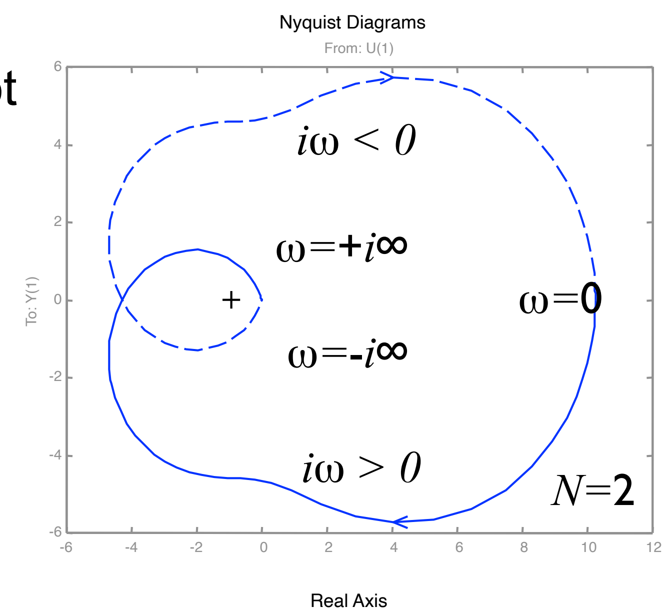
P # RHP poles of $L(s)$

N # clockwise encirclements of -1

Z # RHP zeros of $1 + L(s)$

Then

$$Z = N + P$$



- Trace frequency response for $L(s)$ along the Nyquist "D" contour
- Count net # of clockwise encirclements of the -1 point