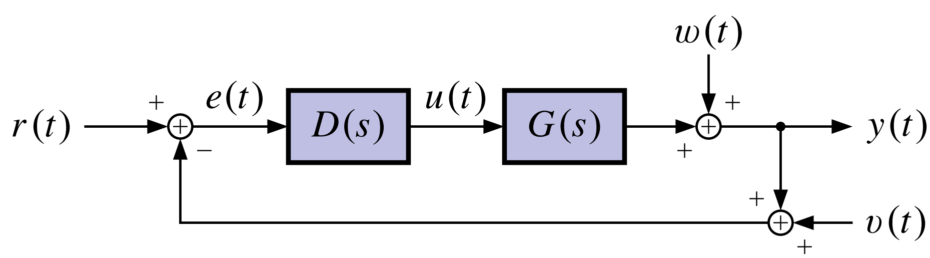
INTRODUCTION TO DIGITAL CONTROL

1.1: Introduction

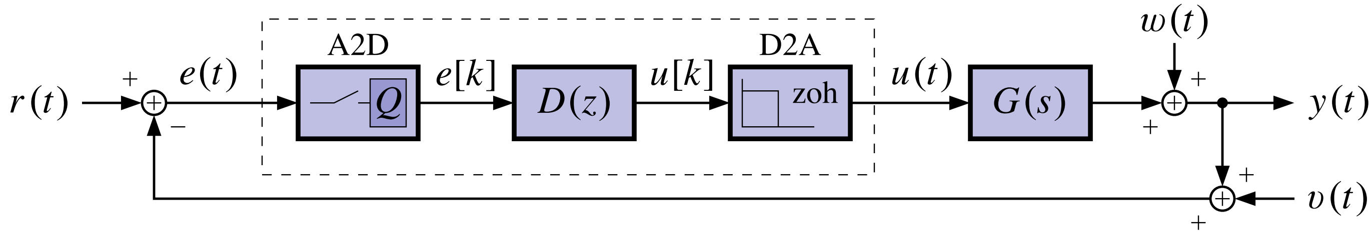
- In *ECE4510/ECE5510 Feedback Control Systems*, we learned how to make an analog controller $D(s)$ to control a linear-time-invariant (LTI) plant $G(s)$.
- We used feedback to:
  - Reduce error due to disturbance,
  - Reduce error due to model mismatch,
  - Stabilize an unstable plant,
  - Improve transient response.



- There are advantages to using a digital computer to replace $D(s)$.
  - Flexible: Modifications to design easy to implement.
  - Accurate: Does not rely on imprecise R, C, … values.
  - Advanced algorithms are possible (nonlinear, adaptive…).
  - Logic statements may be included.
- There are "problems" with using a digital computer:

- $D(s)$ "knows" $e(t)$ at each instant in time but digital computers cannot process input infinitely quickly.
- $D(s)$ outputs $u(t)$ at each instant of time but...
- $e(t)$ is an infinite-precision value, but digital computers must work with finite-precision values.
- Digital computers are finite.

■ The necessary compromises are to "sample" the input $e(t)$ to get $e[k] = e(kT)$, to compute an output $u[k]$ that is held constant from $u(kT)$ to $u((k+1)T)$, and to "quantize" internal computer variables.



■ $D(z)$ is the "transfer function" of the controller; discrete-time.

■ We need to know how to design $D(z)$, how quickly to sample $(1/T)$, and how finely to quantize.

■ Fortunately, many of the techniques are similar either in principle or practice to analog-control techniques. Proceeding, we look at:

- Review.
- Examine impact A2D/D2A with approximate control method.
- Study z-transform.
- More in-depth study of A2D/D2A using z-transform knowledge.
- Control design of $D(z)$ using transform methods.
- Implementation using DSP.
- Adaptive form of digital control called "Adaptive Inverse Control."

1.2: Review: System modeling

Goals of feedback control

- Change dynamic response of a system to have desired properties.
- Ensure that closed-loop system is stable.
- Constrain system output to track a reference input with acceptable transient and steady-state responses.
- Reject disturbances.
- These goals are accomplished via frequency-domain (Laplace) analysis and design tools for continuous-time systems.

Dynamic response

- We wish to control linear time-invariant (LTI) systems.
- These dynamics may be specified via linear, constant-coefficient ordinary differential equations (LCCODE).
- Examples include:
  - Mechanical systems: Use Newton's laws.
  - Electrical systems: Use Kirchoff's laws.
  - Electro-mechanical systems (generator/motor).
  - Thermodynamic and fluid-dynamic systems.

EXAMPLE: Second-order system in "standard form":

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t).$$

- "$u(t)$" is the input and "$y(t)$" is the output.

- $\dot{y}(t) \triangleq \dfrac{dy(t)}{dt}$ and $\ddot{y}(t) \triangleq \dfrac{d^2y(t)}{dt^2}$.

- The *Laplace Transform* is a tool to help analyze dynamic systems. $Y(s) = H(s)U(s)$, where
  - $Y(s)$ is Laplace transform of output $y(t)$;
  - $U(s)$ is Laplace transform of input $u(t)$;
  - $H(s)$ is transfer function—the Laplace tx of impulse response $h(t)$.

KEY IDENTITY: $\mathcal{L}\{\dot{y}(t)\} = sY(s)$ for system initially "at rest."

EXAMPLE: Second-order system:

$$s^2Y(s) + 2\zeta\omega_n sY(s) + \omega_n^2 Y(s) = \omega_n^2 U(s)$$

$$Y(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}U(s).$$

- Transforms for systems with LCCODE representations can be written as $Y(s) = H(s)U(s)$, where

$$H(s) = \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0},$$

and where $n \geq m$ for physical systems.

- These can be represented in MATLAB using vectors of numerator and denominator polynomials:

```
num=[bm  b1 b0];
den=[an  a1 a0];
sys=tf(num,den);
```

- Can also represent these systems by factoring the polynomials into zero-pole-gain form:

$$H(s) = K\frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)}.$$

```
sys=zpk(z,p,k);   % in MATLAB
```

- Input signals of interest include the following:

$$
\begin{array}{rclcrcll}
u(t) & = & k\,\delta(t) & \ldots & U(s) & = & k & \text{impulse} \\
u(t) & = & k\;1(t) & \ldots & U(s) & = & k/s & \text{step} \\
u(t) & = & kt\;1(t) & \ldots & U(s) & = & k/s^2 & \text{ramp} \\
u(t) & = & kt^2/2\;1(t) & \ldots & U(s) & = & k/s^3 & \text{parabola} \\
u(t) & = & k\sin(\omega t)\;1(t) & \ldots & U(s) & = & \dfrac{k\omega}{s^2+\omega^2} & \text{sinusoid}
\end{array}
$$

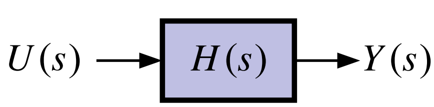
- MATLAB's "`impulse`," "`step`," and "`lsim`" commands can be used to find output time histories.
- The *Final Value Theorem* states that if a system is stable and has a final, constant value,

$$\lim_{t\to\infty} x(t) = \lim_{s\to 0} sX(s).$$

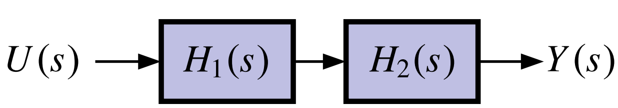
- This is useful when investigating steady-state errors.

Block diagrams

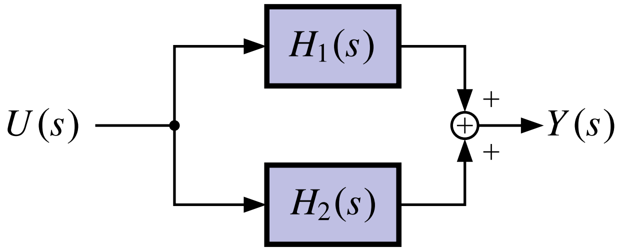
- Useful when analyzing systems comprised of a number of sub-units.



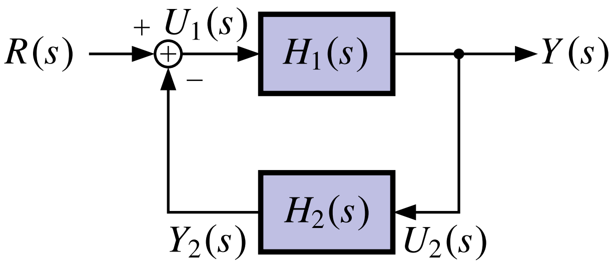
$Y(s) = H(s)U(s)$



$Y(s) = [H_1(s)H_2(s)]\,U(s)$

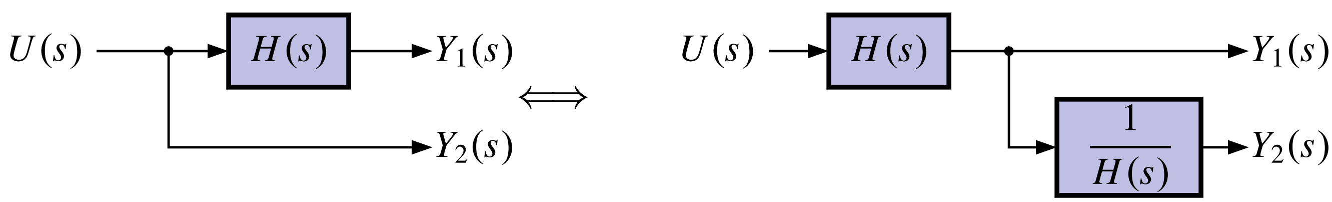


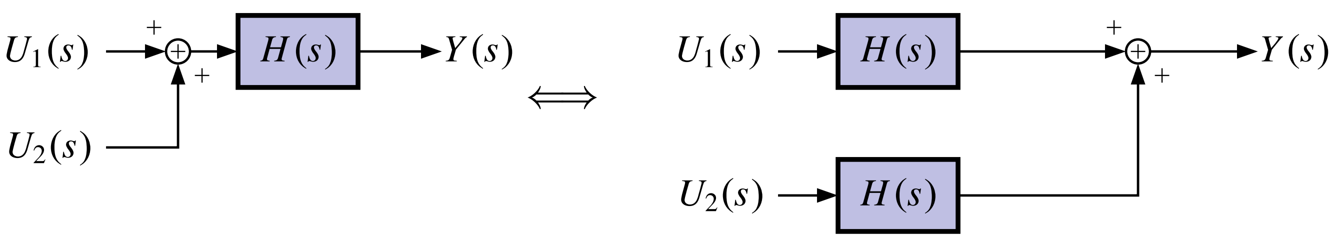
$$Y(s) = [H_1(s) + H_2(s)]\,U(s)$$

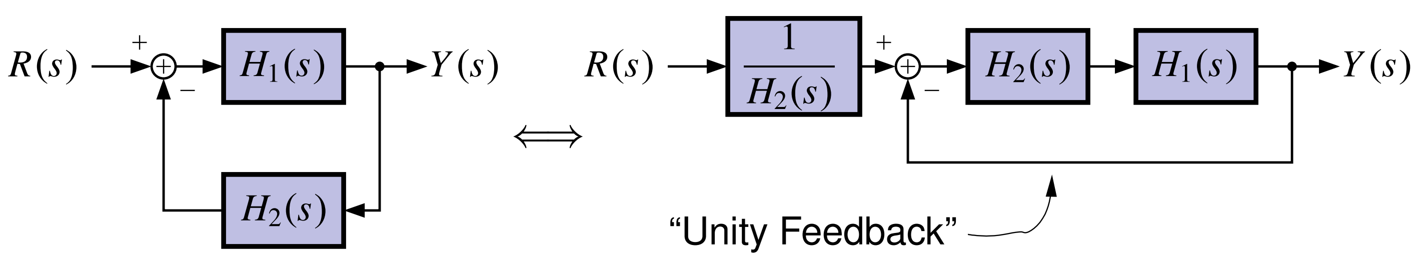


$$Y(s) = \frac{H_1(s)}{1 + H_2(s)H_1(s)}R(s)$$

- Block-diagram algebra (or Mason's rule) may be used to reduce block diagrams to a single transfer function.





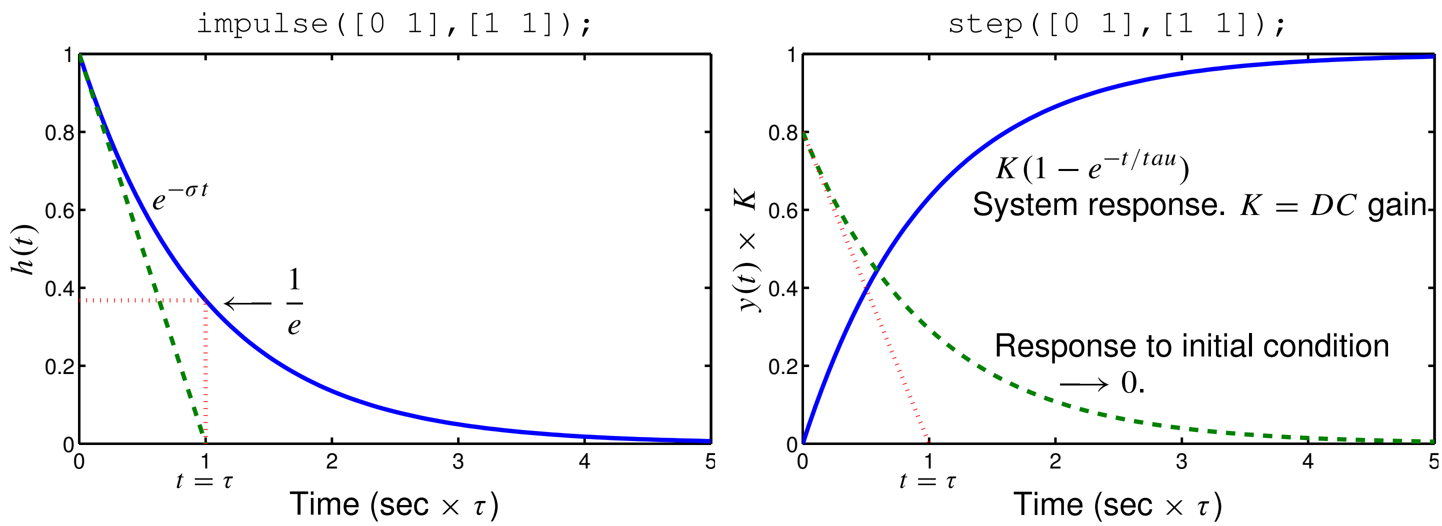


1.3: Review: Actual and desired dynamic response

- The poles of $H(s)$ determine (qualitatively) the dynamic response of the system. The zeros of $H(s)$ quantify the relationship.
- If the system has only real poles, each one has the form

$$H(s) = \frac{1}{s+\sigma}.$$

- If $\sigma > 0$, the system is stable, and $h(t) = e^{-\sigma t}1(t)$. The time constant is $\tau = 1/\sigma$, and the response of the system to an impulse or step decays to steady-state in about 4 or 5 time constants.

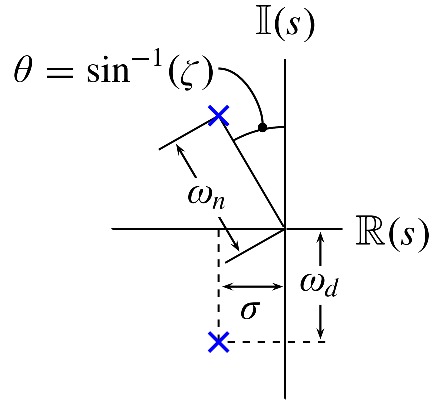


- If a system has complex-conjugate poles, each may be written as:

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}.$$

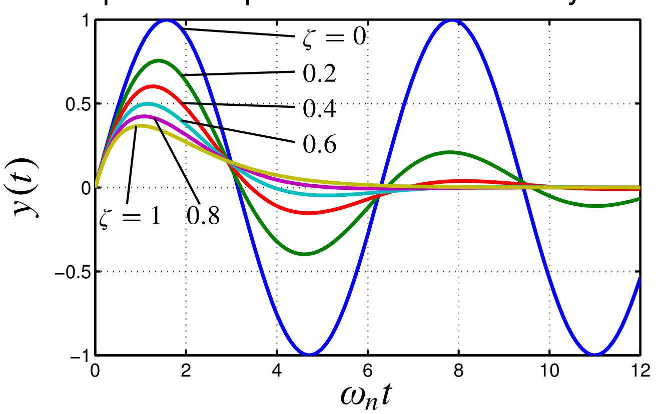
- We can extract two more parameters from this equation:

$$\sigma = \zeta\omega_n \text{ and } \omega_d = \omega_n\sqrt{1-\zeta^2}.$$

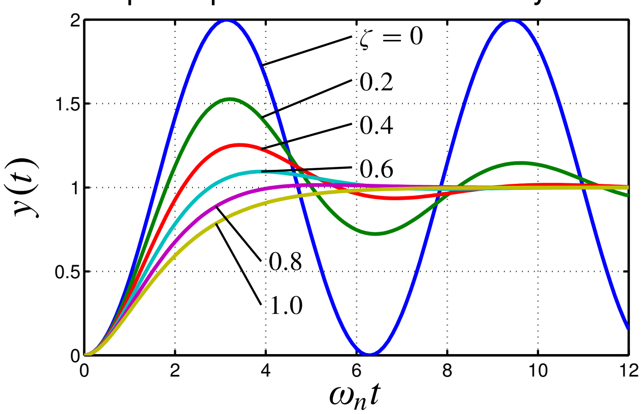


- σ plays the same role as above—it specifies decay rate of the response.
- ω_d is the oscillation frequency of the output. Note: $\omega_d \neq \omega_n$ unless $\zeta = 0$.
- ζ is the "damping ratio" and it also plays a role in decay rate and overshoot.
- Impulse response $h(t) = \omega_n e^{-\sigma t} \sin(\omega_d t)\ 1(t)$.
- Step response $y(t) = 1 - e^{-\sigma t}\left(\cos(\omega_d t) + \dfrac{\sigma}{\omega_d}\sin(\omega_d t)\right)$.

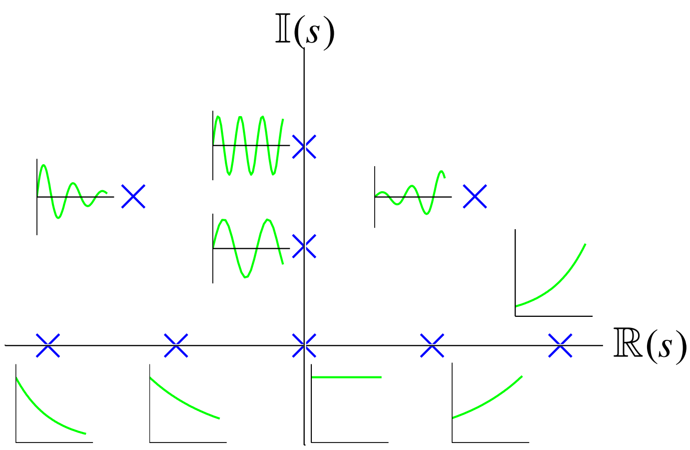
Impulse Responses of 2nd-Order Systems



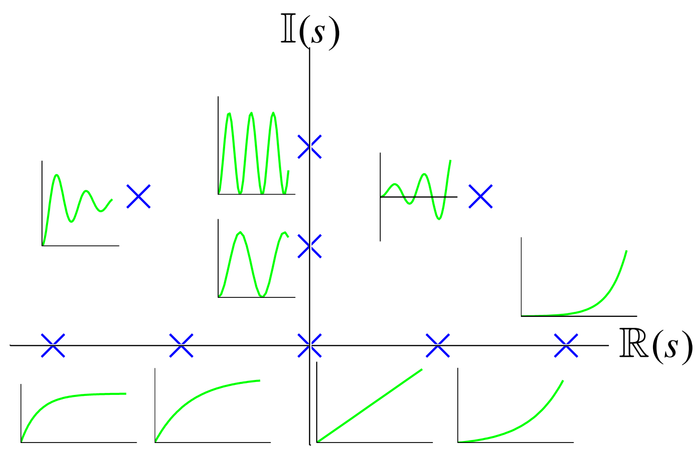
Step Responses of 2nd-Order Systems



- A summary chart of impulse responses and step responses versus pole locations is:

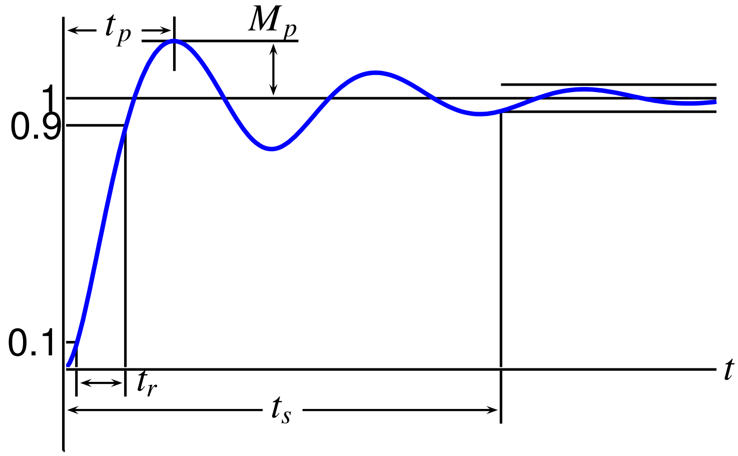


Impulse responses vs. pole locations

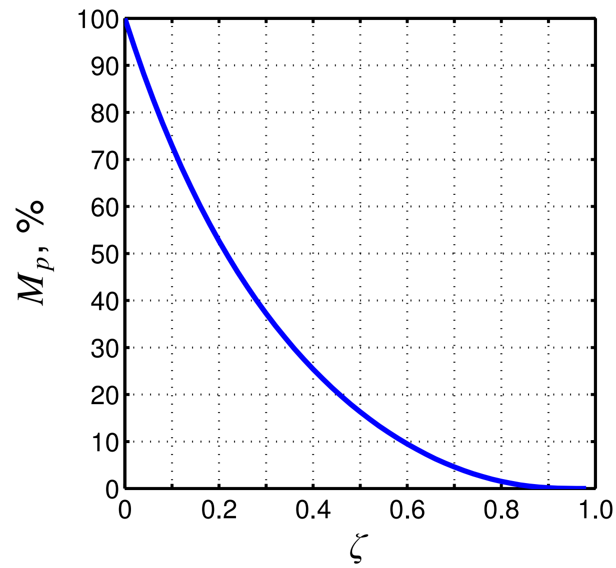


Step responses vs. pole locations

- Time-domain specifications determine where poles *SHOULD* be placed in the s-plane (step-response).



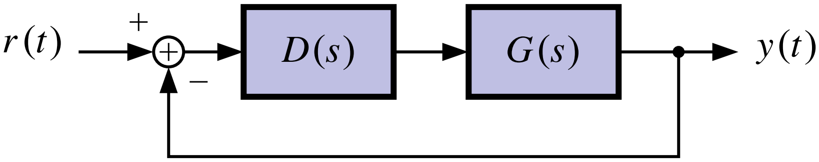
- Rise time t_r = time to go from 10% to 90% of final value.
- Settling time t_s = time until permanently within $\approx 1\%$ of final value.
- Overshoot M_p = maximum PERCENT overshoot.



- Some approximate relationships useful for initial design processes are:

$$
\begin{aligned}
t_r &\approx 1.8/\omega_n & \ldots && \omega_n &\ge 1.8/t_r \\
t_s &\approx 4.6/\sigma & \ldots && \sigma &\ge 4.6/t_s \\
M_p &\approx e^{-\pi\zeta/\sqrt{1-\zeta^2}} & \ldots && \zeta &\ge \mathrm{fn}(M_p).
\end{aligned}
$$

1.4: Review: Feedback control



$$\frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)} = T(s).$$

- Stability depends on roots of denominator of $T(s)$: $1 + D(s)G(s) = 0$.
- Routh test used to determine stability.
- Steady-state error found from (for unity-feedback case)

$$\frac{E(s)}{R(s)} = \frac{1}{1 + D(s)G(s)}.$$

- $e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s)$ if the limit exists.
  - System type $= 0$ iff e_{ss} is finite for unit-step $1(t)$.
  - System type $= 1$ iff e_{ss} is finite for unit-ramp $r(t)$.
  - System type $= 2$ iff e_{ss} is finite for unit-parabola $p(t)$.

    $\vdots$
- For unity-feedback systems,

$$K_p = \lim_{s\to 0} D(s)G(s), \qquad \text{“position error constant”}$$

$$K_v = \lim_{s\to 0} sD(s)G(s). \qquad \text{“velocity error constant”}$$

$$K_a = \lim_{s\to 0} s^2 D(s)G(s). \qquad \text{“acceleration error constant”}$$

- Steady-state errors versus system type for unity feedback:

	Step input	Ramp input	Parabola input
Type 0	$\dfrac{1}{1+K_p}$	∞	∞
Type 1	0	$\dfrac{1}{K_v}$	∞
Type 2	0	0	$\dfrac{1}{K_a}$

Some types of controllers

"Proportional" ctrlr:	$u(t) = Ke(t).$	$D(s) = K.$
"Integral" ctrlr	$u(t) = \dfrac{K}{T_I}\displaystyle\int_{-\infty}^{t} e(t)\,\mathrm{d}t.$	$D(s) = \dfrac{K}{T_I s}$
"Derivative" ctrlr.	$u(t) = KT_D\dot{e}(t)$	$D(s) = KT_D s$
Combinations:	PI: $\Longrightarrow D(s) = K(1 + \dfrac{1}{T_I s});$	
	PD: $\Longrightarrow D(s) = K\,(1 + T_D s)\,;$	
	PID: $\Longrightarrow D(s) = K\left(1 + \dfrac{1}{T_I s} + T_D s\right).$	
Lead:	$D(s) = K\dfrac{Ts+1}{\alpha Ts+1},$	$\alpha < 1$ (approx PD)
Lag:	$D(s) = K\dfrac{Ts+1}{\alpha Ts+1},$	$\alpha > 1$ (approx PI; often, $K = \alpha$)
Lead/Lag:	$D(s) = K\dfrac{(T_1 s+1)(T_2 s+1)}{(\alpha_1 T_1 s+1)(\alpha_2 T_2 s+1)},$	$\alpha_1 < 1$, $\alpha_2 > 1$.

Root locus

- A root locus plot shows (parametrically) the possible locations of the roots of the equation
$$1 + K\frac{b(s)}{a(s)} = 0.$$
- For a unity-gain feedback system,
$$T(s) = \frac{D(s)G(s)}{1 + D(s)G(s)}.$$
- The poles of the closed-loop system $T(s)$ depend on the open-loop transfer functions $D(s)G(s)$. Suppose $D(s) = KD_0(s)$.

$$\text{closed-loop poles at } 1 + K(D_0(s)G(s)) = 0$$

which is the root-locus form.

- Drawing the root locus allows us to select K for good pole locations. Intuition into the root-locus helps us design $D_0(s)$ with lead/ lag/ PI/ PID... controllers.

Root-locus drawing rules

- The steps in drawing a $180°$ root locus follow from the basic phase definition. This is the locus of

$$1 + K\frac{b(s)}{a(s)} = 0, \qquad K \geq 0 \qquad \left(\text{phase of } \frac{b(s)}{a(s)} = -180°\right)$$

- They are
  - STEP 1: On the s-plane, mark poles (roots of $a(s)$) by an $\times$ and zeros (roots of $b(s)$) with an $\circ$. There will be a branch of the locus departing from every pole and a branch arriving at every zero.
  - STEP 2: Draw the locus on the real axis to the left of an odd number of real poles plus zeros.
  - STEP 3: Draw the asymptotes, centered at α and leaving at angles ϕ, where

$$\begin{aligned}
n - m &= \text{number of asymptotes.} \\
n &= \text{order of } a(s) \\
m &= \text{order of } b(s) \\
\alpha &= \frac{\sum p_i - \sum z_i}{n-m} = \frac{-a_1 + b_1}{n-m} \\
\phi_l &= \frac{180° + (l-1)360°}{n-m}, \quad l = 1, 2, \ldots n-m
\end{aligned}$$

  For $n - m > 0$, there will be a branch of the locus approaching each asymptote and departing to infinity. For $n - m < 0$, there will be a branch of the locus arriving from infinity along each asymptote.
  - STEP 4: Compare locus departure angles from the poles and arrival angles at the zeros where

$$\begin{aligned}
q\phi_{dep} &= \sum \psi_i - \sum \phi_i - 180° - l360° \\
q\psi_{arr} &= \sum \phi_i - \sum \psi_i + 180° + l360°
\end{aligned}$$

  where q is the order of the pole or zero and l takes on q integer values so that the angles are between $\pm 180°$. ψ_i is the angle of the line going from the i_{th} zero to the pole or zero whose angle of departure or arrival is being computed. Similarly, ϕ_i is the angle of the line from the i_{th} pole.

- STEP 5: If further refinement is required at the stability boundary, assume $s_o = j\omega_o$ and compute the point(s) where the locus crosses the imaginary axis for positive K.
- STEP 6: For the case of multiple roots, two loci come together at $180°$ and break away at $\pm 90°$. Three loci segments approach each other at angles of $120°$ and depart at angles rotated by $60°$.
- STEP 7 Complete the locus, using the facts developed in the previous steps and making reference to the illustrative loci for guidance. The loci branches start at poles and end at zeros or infinity.
- STEP 8 Select the desired point on the locus that meets the specifications (s_o), then use the magnitude condition to find that the value of K associated with that point is

$$K = \frac{1}{|b(s_o)/a(s_o)|}.$$

- When K is negative, the definition of the root locus in terms of the phase relationship changes. We need to plot a $0°$ locus instead.

  $0°$ *locus definition:* The root locus of $b(s)/a(s)$ is the set of points in the s-plane where the phase of $b(s)/a(s)$ is $0°$.

- For this case, the steps above are modified as follows:
  - STEP 2: Draw the locus on the real axis to the left of an *even* number of real poles plus zeros.
  - STEP 3: The asymptotes depart at

$$\phi_l = \frac{(l-1)360°}{n-m}, \quad l = 1, 2, \ldots n-m.$$

  - STEP 4: The locus departure and arrival angles are modified to

$$q\phi_{dep} = \sum \psi_i - \sum \phi_i - l360°$$
$$q\psi_{arr} = \sum \phi_i - \sum \psi_i + l360°.$$

    Note that the $180°$ term has been removed.

  - STEP 8 Select the desired point on the locus that meets the specifications (s_o), then use the magnitude condition to find that the value of K associated with that point is

$$K = \frac{-1}{|b(s_o)/a(s_o)|}.$$

1.5: Review: Frequency response

- The frequency response of a system tells us directly the relative magnitude and phase of a system's output sinusoid if the system input is a sinusoid. [What about output frequency?]
- If the plant's transfer function is $G(s)$, the open-loop frequency response is $G(j\omega) = G(s)|_{s=j\omega}$.
- We can plot $G(j\omega)$ as a function of ω in three ways:
  - Bode Plot.
  - Nyquist Plot.
  - Nichols Plot (not covered in ECE4510/5510).

Bode plots

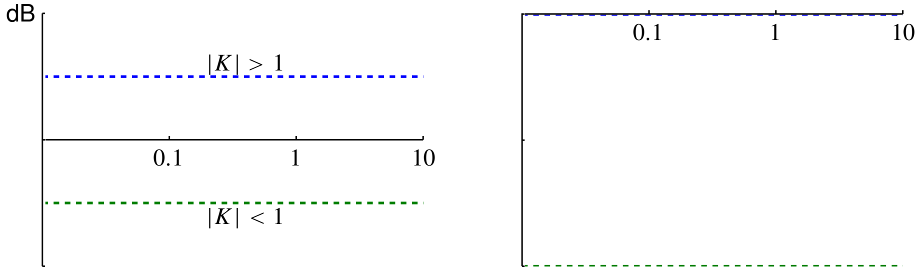
- A 'Bode plot' is really two plots: $20\log_{10}|G(j\omega)|$ versus ω and $\angle G(j\omega)$ versus ω. The Bode-magnitude plot is plotted on a log-log scale. The Bode-phase plot is plotted linear-log scale.
- The transfer function $G(s)$ is broken up into its component parts, and each part is plotted separately.
- The factorization must be of the form

$$KG(s) = K_o(s)^n \frac{(s\tau_1+1)(s\tau_2+1)\cdots}{(s\tau_a+1)(s\tau_b+1)\cdots},$$

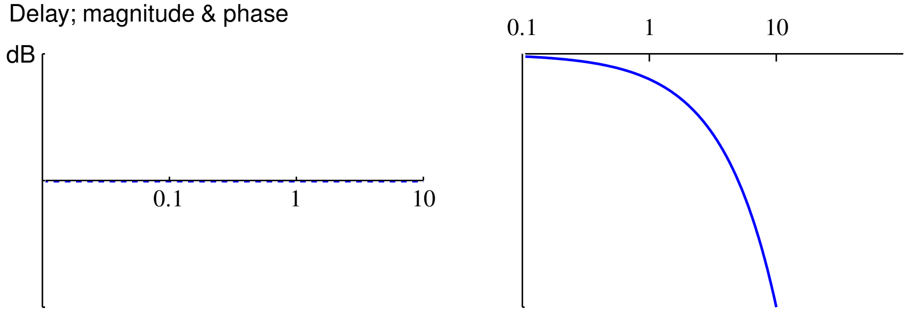
  which separates real zeros and poles, complex zeros and poles, zeros or poles at the origin, delay, and gains.
- The following "dictionary" is used when plotting a transfer function which has been factored into Bode form.

Dictionary of Bode-plot relationships

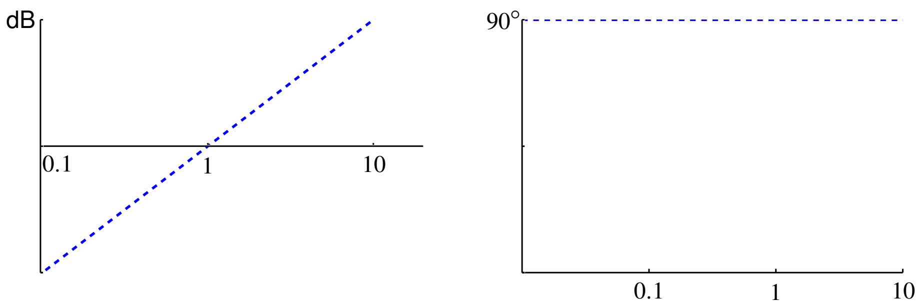
Gain of K; magnitude & phase



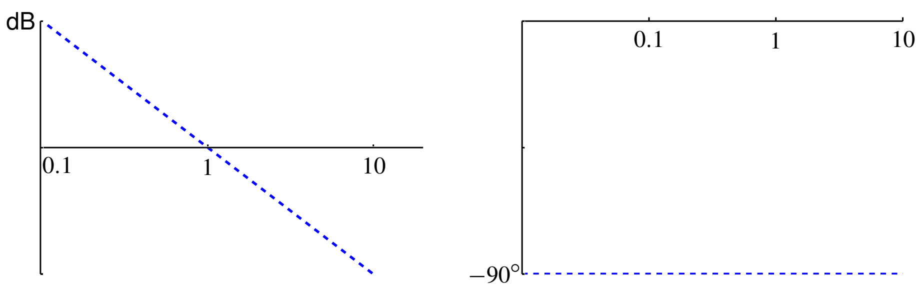
Delay; magnitude & phase



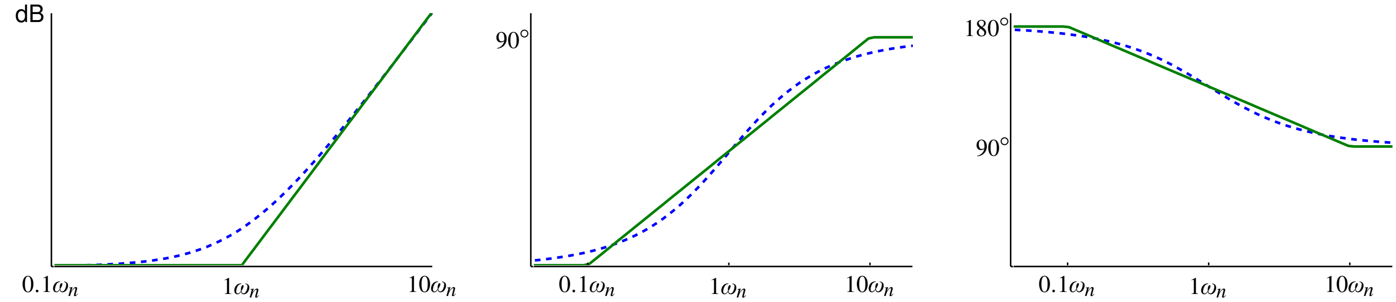
Zero at origin; magnitude & phase



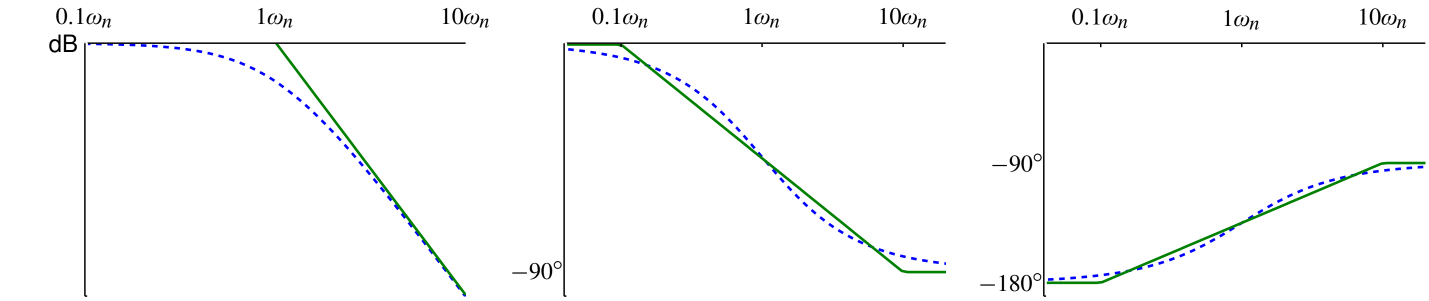
Pole at origin; magnitude & phase



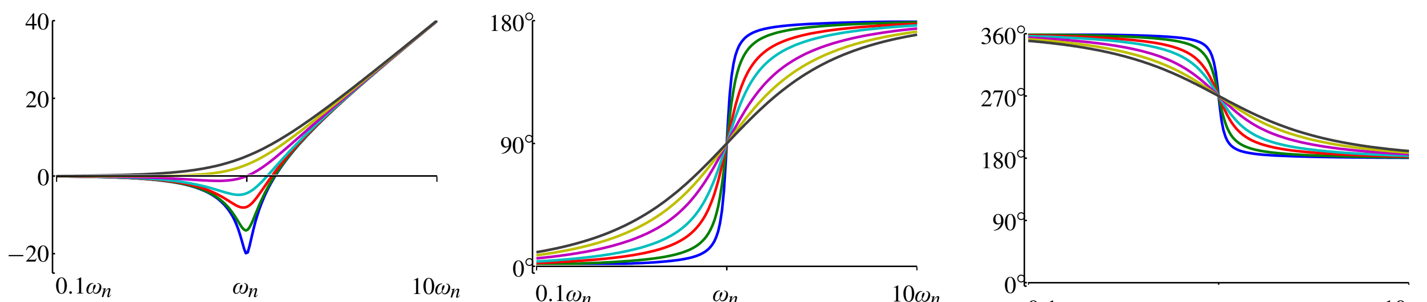
Zero on real axis; magnitude & mp-phase & nmp-phase



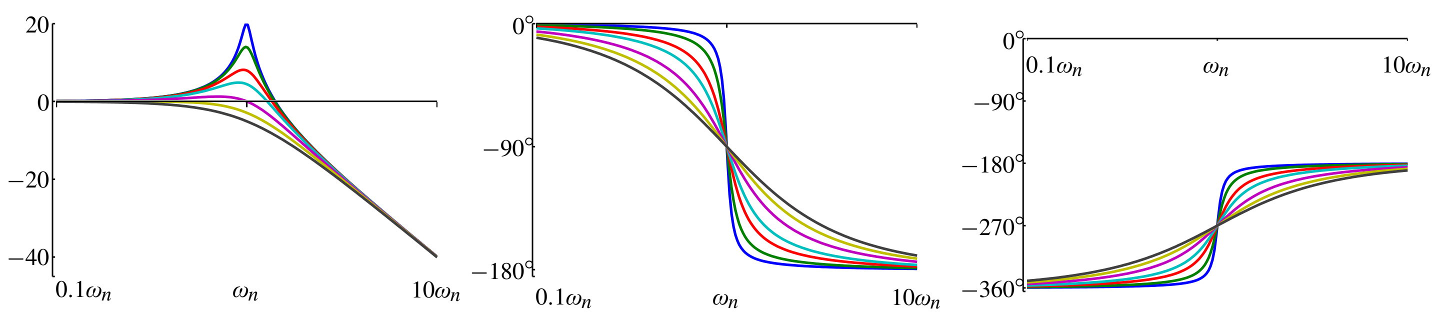
Pole on real axis; magnitude & mp-phase & nmp-phase



Complex zeros; magnitude & mp-phase & nmp-phase



Complex poles; magnitude & mp-phase & nmp-phase



Bode-plot techniques

- It is useful to be able to plot the frequency response of a system by hand in order to
  - Design simple systems without the aid of a computer,
  - Check computer-based results, and

- Understand the effect of compensation changes in design iterations.

■ H.W. Bode developed plotting techniques in the 1930s that enabled quick hand potting of the frequency response. His rules are:

- STEP 1: Manipulate the transfer function into the **Bode form**

$$KG(j\omega) = K_o(j\omega)^n \frac{(j\omega\tau_1 + 1)(j\omega\tau_2 + 1)\cdots}{(j\omega\tau_a + 1)(j\omega\tau_b + 1)\cdots}$$

- STEP 2: Determine the value of n for the $K_o(j\omega)^n$ term. Plot the low-frequency magnitude asymptote through the point K_o at $\omega = 1$ rad/sec with the slope of n (or $n \times 20$ dB per decade).
- STEP 3: Determine the **break point**s where $\omega = 1/\tau_i$. Complete the composite magnitude asymptotes by extending the low frequency asymptote until the first frequency break point, then stepping the slope by ± 1 or± 2, depending on whether the break point is from a first or second order term in the numerator or denominator, and continuing through all break points in ascending order.
- STEP 4: Sketch in the approximate magnitude curve by increasing from the asymptote by a factor of 1.4 ($+3$dB) at first order numerator breaks and decreasing it by a factor of 0.707 (-3dB) at first order denominator breaks. At second order break points, sketch in the resonant peak (or valley) using the relation that $|G(j\omega)| = 1/(2\zeta)$ at the break.
- STEP 5: Plot the low frequency asymptote of the phase curve, $\phi = n \times 90°$.
- STEP 6: As a guide, sketch in the approximate phase curve by changing the phase gradually over two decades by $\pm 90°$ or $\pm 180°$ at each break point in ascending order. For first order terms in the numerator, the gradual change of phase is $+90°$; in the denominator, the change is $\pm 180°$.
- STEP 7: Locate the asymptotes for each individual phase curve so that their phase change corresponds to the steps in the phase from the approximate curve indicated by Step 6. Sketch in each individual phase.
- STEP 8 Graphically add each phase curve. Use dividers if an accuracy of about $\pm 5°$ is desired. If lessor accuracy is acceptable, the composite curve can be done by eye, keeping in mind that the curve will start at the lowest frequency asymptote and end on the highest frequency asymptote, and will approach the intermediate asymptotes to an extent that is determined by the proximity of the break points to each other.

Nyquist plots

■ Nyquist plot is a mapping of loop transfer function $D(s)G(s)$ along the Nyquist path to a polar plot.

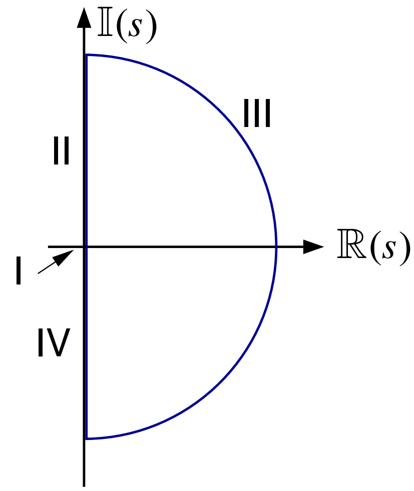
- Think of Nyquist path as four parts:

  I: Origin. Sometimes a special case.

  II: $+j\omega$ axis. *FREQUENCY* response of O.L. system! Just plot it as a polar plot.

  III: For many physical systems, zero. Some counter-examples.

  IV: Complex conjugate of **II**.

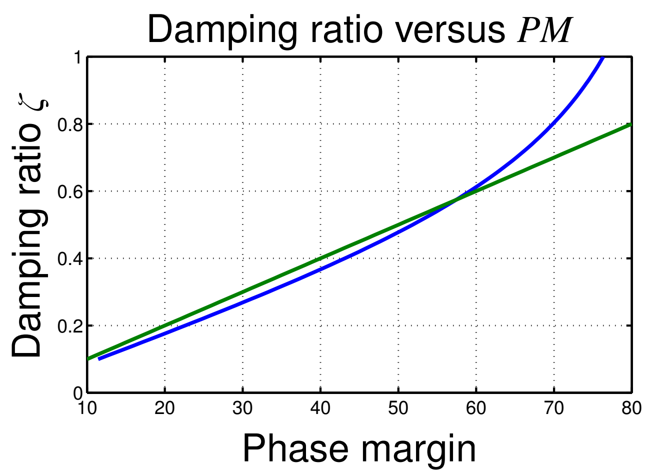


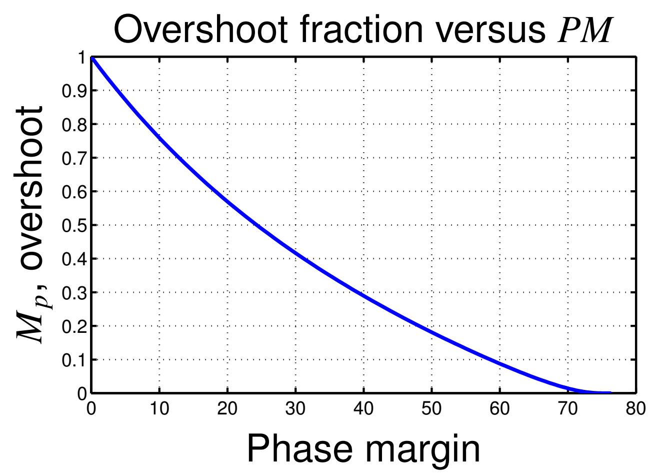
- The test:
  - N = #CW encirclements of $-1/K$ point when $F(s) = D(s)G(s)$.
  - P = # of OPEN-LOOP unstable poles.
  - Z = # of CLOSED-LOOP unstable poles.
  - $Z = N + P$
- The system is stable iff $Z = 0$.
- For poles on the $j\omega$-axis, use modified Nyquist path.

Bode-plot performance

- At low frequency, approximate Bode plot with $KG(j\omega) = K_0(j\omega)^n$.
  - n = system type ... slope = $-20n$ dB/decade.
  - $K_0 = K_p$ for type 0, $= K_v$ for type 1, ...
- Gain margin (for systems that become unstable with increasing gain) = factor by which gain is less than 0 dB when phase = $-180°$.
- Phase margin (for same systems) = amount phase is greater than $-180°$ when gain = 1.

- PM related to damping. $\zeta \approx PM/100$ when $PM < 70°$.
- PM therefore related to M_p.





- Gain-phase relationship: For minimum-phase systems, $\angle G(j\omega) \approx n \times 90°$ where n = slope of log-log plot of gain ($n = -1$ if slope $= -20$ dB/decade, $n = -2$ if slope $= -40$ dB/decade ...)
- Therefore want slope at gain crossover ≈ -20 dB/decade for decent phase margin.

Where from here?

- We have reviewed continuous-time control concepts.
- Can any of these be applied directly to the discrete-time control problem?
- We look next at two different ways to emulate continuous-time controllers using discrete-time methods, allowing a quick redesign and control implementation.
- However, we also find that these methods are not optimized and that we can do better by designing digital controllers from scratch.